

**AIX Version 3.2
Systembenutzung**



**AIX Version 3.2
Systembenutzung**

Diese Veröffentlichung ist eine Übersetzung des *System User's Guide* IBM Form GC23-2377-03, herausgegeben von der IBM International Business Machines Corporation, USA.

- © Copyright International Business Machines Corporation, 1987, 1993
- © Copyright IBM Deutschland GmbH, 1993
- © Copyright AT&T, 1984, 1985, 1986, 1987, 1988, 1989. Alle Rechte vorbehalten.
- © Copyright INTERACTIVE Systems Corporation 1984. Alle Rechte vorbehalten.

Möglicherweise sind nicht alle in dieser Übersetzung aufgeführten Produkte in Deutschland angekündigt und verfügbar; vor Entscheidungen empfiehlt sich der Kontakt mit der jeweiligen Geschäftsstelle.

Die Verantwortung für eine Verknüpfung von Produkten, Programmen oder Dienstleistungen verschiedener Hersteller liegt beim Kunden.

Für in diesem Handbuch beschriebene Erzeugnisse und Verfahren kann es Patente oder Patentanmeldungen geben. Mit der Auslieferung dieses Handbuchs ist keine Lizenzierung dieser Patente verbunden.

Trotz sorgfältiger Bearbeitung können technische Ungenauigkeiten in dieser Veröffentlichung nicht ausgeschlossen werden.

Änderung des Textes bleibt vorbehalten.

Herausgegeben von:
MH NLS Center
Kst. 6599
Juli 1993

Warenzeichen

Auf folgende Warenzeichen wird im vorliegenden Handbuch verwiesen:

AIX ist ein Warenzeichen der International Business Machines Corporation.

AIXwindows ist ein Warenzeichen der International Business Machines Corporation.

IBM ist ein eingetragenes Warenzeichen der International Business Machines Corporation.

INed ist ein Warenzeichen der INTERACTIVE Systems Corporation.

InfoExplorer ist ein Warenzeichen der International Business Machines Corporation.

PostScript ist ein Warenzeichen von Adobe Systems Incorporated.

RISC System/6000 ist ein Warenzeichen der International Business Machines Corporation.

UNIX wurde lizenziert und ist ein eingetragenes Warenzeichen der UNIX System Laboratories, Inc.

Einleitung

Das vorliegende Buch enthält Informationen für Einsteiger. Es führt ein in grundlegende Verarbeitungsvorgänge, wie z. B. Anmelden, Verwenden verschiedener Benutzerschnittstellen, Arbeiten mit Dateien und Verzeichnissen und Verwenden der Information, die dieses System unterstützt. Es führt außerdem ein in die wesentlichen Komponenten des Betriebssystems, wie z. B. das Dateisystem und die verschiedenen Shells.

Hinweis: Die Informationen in diesem Handbuch sind auch in der Hypertextdatenbank enthalten. Diese Online-Dokumentation ist für die Verwendung mit dem Programm InfoExplorer vorgesehen.

Zielgruppe

Dieses Handbuch richtet sich an alle Systembenutzer.

Aufbau des Handbuchs

Überblick über den Inhalt

Das vorliegende Handbuch besteht aus folgenden Kapiteln:

- Kapitel 1, "Lernprogramm: Erste Schritte", enthält Prozeduren zum Starten des Systems, Anmelden im und Abmelden aus dem System und Ausführen einiger einfacher Befehle. Dieses Kapitel führt außerdem ein in die grafische (Fenster) und in die Befehlszeilen-Schnittstelle (Shell).
- Kapitel 2, "Grundlagen des Betriebssystems", beinhaltet Überblicke und einige Prozeduren zu den Grundlagen des Betriebssystems, wie z. B. Dateisysteme, Prozesse, Shells, Ein- und Ausgabe, Profile, An- und Abmelden, virtuelle Datenstationen und Befehle.
- Kapitel 3, "Dateien, Verzeichnisse und allgemeine Aufgaben", enthält eine Einführung und Beispiele zum Arbeiten mit Dateien und Verzeichnissen sowie alltägliche Aufgaben im Betriebssystem, wie z. B. Sichern und Drucken von Dateien, Formatieren von Disketten und Wechseln zu einem Root.
- Kapitel 4, "AIXwindows Window Manager", beschreibt die Verwaltung von Fenstern mit dem AIXwindows Window Manager.
- Kapitel 5, "AIXwindows Desktop", führt ein in die Arbeit mit dem Desktop und die verschiedenen Desktop-Typen.
- Kapitel 6, "Netze und TCP/IP", definiert Begriffe, Komponenten und Anwendungen für physische und logische Netze und beinhaltet Informationen und Prozeduren für die am häufigsten benutzte Protokollgruppe, das Transmission Control Protocol/Internet Protocol (TCP/IP).
- Kapitel 7, "Post", enthält Informationen und Prozeduren zum Senden, Empfangen und Speichern von Nachrichten.
- Kapitel 8, "Online-Informationen und Hilfetext", bietet eine Einführung in InfoExplorer und seine neusten Funktionen. Darüberhinaus enthält es ein Lernprogramm, das Einsteigern hilft, die Grundlagen von InfoExplorer in einer Grafik- oder ASCII-Schnittstelle zu erlernen. Weitere Prozeduren beschreiben, wie Befehle zum Lokalisieren von Systeminformationen verwendet werden.

- Kapitel 9, "Editoren und Textformatierung", beschreibt die wichtigsten Editoren und das Textformatierungssystem.
- Kapitel 10, "Problemlösung und Hardwareinformation", bezieht sich auf die Handbücher, die Informationen über Hardware- und Softwareoperationen enthalten.
- Anhang A, "Korn-Shell", beschreibt die Verwendung der Korn-Shell.
- Anhang B, "Bourne-Shell", beschreibt die Verwendung der Bourne-Shell.
- Anhang C, "C-Shell", beschreibt die Verwendung der C-Shell.

Hervorhebung

Die folgenden Konventionen werden im vorliegenden Handbuch zur Hervorhebung verwendet:

Fett	Kennzeichnet Befehle, Schlüsselwörter, Dateien, Verzeichnisse und andere Elemente, deren Namen vom System festgelegt werden. Kennzeichnet außerdem grafische Objekte wie Knöpfe, Markierungen und Symbole, die vom Benutzer ausgewählt werden.
<i>Kursiv</i>	Kennzeichnet Parameter, deren tatsächliche Namen oder Werte vom Benutzer einzugeben sind.
Monospace	Kennzeichnet Beispiele für bestimmte Datenwerte, Textbeispiele, die so oder ähnlich angezeigt werden, Beispiele für Teile des Programmcodes, die so oder ähnlich von Programmierern geschrieben werden, Systemnachrichten sowie Angaben, die vom Benutzer einzugeben sind.

Referenzliteratur

Die folgende Liste enthält Referenzliteratur. Informationen zum Bestellen dieser Veröffentlichungen erhalten Sie bei Ihrem Vertragshändler oder Vertriebsbeauftragten.

- *Dokumentationsüberblick*, IBM Form SC12-2158
- *Task Index, Topic Index and Glossary*, IBM Form GC23-2201.
- *Systemüberblick*, IBM Form GC12-2111
- *Hypertext Information Base Library for AIX, Version 3.2* (eine CD-ROM, die die Hypertext-Bibliothek enthält), IBM Form SC23-2163
- *Kurzreferenz*, IBM Form SC12-2118
- *Installationsplanung*, IBM Form GC12-2112
- *AIX Version 3.2 System Management Guide*, IBM Form SC23-2457
- *AIX Version 3.2 Befehlsreferenz Endbenutzerbefehle*, IBM Form GC12-2165 (Bd. 1), GC12-2116 (Bd. 2)
- *AIX Version 3.2 Files Reference*, IBM Form GC23-2200
- *AIXwindows Environment/6000 Version 1.2 AIXwindows und AIXwindows Desktop-Benutzung*, IBM Form GC23-2432
- *AIX Version 3.2 Communication Concepts and Procedures*, IBM Form GBOF-1524
- *RISC System/6000 AIX Version 3.2 Editorkonzepte und -prozeduren*, IBM Form GC12-1972
- *AIX Version 3.2 Text Formatting Reference*, IBM Form SC23-2453
- *AIX Version 3.2 Glossar*, IBM Form GC12-1972

Bestellung weiterer Exemplare dieses Handbuchs

Weitere Exemplare dieses Handbuchs können über die folgende Formnummer bestellt werden: GC12-2092-03.

Inhalt

Warenzeichen	iii
Einleitung	v
Zielgruppe	v
Aufbau des Handbuchs	v
Referenzliteratur	vi
Bestellung weiterer Exemplare dieses Handbuchs	vii
Kapitel 1. Lerntext: Erste Schritte	1-1
1. Schritt: System einschalten	1-2
2. Schritt: System starten	1-2
3. Schritt: Bildschirm einschalten	1-3
4. Schritt: Anmelden	1-4
5. Schritt: Benutzerschnittstelle festlegen	1-4
6. Schritt: Kennwort erstellen und ändern	1-6
7. Schritt: Spezielle Tastenfunktionen benutzen	1-7
8. Schritt: Befehle eingeben	1-7
9. Schritt Abmelden	1-10
Kapitel 2. Grundlagen des Betriebssystems	2-1
Überblick über Dateisysteme	2-2
Reorganisierter Dateibaum	2-2
Dateisystemtypen	2-3
Dateisystemverwaltungsaufgaben	2-3
Einführung in den reorganisierten Dateibaum	2-5
Der neue Dateibaum	2-5
Einführung in das Stammdateisystem	2-6
Einführung in das Dateisystem /usr	2-8
Einführung in das Verzeichnis /usr/share	2-10
Einführung in das Dateisystem /var	2-11
Einführung in das Verzeichnis /export	2-12
Überblick über Prozesse	2-14
Prozesse starten	2-15
Prozesse abbrechen	2-16
Alle aktuellen Prozesse anzeigen	2-18
Alle Prozesse von der Befehlszeile aus anzeigen	2-18
Alle Prozesse mit der Einrichtung SMIT anzeigen	2-18
Prozeßstatus anzeigen	2-19
Prozeßstatus von der Befehlszeile aus anzeigen	2-19
Prozeßstatus mit der Einrichtung SMIT anzeigen	2-19
Prozesse löschen	2-20
Vorbedingung	2-20
Prozesse von der Befehlszeile aus löschen	2-20
Prozesse mit der Einrichtung SMIT löschen	2-20
Überblick über Shells	2-21
Einführung in die Shells	2-21

Shell-Funktionen	2-21
Zur Verfügung stehende Shells	2-22
Definition von Shell-Begriffen	2-22
Verwendung der Shell-Prozedur	2-25
Eine Shell für eine Prozedurdatei angeben	2-25
Eine Shell-Prozedur erstellen und ausführen	2-27
Voraussetzungen	2-27
Erforderliche Schritte	2-27
Überblick über Ein- und Ausgabe	2-28
Ein- und Ausgabe mit Befehlen steuern	2-28
Ein- und Ausgabe mit Symbolen steuern	2-28
Überblick über Profile	2-31
Datei /etc/profile	2-31
Die Datei .profile	2-31
Überblick über das An- und Abmelden	2-33
Anmelden an mehreren Systemen	2-33
Mehrfaches Anmelden	2-33
Anmelden am Betriebssystem	2-35
Bei nicht aktiviertem System	2-35
Bei aktiviertem System	2-35
Abmelden vom Betriebssystem	2-36
Abmelden und aktiviertes System verlassen	2-36
Betriebssystem stoppen	2-36
Anmeldnachrichten unterdrücken	2-37
Erforderliche Schritte	2-37
Überblick über virtuelle Datenstationen	2-38
Empfohlene Dokumentation	2-38
Tastenfunktionen des Betriebssystems AIX	2-38
Mit virtuellen Datenstationen arbeiten	2-40
Erforderliche Schritte	2-40
Überblick über Befehle	2-41
Syntaxdiagramme	2-41
Befehle im Hintergrund ausführen	2-44
Befehle eingeben	2-44
Kapitel 3. Dateien, Verzeichnisse und allgemeine Aufgaben	3-1
Überblick über Dateien und Verzeichnisse	3-2
Einführung in Dateien	3-3
Einführung	3-3
Dateitypen	3-4
Konventionen beim Benennen von Dateien	3-4
Mit Dateien arbeiten	3-5
Dateien erstellen, kombinieren und editieren	3-5
Dateien anzeigen	3-5
Dateien verschieben, kopieren und umbenennen	3-6
Dateien löschen	3-6
Dateien verbinden	3-6
DOS-Dateien umsetzen	3-7
Dateien sichern	3-7
Andere Dateibefehle verwenden	3-7
Einführung in Verzeichnisse	3-8
Einführung in den Aufbau von Verzeichnissen	3-8

Einführung in Pfadnamen	3-9
Einführung in die Verzeichnistypen	3-10
Mit Verzeichnissen arbeiten	3-11
Einführung in die Verbindungsfunktion	3-11
Einführung in den Zugriff auf Dateien und Verzeichnisse	3-13
Eignerstatus und Benutzergruppen	3-13
Dateizugriffsmodi	3-13
Dateiverwaltung	3-16
Dateien löschen	3-16
Dateien verschieben	3-17
Dateien kopieren	3-17
Dateien drucken	3-18
Inhalt von Dateien anzeigen	3-19
Dateien umbenennen	3-19
Programmausgabe anzeigen und in eine Datei kopieren	3-20
Dateien verbinden	3-20
Dateien suchen	3-21
Zeichenfolgen suchen	3-22
Wörter, Zeilen und Zeichen zählen	3-23
Teile von Dateien anzeigen	3-24
Den Dateityp anzeigen	3-24
Verfügbaren Speicherbereich in einem Dateisystem anzeigen	3-25
Text in einer Datei bearbeiten	3-26
Dateien verknüpfen	3-26
Abschnitte aus Dateien ausschneiden	3-26
Abschnitte in eine Datei einfügen	3-27
Zeilen in einer Datei numerieren	3-28
Spalten aus einer Datei löschen	3-28
Daten in einer Datei komprimieren	3-28
Gepackte Dateien entpacken	3-30
Mit Verzeichnissen arbeiten	3-31
Verzeichnisse erstellen	3-31
Verzeichnisse verschieben	3-31
Aktuelles Verzeichnis ermitteln	3-32
Aktuelles Verzeichnis wechseln	3-32
Verzeichnisse kopieren	3-33
Inhalt von Verzeichnissen anzeigen	3-33
Verzeichnisse löschen bzw. entfernen	3-36
Eignerstatus für Verzeichnisse ändern	3-36
Inhalt von Verzeichnissen vergleichen	3-37
DOS-Dateien umwandeln	3-38
DOS-Dateien in AIX-Dateien kopieren	3-38
AIX-Dateien in DOS-Dateien kopieren	3-38
DOS-Dateien löschen	3-39
Inhalt des DOS-Verzeichnisses auflisten	3-39
Überblick über Möglichkeiten der Sicherung	3-40
Einführung in die Möglichkeiten der Sicherung	3-40
Arten von Sicherungsdatenträgern	3-40
Daten zurückschreiben	3-41
Richtlinien für die Vorgehensweise beim Sichern	3-42
Dateien sichern	3-44
Beispiele	3-44

Überblick über Drucker und Warteschlangen	3-46
Druckjob starten	3-46
Priorität an einen lokalen Druckjob vergeben	3-47
Status eines Druckjobs anzeigen	3-47
Druckjob abbrechen	3-47
Starten eines Druckjobs	3-48
Voraussetzungen	3-48
Erforderliche Schritte	3-48
Einen Druckjob abbrechen	3-50
Voraussetzungen	3-50
Erforderliche Schritte	3-50
Den Status eines Druckjobs prüfen	3-51
Voraussetzungen	3-51
Erforderliche Schritte	3-51
Priorität an einen Druckjob vergeben	3-52
Voraussetzungen	3-52
Erforderliche Schritte	3-52
Umsetzen und Drucken von ASCII-Dateien auf einem PostScript-Drucker	3-53
Voraussetzungen	3-53
Erforderliche Schritte	3-53
Umsetzungsoptionen	3-54
Allgemeine Aufgaben	3-56
Disketten formatieren	3-56
Den Anmeldenamen anzeigen	3-57
Liste der angemeldeten Benutzer anzeigen	3-57
Die Systemidentifikation eines angegebenen Benutzers anzeigen	3-57
Den Systemnamen anzeigen	3-58
Die aktuelle Umgebung anzeigen	3-59
Das Kennwort ändern	3-59
Zur Superuserberechtigung wechseln	3-59
Eine Nachricht an einen anderen angemeldeten Benutzer senden	3-60
Einen Kalender anzeigen	3-61
Den Bildschirm löschen	3-62
Kapitel 4. AIXwindows Window Manager	4-1
Übersicht über den AIXwindows Window Manager	4-1
AIXwindows Window Manager and AIXwindows Desktop	4-2
Einführung in AIXwindows	4-5
X Window System und AIXwindows	4-5
Einführung in das Fensterkonzept	4-5
Funktionsweise von AIXwindows	4-7
X und AIXwindows Window Manager starten und verlassen	4-9
Standardeinstellungen wiederherstellen	4-9
Benutzung des AIXwindows Window Manager	4-11
Interaktion mit MWM	4-11
Zugriff auf Fensterverwaltungsfunktionen	4-12
Fenster für Eingabe auswählen	4-13
Fenster in den Vordergrund stellen	4-14
Benutzung des MWM-Fenstermenüs	4-15
Fenster verschieben	4-16
Fenstergröße ändern	4-17

Fenster auf maximale Größe vergrößern	4-17
Benutzung von Symbolen	4-18
Fenster schließen	4-19
Benutzung des Standardstammenüs	4-19
Interaktion mit AIXwindows-Anwendungen	4-21
Arbeiten mit dem Hauptfenster	4-21
Benutzung von Knöpfen	4-22
Benutzung von Menüs	4-24
Benutzung von Dialogfenstern	4-29
Benutzung von einstellbaren Komponenten	4-32
Benutzung von Listen	4-35
Text eingeben und editieren	4-36
Benutzung der Tastatur	4-38
Ziehen und übergeben	4-39
Kapitel 5. AIXwindows Desktop	5-1
AIXwindows Desktop-Merkmale	5-2
AIXwindows Desktop starten	5-5
AIXwindows Desktop-Anzeigekomponenten	5-7
Mit der Maus bei AIXwindows Desktop arbeiten	5-9
Drei- oder Zweitastenmaus verwenden	5-9
Basisaktionen der Maustasten	5-10
AIXwindows Desktop-Symbole	5-12
Symboltypen	5-13
Zugriffsmerkmale für Symbole	5-15
Symbolmenüs	5-16
Symbole auswählen und die Auswahl zurücknehmen	5-18
Symbole aktivieren	5-20
Symbole verschieben und kopieren	5-21
Symbole umbenennen	5-22
Symbole löschen	5-23
AIXwindows Desktop-Menüs	5-24
Eine AIXwindows-Menüoption auswählen	5-25
Arbeitsoberflächenmenüs und -optionen bei AIXwindows	5-26
Verzeichnismenüs und -optionen bei AIXwindows	5-30
Tastenäquivalente für Menüoptionen bei AIXwindows	5-34
Arbeitsoberflächentypen bei AIXwindows Desktop	5-38
General Desktop	5-39
Symbole des General Desktop	5-40
Sonderdateien des General Desktop	5-42
Sichtbare Dateien des General Desktop	5-42
Zubehörsymbole des General Desktop	5-43
Power Desktop	5-45
Symbole des Power Desktop	5-46
Sonderdateien des Power Desktop	5-48
Sichtbare Dateien des Power Desktop	5-49
Symbole des Verzeichnisses "Power"	5-49
Admin Desktop	5-51
Symbole des Admin Desktop	5-52
Sichtbare Dateien des Admin Desktop	5-55
Symbole des Verzeichnisses "Admin"	5-55

Neue Benutzer definieren	5-57
Erforderliche Schritte	5-57
Den codierten Zeichensatz ISO8859 installieren	5-58
Erforderliche Schritte	5-58
Kapitel 6. Netze und TCP/IP	6-1
Überblick über Netze	6-2
Einführung in Netze	6-2
Physische Netze	6-3
Übertragungsunterstützung für Systemhard- und -software	6-3
DFV-Funktionen für Benutzer und Anwendungen	6-6
Mit anderen Betriebssystemen kommunizieren	6-6
Überblick über das TCP/IP	6-8
TCP/IP und Internet	6-8
Defense Advanced Research Projects Agency (DARPA)	6-8
Einführung in TCP/IP-Befehle für Endbenutzer	6-9
Befehle zur Dateiübertragung	6-9
Fernanmeldung	6-10
Statusbefehle	6-10
Fernverbindung	6-11
Einführung in den Befehl telnet	6-12
Einführung in den Befehl rlogin	6-12
Einführung in die Befehlsausführung auf einem fernen System	6-13
Der Befehl rsh	6-13
Der Befehl rexec	6-13
Informationen über angemeldete Benutzer anzeigen	6-15
Voraussetzungen	6-15
Informationen über alle an einem Host angemeldeten Benutzer anzeigen	6-15
Informationen über einen einzelnen, an einem Host angemeldeten Benutzer anzeigen	6-15
Mit dem Befehl ftp Dateien kopieren	6-16
Voraussetzungen	6-16
An einem fernen Host anmelden	6-16
Eine Datei von einem fernen zu einem lokalen Host kopieren	6-17
Eine Datei von einem lokalen zu einem fernen Host kopieren	6-17
Von einem fernen System aus drucken	6-19
Voraussetzungen	6-19
Erforderliche Schritte	6-19
An einem fernen System drucken	6-20
Voraussetzungen	6-20
Verwendung des Befehls enq *	6-20
Verwendung des System Management Interface Tool (SMIT)	6-20
Dialog mit einem Benutzer auf einem fernen System	6-21
Voraussetzungen	6-21
Den Dialog mit dem Benutzer auf einem fernen System starten	6-21
Eine Dialoganfrage beantworten	6-21
Eine Dialoganfrage beenden	6-21
Einen lokalen Host mit einem fernen Host verbinden	6-22
Voraussetzungen	6-22
Erforderliche Schritte	6-22
Dokumentation	6-22

Mit dem Befehl tftp Dateien kopieren	6-23
Voraussetzungen	6-23
Eine Datei von einem fernen Host kopieren	6-23
Eine Datei zu einem fernen Host kopieren	6-23
Kapitel 7. Post	7-1
Überblick über die Postfunktion	7-2
Post erstellen und senden	7-3
Post adressieren	7-3
Post an Benutzer des lokalen Systems senden	7-4
Post an Benutzer des Netzes senden	7-4
Den Namen eines Systems feststellen	7-4
Zugriffsmöglichkeiten des Systems feststellen	7-5
Post an Benutzer in einem anderen Netz senden	7-5
Post senden unter Verwendung einer zentralen Namens- und Adressdatenbank	7-5
Post senden über Domännennamen	7-6
Post senden über eine BNU- oder UUCP-Verbindung	7-7
Adressierung mit einer UUCP-Verbindung auf dem eigenen System	7-7
Adressierung mit einer UUCP-Verbindung auf einem anderen System	7-8
Der Editor für elektronische Post	7-10
Einführung in den Editor für elektronische Post	7-10
Editor für elektronische Post: eine Nachricht anzeigen oder ändern	7-11
Editor für elektronische Post: Kopffelder ändern oder hinzufügen	7-11
Post empfangen und bearbeiten	7-13
Post bearbeiten	7-13
Post weiterleiten	7-13
Der elektronische Briefkasten und die Postordner	7-14
Der elektronische Systembriefkasten	7-14
Der persönliche elektronische Briefkasten	7-14
Die Datei dead.letter	7-14
Ordner	7-15
Unterbefehle des elektronischen Briefkastens im Postprogramm	7-15
Eine Nachrichtenliste angeben	7-15
Datei- und Ordnernamen angeben	7-16
Das Postprogramm einstellen	7-18
Postoptionen einstellen	7-18
Optionen aktivieren und inaktivieren	7-19
Die Nachrichtenanzeige im Postprogramm ändern	7-19
In Anzeigen blättern	7-20
Art der angezeigten Informationen steuern	7-21
Den Löschbefehl mit dem Druckbefehl kombinieren	7-22
Post senden	7-23
Voraussetzungen	7-23
Erforderliche Schritte	7-23
Verwendung von Unterbefehlen zum Senden von Post	7-25
Voraussetzungen	7-25
Erforderliche Schritte	7-25
Kopffelder in einer Nachricht der elektronischen Post ändern und hinzufügen	7-28
Voraussetzungen	7-28
Erforderliche Schritte	7-28

Rechtschreibprüfung beim Editor für elektronische Post	7-31
Voraussetzungen	7-31
Erforderliche Schritte	7-31
Eine Nachricht im Editor für elektronische Post neu formatieren	7-32
Voraussetzungen	7-32
Erforderliche Schritte	7-32
Post empfangen und lesen	7-33
Voraussetzungen	7-33
Erforderliche Schritte	7-33
Post beantworten	7-35
Voraussetzungen	7-35
Erforderliche Schritte	7-35
Den Inhalt eines elektronischen Briefkastens anzeigen	7-37
Voraussetzungen	7-37
Erforderliche Schritte	7-37
Informationen zu ausgewählten Nachrichten anzeigen	7-38
Voraussetzungen	7-38
Erforderliche Schritte	7-38
Eine Nachricht im elektronischen Briefkasten editieren	7-40
Voraussetzungen	7-40
Erforderliche Schritte	7-40
Post speichern	7-41
Voraussetzungen	7-41
Erforderliche Schritte	7-41
Ordner im elektronischen Briefkasten verwenden	7-43
Voraussetzungen	7-43
Erforderliche Schritte	7-43
Post weiterleiten	7-45
Voraussetzungen	7-45
Erforderliche Schritte	7-45
Urlaubsnachricht senden	7-46
Voraussetzungen	7-46
Erforderliche Schritte	7-46
Geheime Post senden und empfangen	7-47
Voraussetzungen	7-47
Erforderliche Schritte	7-47
Das Postprogramm anpassen	7-48
Voraussetzungen	7-48
Erforderliche Schritte	7-48
Alias-Namen oder Verteiler erstellen	7-50
Voraussetzungen	7-50
Erforderliche Schritte	7-50
Definierte Alias-Namen und Verteiler bestimmen	7-51
Voraussetzungen	7-51
Erforderliche Schritte	7-51
Bedienerrführung für Betrefffeld ändern	7-52
Voraussetzungen	7-52
Erforderliche Schritte	7-52
Bedienerrführung für Kopienfeld ändern	7-53
Voraussetzungen	7-53
Erforderliche Schritte	7-53

Texteditor zur Eingabe von Nachrichten wechseln	7-54
Voraussetzungen	7-54
Erforderliche Schritte	7-54
Kapitel 8. Online-Informationen und Hilfetext	8-1
Ein neuer Zugriff auf Dokumentation	8-2
Informationen in der Datenbank	8-3
Methode für das Abrufen von Informationen festlegen	8-4
Neue Funktionen in InfoExplorer Version 1.2.0	8-9
Navigationsfunktionen	8-9
Standardwerteanpassung	8-11
Unterstützung mehrerer Bibliotheken	8-11
Druckoptionen	8-11
InfoExplorer-Grafikschnittstelle	8-12
Hilfetextkennungen in InfoExplorer	8-13
Erste Schritte (InfoExplorer-ASCII-Schnittstelle)	8-14
InfoExplorer starten	8-14
Anzeigearten erkennen	8-15
Hypertextverbindung auswählen	8-17
In Anzeigentext blättern	8-18
Menüoption auswählen	8-18
Zu einer vorherigen Position zurückkehren	8-19
Hilfetext zu einer Anzeige aufrufen	8-20
Informationen suchen	8-20
Liste der Hilfetext aufrufen	8-23
InfoExplorer verlassen	8-24
InfoExplorer-Funktionen selbständig erarbeiten	8-24
Erste Schritte (InfoExplorer-Grafikschnittstelle)	8-25
InfoExplorer starten	8-25
Fenstertypen	8-26
Hypertextverbindung auswählen	8-28
Knopf auswählen	8-28
Menüoption auswählen	8-29
Informationen durchblättern	8-29
Lesefenster halten und freigeben	8-30
Zu einer vorherigen Position zurückkehren	8-31
Informationen suchen	8-32
Hilfetext aufrufen	8-34
InfoExplorer verlassen	8-35
InfoExplorer-Funktionen selbständig erarbeiten	8-36
Informationen über Befehle anzeigen	8-37
Erforderliche Schritte	8-37
Elektronische Handbuchseiten (Manual Pages) anzeigen	8-38
Beispiel	8-38
Kapitel 9. Editoren und Textformatierung	9-1
Überblick über Editoren	9-2
Überblick über die Textformatierung	9-4
Kapitel 10. Problemlösung und Hardwareinformationen	10-1
Problemlösung	10-2
Allgemeine Symptome	10-2

Spezifische Fehler	10-2
Hilfsprogramme zur Problemlösung	10-2
Grundlegende Informationen	10-3
Hilfe zu Fehlermeldungen	10-4
Informationen zu Fehlermeldungen aufrufen	10-4
Hardwareinformationen	10-5
Anhang A. Korn-Shell	A-1
Verwendung der Korn-Shell	A-1
Funktionen der Korn-Shell	A-1
Shell-Umgebung	A-1
Arbeiten mit der Shell	A-2
Koprozeßeinrichtung	A-3
Verwendung von Korn-Shell-Befehlen	A-5
Befehlsinterpretation in der Korn-Shell	A-5
Quotierungszeichen	A-6
Zusammengesetzte Befehle in der Korn-Shell	A-7
Reservierte Wörter	A-8
Sonderbefehle in der Korn-Shell	A-11
Liste der Sonderbefehle	A-12
Das Befehlsprotokoll in der Korn-Shell	A-22
Befehlsprotokollsubstitution	A-22
Arithmetische Berechnungen in der Korn-Shell	A-23
Befehlssubstitution in der Korn-Shell	A-24
Interpretation von Leerzeichen	A-25
Befehls-Alias-Namenprüfung in der Korn-Shell	A-26
Alias-Namen mit absoluter Pfadangabe	A-26
Tildensubstitution	A-27
Parametersubstitution in der Korn-Shell	A-28
Interpretation von Leerzeichen	A-33
Dateinamenssubstitution in der Korn-Shell	A-33
Ein- und Ausgabe in der Korn-Shell umleiten	A-35
Jobsteuerung in der Korn-Shell	A-37
Befehlszeileneditierung in der Korn-Shell	A-38
Der Editiermodus emacs	A-38
Editiermodus vi	A-41
Anhang B. Bourne-Shell	B-1
Verwendung der Bourne-Shell	B-1
Die Shell-Umgebung	B-1
Verwendung der eingeschränkten Shell	B-3
Verwendung von Bourne-Shell-Befehlen	B-4
Quotierungszeichen	B-5
Signalverarbeitung	B-5
Sonderbefehle in der Bourne-Shell	B-7
Liste der Sonderbefehle	B-7
Befehlssubstitution in der Bourne-Shell	B-12
Variablensubstitution in der Bourne-Shell	B-13
Benutzerdefinierte Variablen	B-13
Bedingte Substitution	B-17
Positionsparameter	B-18

Dateinamenssubstitution in der Bourne-Shell	B-19
Zeichenklassen	B-20
Ein- und Ausgabe in der Bourne-Shell umleiten	B-21
Standardein- und -ausgabe	B-21
Diagnose- und sonstige Ausgabe	B-22
Befehlszeileneingabedokument ("bin da"-Dokument)	B-22
Ein- und Ausgabeumadressierung mit Dateideskriptoren	B-23
Zusammenfassung der Umadressierungsoptionen	B-23
Anhang C. C-Shell	C-1
Verwendung der C-Shell	C-1
Beschränkungen der C-Shell	C-1
Signalverarbeitung	C-1
Verwendung von C-Shell-Befehlen	C-2
Einführung in Shell-Befehle	C-2
Ausführung nicht integrierter Befehle	C-2
Integrierte Befehle in der C-Shell	C-3
Liste der integrierten C-Shell-Befehle	C-3
Ausdrücke	C-12
Befehlssubstitution in der C-Shell	C-13
Protokollsubstitution in der C-Shell	C-14
Protokollisten	C-14
Ereignisspezifikation	C-15
Die Verwendung von Anführungszeichen und halben Anführungszeichen	C-17
Alias-Namenssubstitution in der C-Shell	C-18
Variablensubstitution in der C-Shell	C-19
Vordefinierte Variablen und Umgebungsvariablen in der C-Shell	C-21
Dateinamenssubstitution in der C-Shell	C-24
Dateinamenerweiterung	C-24
Dateinamenabkürzung	C-25
Zeichenklassen	C-25
Ein- und Ausgabe in der C-Shell umleiten	C-27
Steuerungsfluß	C-28
Jobsteuerung in der C-Shell	C-29
Index	X-1

Kapitel 1. Lerntext: Erste Schritte

Dieses Kapitel führt den Einsteiger schrittweise in das Betriebssystem ein. Es enthält Prozeduren zum Starten des Systems, zum An- und Abmelden sowie zum Ausführen einfacher Befehle. Dem Benutzer wird erklärt, wo er Hilfe findet und welche verschiedenen Benutzerschnittstellen es gibt. Außerdem enthält dieses Kapitel eine Tabelle mit häufig verwendeten Tasten bzw. Tastenfunktionen.

Dieses Kapitel setzt voraus, daß das Betriebssystem bereits installiert ist und der Systemmanager (Systemadministrator) eine Benutzer-ID und ein Kennwort für den Benutzer festgelegt hat.

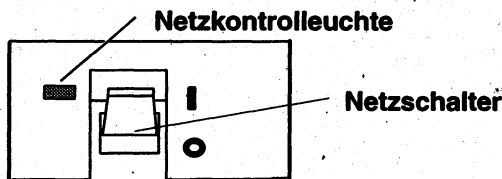
Die in diesem Lerntext verwendeten Abbildungen stimmen möglicherweise je nach verwendeter Hardware nicht mit dem System überein.

Folgende Schritte werden in diesem Kapitel behandelt:

- 1. Schritt: System einschalten
- 2. Schritt: System starten
- 3. Schritt: Bildschirm einschalten
- 4. Schritt: Anmelden
- 5. Schritt: Benutzerschnittstelle festlegen
- 6. Schritt: Kennwort erstellen und ändern
- 7. Schritt: Spezielle Tastenfunktionen benutzen
- 8. Schritt: Befehle eingeben
- 9. Schritt: Abmelden

1. Schritt: System einschalten

Das System wird eingeschaltet, indem der Netzschalter auf "Ein" (I) geschaltet wird. Dies wird durch das Leuchten der Netzkontrolleuchte angezeigt.



Ja Weiter mit Schritt 3.

Nein Weiter mit Schritt 2.

2. Schritt: System starten

1. Alle angeschlossenen Einheiten einschalten, d. h. den Netzschalter auf die Position "Ein" setzen.
2. Den Moduswählschalter an der Systemeinheit auf "Normal" stellen.



Mit dem Moduswählschalter wird der Pfad zum einleitenden Programmladen (IPL) gewählt. Steht der Schalter auf "Normal", wird das Betriebssystem geladen. Näheres über den Moduswählschalter steht im *Bedienerhandbuch* der Systemeinheit.

3. Zum Starten der Systemeinheit den Netzschalter auf die Position "On" (I) setzen.

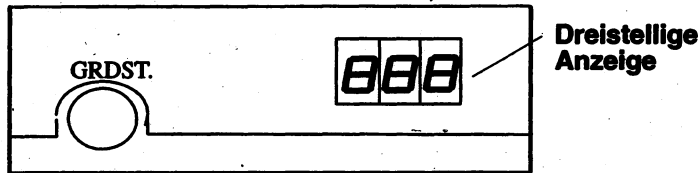
Nach dem Einschalten der Systemeinheit leuchtet die Netzkontrolleuchte, und das System führt einen internen Selbsttest (POST) durch.

Leuchtet die Netzkontrolleuchte nach dem Einschalten des Netzschalters nicht auf, überprüfen, ob das Netzkabel an der Rückseite der Systemeinheit an die Wandsteckdose angeschlossen ist. Ist das Netzkabel ordnungsgemäß angeschlossen, das Handbuch *IBM RISC System/6000 Handbuch Problemlösung* hinzuziehen.

4. Nach dem Einschalten der Systemeinheit erscheinen Zahlen auf der dreistelligen Anzeige.

Die dreistellige Anzeige auf der Bedienerkonsole hat folgende Funktionen:

- Ablaufverfolgung beim Selbsttest der Systemeinheit und beim Konfigurationsprogramm.
- Anzeige von Codes, wenn der Betrieb des Systems aufgrund einer Fehlerbedingung abgebrochen wird.
- Anzeige von Diagnoseprogrammnachrichten.

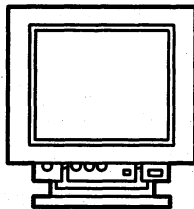


Während des Selbsttests beim Einschalten (POST) werden Zahlen angezeigt, die den Verlauf des Tests wiedergeben. Werden beim Selbsttest keine Fehler festgestellt, ist die Anzeige anschließend leer.

Wird ein Fehler festgestellt, der behoben werden muß, wird der Test abgebrochen und eine dreistellige Fehlernummer zur Kennzeichnung des Fehlers angezeigt. Näheres über Fehlercodes und Fehlerbehebung steht im Handbuch *IBM RISC System/6000 Handbuch Problemlösung* bzw. ist über den Systemmanager zu erfragen.

3. Schritt: Bildschirm einschalten

a. Ist der Bildschirm eingeschaltet?



Ja Weiter mit Schritt 3b.

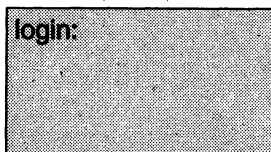
Nein Bildschirm einschalten und einstellen, sofern erforderlich.

Der Bildschirm ist eingeschaltet, wenn die Netzkontrolleuchte leuchtet. Bleibt der Bildschirm nach dem Einschalten dunkel, die Helligkeit mit Hilfe des Helligkeitsreglers einstellen. Der Netzschalter und die Regler sind normalerweise unterhalb der Anzeige angebracht.

Einige Bildschirme besitzen auch eine Kippvorrichtung, mit deren Hilfe sich die Bildschirmneigung einstellen läßt. Zum Arbeiten mit dem System kann der Benutzer mit den entsprechenden Reglern die Anzeige an seine Anforderungen anpassen. Näheres dazu enthält das *Bedienerhandbuch* des Bildschirms.

b. Ist der Anmeldedialog angezeigt?

Ein Anmeldedialog ähnlich wie im folgenden Bild sollte auf der Anzeige zu sehen sein:



Ja Weiter mit Schritt 4.

Nein Den Systemmanager konsultieren oder in den Schritten im Abschnitt "Fehlerbestimmung" des Handbuchs *IBM RISC System/6000 Handbuch Problemlösung* nachlesen

Hinweis: Bei Erscheinen einer Fehlermeldung im Abschnitt "Hilfe bei Fehlermeldungen" auf Seite 10-4 nachlesen.

4. Schritt: Anmelden

Damit der Benutzer auf das Betriebssystem zugreifen kann, muß er sich dem Betriebssystem durch Eingabe der Benutzer-ID (Anmeldename) als zugelassener Benutzer zu erkennen geben. Eine Benutzer-ID ist eigentlich ein Name, mit dem sich der Benutzer dem System zu erkennen gibt. Der Zugriff auf Systemfunktionen wird durch die Benutzer-ID und durch die Systemgruppen, denen der Benutzer zugeordnet ist, gesteuert. Das Anmelden geschieht folgendermaßen:

1. Benutzer-ID (Anmeldename) beim Anmeldedialog eingeben und die Eingabetaste drücken.
2. Wenn die Eingabeaufforderung `password:` (Kennwort) angezeigt wird, das Kennwort eingeben und die Eingabetaste drücken. (Das Kennwort wird nicht angezeigt.)

Falls die Benutzer-ID oder das Kennwort nicht bekannt ist, mit dem Systemmanager abklären.

Was nach dem Anmelden als nächstes geschieht, hängt davon ab, wie das Betriebssystem konfiguriert ist. Es kann entweder eine Befehlszeilen- oder eine Grafikschnittstelle auf der Anzeige erscheinen.

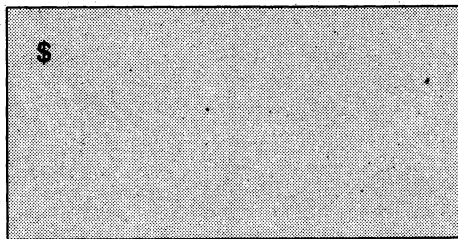
5. Schritt: Benutzerschnittstelle festlegen

Wie mit dem System gearbeitet wird, hängt von der Benutzerschnittstelle ab. Nach dem Anmelden wird eine der folgenden Schnittstellen auf der Anzeige angezeigt:

- Befehlszeilenschnittstelle
- Grafikschnittstelle.

Befehlszeilenschnittstelle

Die meisten Benutzer sind mit der Befehlszeilenschnittstelle vertraut (die manchmal auch als nichtgrafische oder Shell-Schnittstelle bezeichnet wird). Auf der Anzeige ist außer der Eingabeaufforderung in der Regel nichts zu sehen. Befehle werden mit Hilfe der Tastatur bei der Eingabeaufforderung eingegeben. (Eine Eingabeaufforderung kann aus einem Symbol oder einer Nachricht bestehen.)



Anzeige mit Befehlszeile

Die in den Schritten 6 und 8 dieses Lern textes beschriebenen Befehle werden bei der Eingabeaufforderung eingegeben.

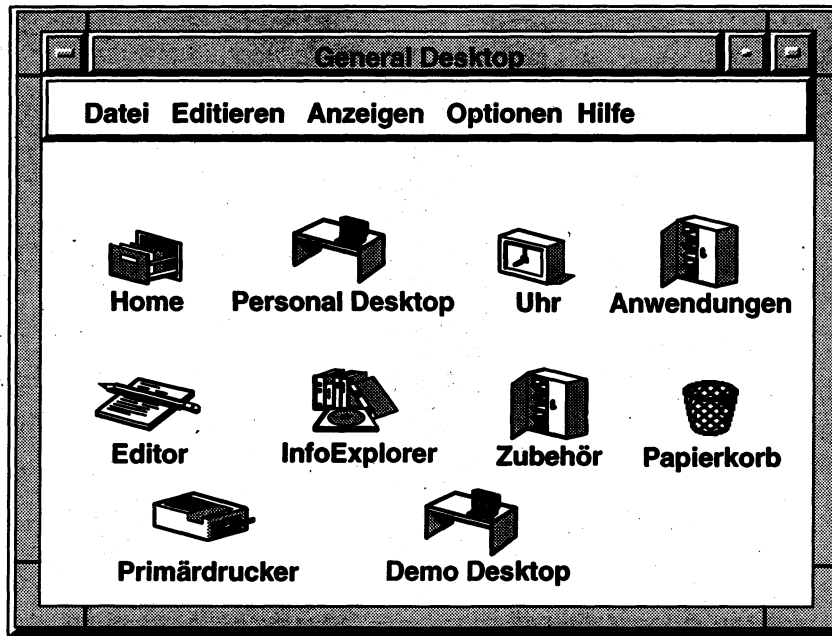
Grafikschnittstelle

AIXwindows Window Manager und AIXwindows Desktop sind die Grafikschnittstelle zwischen dem Benutzer und seiner Datenstation.

AIXwindows Desktop ist eine grafische Benutzerschnittstelle, die dem Benutzer die Anzeige als "elektronischen Schreibtisch" erscheinen läßt. Das Desktop zeigt Symbole für verschiedene Funktionen (wie z. B. Dateien, Verzeichnisse und Programme) an, die der

Benutzer mit Hilfe der Maus auswählen und bewegen kann. Durch das Öffnen von Fenstern auf dem Desktop kann der Benutzer mehrere Aktivitäten gleichzeitig anzeigen und einfach von einer zur anderen wechseln.

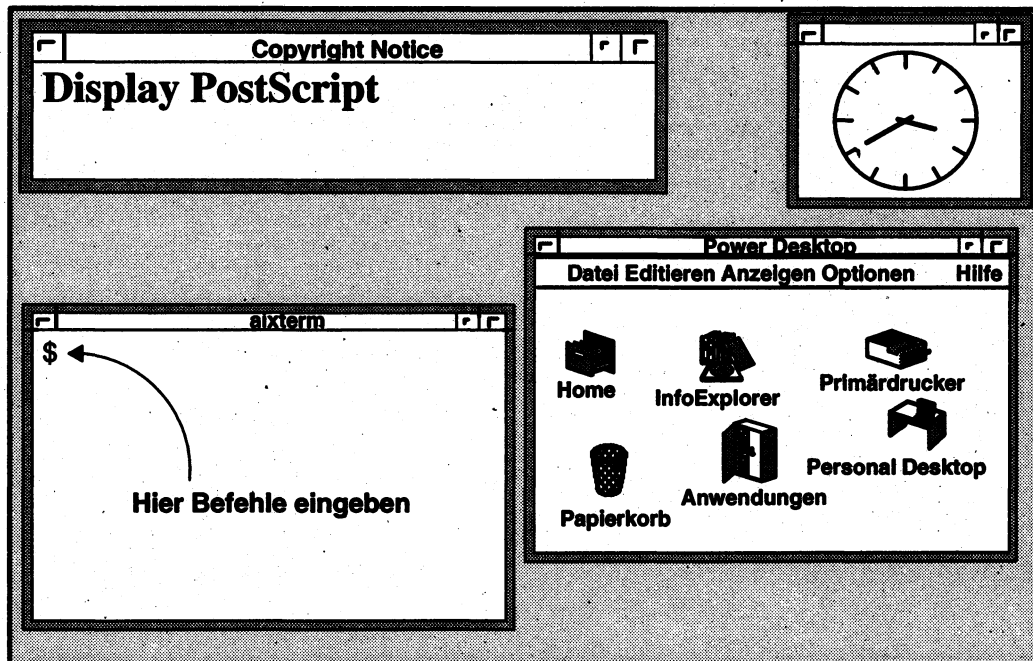
Das Desktop kann entweder als ganze Anzeige angezeigt werden, wie in der folgenden Abbildung, oder zusammen mit weiteren Fenstern, wie in der Abbildung "Anzeige Desktop und Window Manager".



Anzeige General Desktop

Die Grafikschnittstelle AIXwindows Desktop ist allein nicht lauffähig. Sie ist abhängig vom AIXwindows Window Manager, der wiederum vom X Window System abhängig ist.

Der Window Manager stellt Werkzeuge für die Arbeit mit Fenstern zur Verfügung. Fenster sind definierte Arbeitsbereiche in der Anzeige, in denen Programme Informationen anzeigen und Daten eingegeben werden können. Viele Anwendungen, die für Computer erhältlich sind, basieren auf dem Window Manager, so daß die Fertigkeiten, die beim Arbeiten mit dem Window Manger erlernt werden, auch für andere Anwendungen genutzt werden können.



Anzeige Desktop und Window Manager

Die in den Schritten 6 und 8 dieses Lerntextes beschriebenen Befehle werden bei der Eingabeaufforderung im Fenster **aixterm** eingegeben.

Weitere Informationen über die Abhängigkeit zwischen dem X Window System, dem AIXwindows Window Manager und dem Desktop stehen im Abschnitt "AIXwindows Window Manager und AIXwindowsDesktop" auf Seite KEIN MERKER. Nähere Informationen und Prozeduren zum Handhaben von Fenstern und Arbeiten auf dem Desktop stehen in den Kapiteln 4 und 5.

6. Schritt: Kennwort erstellen und ändern

Zusätzlich zur Benutzer-ID sollte der Benutzer über ein Kennwort verfügen. Ein Kennwort gibt dem System die Identität des Benutzers zu erkennen und schützt die Daten vor unerlaubtem Zugriff. Mit Hilfe des Befehls **passwd** kann ein Kennwort erstellt oder geändert werden.

Folgendes eingeben, um das Kennwort zu ändern:

```
passwd
```

Das alte Kennwort (falls vorhanden) wird angefordert. Für das neue Kennwort erfolgt zweimal die Eingabeaufforderung.

7. Schritt: Spezielle Tastenfunktionen benutzen

Bestimmte Tasten oder Tastenkombinationen erleichtern das Arbeiten in der Befehlszeile. Um z. B. die Ausgabe zu stoppen und zur Eingabeaufforderung zurückzukehren, die Tastenfolge Strg-C drücken. Zum Eingeben der Tastenfolge die Steuertaste gedrückt halten und die Taste C drücken.

Es folgt eine Liste hilfreicher Tastenfolgen und ihrer Funktionen:

Funktionen	Tastenfolge
Unterbrechen	Strg-C
Dateiende	Strg-D
Verlassen	Strg-\
Abbruch	Esc
Vorheriges Zeichen löschen	Rückschritt
Vorheriges Wort löschen	Strg-W
Zeile löschen	Strg-U
Ausgabe unterbrechen	Strg-S
Ausgabe wiederaufnehmen	Strg-Q
Nächste virtuelle Datenstation	Alt-Ausf
Vorherige virtuelle Datenstation	Umsch.-Ausf

8. Schritt: Befehle eingeben

Dem Betriebssystem muß mit Hilfe von Befehlen mitgeteilt werden, welchen Verarbeitungsvorgang es durchführen soll. Ein Befehl ist eine Anforderung zum Ausführen einer Operation oder eines Programms. Wenn ein Befehl eingegeben wird, wird er von einem Befehls-Interpreter (auch "Shell" genannt) gelesen und anschließend verarbeitet.

Einige Befehle können einfach als Wörter eingegeben werden, andere Befehle müssen durch Optionen und Parameter ergänzt werden. Jeder Befehl besitzt eine Syntax, die die erforderlichen und möglichen Optionen, Dateien und Parameter bestimmt. Näheres über Befehle und das Lesen von Befehlssyntaxdiagrammen steht im Abschnitt "Überblick über Befehle" auf Seite 2-41.

Am besten macht man sich mit dem Betriebssystem vertraut, indem man einige grundlegende Befehle eingibt. Im folgenden werden einige Beispiele dafür angeführt.

Benutzer-ID anzeigen

Der Befehl **whoami** zeigt die Benutzer-ID (Anmeldename) an.

Folgendes bei der Eingabeaufforderung eingeben:

```
whoami
```

Die Benutzer-ID wird in der nächsten Zeile angezeigt.

```
$ whoami
denise
$
```

Liste der angemeldeten Benutzer anzeigen

Der Befehl **who** zeigt Informationen über alle Benutzer an, die gerade am lokalen System angemeldet sind. Die Informationen beinhalten Benutzer-ID, Datenstationsname sowie Datum und Uhrzeit der Anmeldung.

Folgendes bei der Eingabeaufforderung eingeben:

```
who
```

Folgende oder ähnliche Informationen sind zu sehen:

```
$who
marta  hft/0  Jul 26 7:15
frank  pts4   Jul 26 8:00
$
```

Durch Hinzufügen der Optionen **am** und **i** zum Befehl **who** können die Benutzer-ID und der Datenstationsname des Benutzers sowie Datum und Uhrzeit seiner Anmeldung angezeigt werden.

Folgendes bei der Eingabeaufforderung eingeben:

```
who am i
```

Folgende oder ähnliche Informationen sind zu sehen:

```
$ who am i
denise console Jul 26 7:30
$
```

Eine Datei erstellen

Normalerweise wird der Befehl **cat** verwendet, um den Inhalt einer Datei anzuzeigen; er kann jedoch auch zum Erstellen einer Datei verwendet werden. Im folgenden Beispiel wird eine Datei mit dem Namen `Testdatei` erstellt und am Bildschirm angezeigt.

1. Folgendes bei der Eingabeaufforderung eingeben und anschließend die Eingabetaste drücken:

```
cat >Testdatei
```

```
$cat >Testdatei
-
```

2. Folgenden Satz eingeben und anschließend die Eingabetaste drücken:

```
Das ist die erste Zeile in der Datei.
```

```
$cat >Testdatei
Das ist die erste Zeile in der Datei.
-
```

3. Folgenden Satz eingeben und anschließend die Eingabetaste drücken:

Das ist die letzte Zeile in der Datei.

```
$cat >Testdatei  
Das ist die erste Zeile in der Datei.  
Das ist die letzte Zeile in der Datei.  
_
```

4. Die Tastenfolge Strg-D drücken. (Steuertaste gedrückt halten und Taste D drücken.)

```
$cat >Testdatei.  
Das ist die erste Zeile in der Datei.  
Das ist die letzte Zeile in der Datei.  
$_  
_
```

Mit der Tastenfolge Strg-D wird die Datei beendet. Anschließend ist wieder die Eingabeaufforderung zu sehen.

5. Um den Inhalt der neuen Datei (Testdatei) anzuzeigen, folgendes bei der Eingabeaufforderung eingeben und anschließend die Eingabetaste drücken:

```
cat Testdatei
```

```
$cat >Testdatei
Das ist die erste Zeile in der Datei.
Das ist die letzte Zeile in der Datei.
$cat Testdatei
Das ist die erste Zeile in der Datei.
Das ist die letzte Zeile in der Datei.
$_
```

Hinweis: Das Zeichen > (Größer-als-Zeichen) in der ersten Befehlszeile ist ein E/A-Symbol, das die darauffolgenden zwei Textzeilen in die Datei Testdatei überträgt. Wird der Befehl cat nun ohne > eingegeben, wird der Inhalt der Datei Testdatei angezeigt. Näheres zu Eingabe- und Ausgabesymbolen steht im Abschnitt "Überblick über Eingabe und Ausgabe" auf Seite 2-28.

9. Schritt Abmelden

- Zum Abmelden aus der Grafikschnittstelle:

Tastenfolge Strg-Alt-Rückschritt drücken.

Damit wird die Grafikschnittstelle beendet. Erscheint eine Befehlszeile auf der Anzeige anstelle der Eingabeaufforderung login:, die Befehlszeile durch Verwenden der Tastenfolge Strg-D oder durch Eingabe von exit und anschließendes Drücken der Eingabetaste verlassen.

- Zum Abmelden aus der Befehlszeilenschnittstelle:

Tastenfolge Strg-D drücken oder exit eingeben und anschließend die Eingabetaste drücken.

Nach dem Abmelden zeigt das System die Eingabeaufforderung login: an.

Achtung: Das System nicht ausschalten, da sonst alle auf dem System laufenden Prozesse beendet werden. Wenn weitere Benutzer am System arbeiten oder wenn Jobs im Hintergrund aktiv sind, können Daten verlorengehen. Vor dem Ausschalten des Systems einen ordentlichen Systemabschluß durchführen.

Empfohlene Dokumentation

Bei Problemen mit der Hard- oder Software im *IBM RISC System/6000 Handbuch Problemlösung* nachsehen. Weitere Informationen über die im *IBM RISC System/6000 Handbuch Problemlösung* behandelten Themen stehen im Abschnitt "Problemlösung" auf Seite 10-2.

Weitere Informationen über das Verwalten von Fenstern stehen im Kapitel "AIXwindows Window Manager" auf Seite KEIN MERKER.

Informationen über das Arbeiten auf dem Desktop stehen im Kapitel "AIXwindows Desktop" auf Seite 5-1.

Weitere Informationen über Befehle und das Lesen von Befehlsyntaxdiagrammen stehen im Abschnitt "Überblick über Befehle" auf Seite 2-41.

Kapitel 2. Grundlagen des Betriebssystems

Das Betriebssystem ist die Software, die den Ablauf von Programmen steuert und darüber hinaus Dienstprogramme, wie z. B. Ressourcenzuordnung, Zeitplanung, Eingabe- und Ausgabesteuerung sowie Datenverwaltung beinhalten kann. Das Betriebssystem und einige seiner grundlegenden Funktionen werden in den folgenden Abschnitten vorgestellt:

- Dateisysteme
- Prozesse
- Shells
- Ein- und Ausgabe
- Profile
- An- und Abmelden
- Virtuelle Datenstationen
- Befehle

Weitere Informationen über das System stehen im Handbuch *Dokumentationsüberblick*. Das Handbuch *Dokumentationsüberblick* enthält eine detaillierte Übersicht der zum System erhältlichen Handbücher und Online-Informationen.

Das Handbuch *System Management Guide* enthält Informationen über die Betriebssystemverwaltung.

Das Handbuch *General Programming Concepts* enthält Programmierhinweise zu verschiedenen Abschnitten.

Überblick über Dateisysteme

Ein Dateisystem hat eine hierarchische Struktur (Dateibaum) von Verzeichnissen und Dateien. Diese Struktur ähnelt einem umgekehrten Baum, bei dem sich der Stamm nach unten verzweigt. In diesem Dateibaum werden Daten und Programme in Verzeichnissen gruppiert, wodurch mehrere Verzeichnisse und Dateien gleichzeitig verwaltet werden können.

Bei verschiedenen Aufgaben kann mit einem Dateisystem effizienter gearbeitet werden als jeweils mit den einzelnen Verzeichnissen des Dateisystems. Es ist zum Beispiel möglich, ein gesamtes Dateisystem zu sichern, zu verschieben oder bestimmten Sicherheitsbestimmungen entsprechend zu schützen.

Bei dem Basisdateisystem (JFS – Journaled File System) werden Aufzeichnungstechniken für Datenbanken verwendet, um die strukturelle Konsistenz des Dateisystems zu erhalten. Dadurch werden Beschädigungen des Dateisystems durch abnormale Systembeendigung verhindert.

Ein Dateisystem ist auf einem einzelnen logischen Datenträger gespeichert. Mit dem Befehl **mkfs** (Dateisystem erstellen) oder **smit mkfs** (System Management Interface Tool) wird ein Dateisystem auf einem logischen Datenträger erstellt. Jede Datei bzw. jedes Verzeichnis gehört zu einem Dateisystem auf einem logischen Datenträger.

Ein Dateisystem muß an den Mount-Punkt eines Verzeichnisses angehängt sein, damit darauf zugegriffen werden kann. Wenn Mehrdateiensysteme angehängt werden, wird eine Verzeichnisstruktur erstellt, die das Abbild eines einzelnen Dateisystems zeigt. Es entsteht eine hierarchische Struktur, die von einem einzigen Stamm ausgeht. In dieser Struktur sind die Basisdateisysteme und alle weiteren, vom Benutzer erstellten Dateisysteme enthalten.

Reorganisierter Dateibaum

Bei der Version 3.2 wurde der Standarddateibaum in erheblichem Umfang umstrukturiert, um die Systemverwaltung bei Datenstationen mit oder ohne systemreserviertem Plattenspeicher zu erleichtern.

Bei eigenständigen Systemeinheiten sind die zugehörigen Mount-Punkte standardmäßig in folgenden Dateisystemen enthalten:

/dev/hd1	/home
/dev/hd2	/usr
/dev/hd3	/tmp
/dev/hd4	/
/dev/hd9var	/var

Weitere Informationen zur Struktur des Dateisystems von Version 3.2 befinden sich im Abschnitt "Einführung in den reorganisierten Dateibaum" auf Seite 2-5.

Jedes dieser JFSs befindet sich auf einem separaten logischen Datenträger. Das Betriebssystem hängt sie während der Initialisierung an. Diese Konfiguration des Mehrdateiensystems bewährt sich bei Systemverwaltungsfunktionen wie Sichern, Zurückschreiben und Korrekturen, da ein Teil des Dateibaums separat bearbeitet werden kann.

Mit dem Befehl **mount** kann sowohl auf lokale als auch auf ferne Dateisysteme zugegriffen werden. Somit besteht für das System des Benutzers Lese- und/oder Schreibzugriff auf das Dateisystem. Ein Dateisystem kann normalerweise nur von einem Mitglied der

Systemverwaltungsgruppe an- oder abgehängt werden. Dateisysteme können automatisch angehängt werden, wenn sie in der Datei **/etc/filesystems** definiert sind. Ein lokales oder fernes Dateisystem kann mit dem Befehl **umount** abgehängt werden, sofern kein anderer Benutzer oder Prozeß darauf zugreift.

Dateisystemtypen

Das Betriebssystem Version 3 unterstützt mehrere Dateisystemtypen, unter anderem:

Journaled File System (JFS)

Das Basisdateisystem des Betriebssystems Version 3. Es unterstützt die Semantik aller Dateisysteme.

Network File System (NFS)

Ein Dateisystemtyp, bei dem auf Dateien, die auf fernen Maschinen gespeichert sind, zugegriffen werden kann, als ob sie auf der lokalen Maschine gespeichert wären.

CD-ROM-Dateisystem

Ermöglicht den Zugriff auf den Inhalt einer CD-ROM über normale Dateisystemschnittstellen (Öffnen, Lesen und Schließen).

Eine Reihe von Befehlen können an allen Dateisystemtypen ausgeführt werden. Die Datei **/etc/filesystems** steuert die Liste der Dateisysteme, die mit folgenden Befehlen ausgeführt werden können:

chfs	Ändert die Merkmale eines Dateisystems.
crfs	Fügt ein Dateisystem hinzu.
lsfs	Zeigt die Merkmale eines Dateisystems an.
rmfs	Löscht ein Dateisystem.
mount	Macht ein Dateisystem zur Benutzung verfügbar.

Die nachfolgend aufgelisteten vier Befehle können an virtuellen Dateisystemtypen ausgeführt werden. In der Datei **/etc/vfs** sind Informationen über diese Dateisystemtypen enthalten.

chvfs	Ändert die Merkmale eines Dateisystemtyps.
crvfs	Fügt einen neuen Dateisystemtyp hinzu.
lsvfs	Listet die Merkmale eines Dateisystemtyps auf.
rmvfs	Löscht einen Dateisystemtyp.

Dateisystemverwaltungsaufgaben

Bei einem Dateisystem handelt es sich um eine vollständige Verzeichnisstruktur mit einem Stammverzeichnis, Unterverzeichnissen und Dateien. Dateisysteme sind auf einen einzigen logischen Datenträger beschränkt. Einige der wichtigsten Systemverwaltungsaufgaben beziehen sich zum Beispiel auf Dateisysteme:

- Speicherplatz für Dateisysteme auf logischen Datenträgern zuordnen.
- Dateisysteme erstellen.
- Systembenutzern Speicherplatz in Dateisystemen zur Verfügung stellen.
- Verwendung von Speicherplatz im Dateisystem überwachen.
- Dateisysteme sichern, um beim Auftreten von Systemfehlern oder im Falle eines Systemausfalls den Verlust von Daten zu verhindern.

- Konsistenz von Dateisystemen erhalten.

Bestimmte Befehle dienen in erster Linie der Systemverwaltung. Darunter werden folgende Befehle regelmäßig zum Arbeiten mit Dateisystemen verwendet:

backup	Erstellt eine vollständige Sicherung oder eine Teilsicherung eines Dateisystems.
dd	Kopiert Daten direkt von einer Einheit auf eine andere, um Dateisystemsicherungen zu erstellen.
df	Listet die Anzahl freier und verwendeter Speicherplatzblöcke in einem Dateisystem auf.
fsck	Überprüft Dateisysteme und behebt Inkonsistenzen.
mkfs	Erstellt ein Dateisystem in einer angegebenen Größe auf einem angegebenen logischen Datenträger.
mount	Fügt ein Dateisystem an die systemumfassende Namensstruktur an, so daß auf Dateien und Verzeichnisse in diesem Dateisystem zugegriffen werden kann.
restore	Schreibt Dateien von einer Sicherungskopie zurück.
umount	Entfernt ein Dateisystem aus der systemumfassenden Namensstruktur, so daß auf Dateien und Verzeichnisse in dem Dateisystem nicht mehr zugegriffen werden kann.

Empfohlene Dokumentation

Grundlegende Informationen

Im Kapitel "Einführung in Verzeichnisse" auf Seite 3-8 werden das Arbeiten mit Verzeichnissen und die Verwendung von Pfadnamen beschrieben.

Das Kapitel "Überblick über Mount-Vorgänge" im Handbuch *System Management Guide* beinhaltet Informationen zum Anhängen, zu den Mount-Punkten und automatischen Mounts.

Zugehörige Informationen

Weitere Informationen zur Struktur des Dateisystems von Version 3.2 stehen im Abschnitt "Einführung in den reorganisierten Dateibaum" auf Seite 2-5.

Das Kapitel "Überblick über den Speicherplatz logischer Datenträger" im Handbuch *System Management Guide* enthält Informationen über den Logical Volume Manager, logische Datenträger, physische Datenträger und Datenträgergruppen.

Im Kapitel "Network File System (NFS) Overview for System Management" im Handbuch *System Management Guide* werden NFS-Dämonen, Befehle, Dateien und die Netzverwaltung behandelt.

Die Abschnitte zu den Dateien **filesystems** und **vfs** im Handbuch *Files Reference*.

Die Befehle **backup**, **dd**, **df**, **mkfs**, **mount**, **restore** und **umount** werden bei der Systemverwaltung von Dateisystemen verwendet.

Die Befehle **chfs**, **crfs**, **fsck**, **lsfs** und **rmfs** werden beim Arbeiten mit Dateisystemen verwendet.

Die Befehle **chvfs**, **crvfs**, **lsvfs** und **rmvfs** werden beim Arbeiten mit virtuellen Dateisystemtypen verwendet.

Einführung in den reorganisierten Dateibaum

Mit dem reorganisierten Dateibaum werden Dateien in Verzeichnisse gruppiert, die inhaltlich zusammengehörende Informationen enthalten. Diese Organisation erleichtert das ferne Anhängen von Verzeichnissen und Dateien. Systemadministratoren können diese Verzeichnisse als "Bausteine" zum Aufbau eines eindeutigen Dateibaums für jeden Klienten verwenden, der einzelne Verzeichnisse von einem oder mehreren Servern anhängt.

Das ferne Anhängen von Dateien und Verzeichnissen hat im Vergleich zum Speichern aller Daten auf dem lokalen System folgende Vorteile:

- Weniger Plattenspeicherplatz wird belegt.
- Einfache, zentralisierte Systemverwaltung.
- Eine sicherere Umgebung wird zur Verfügung gestellt.

Bei jedem vorherigen Release-Stand wurde das ferne Anhängen dadurch erschwert, daß verschiedene Arten von Informationen, vor allem Informationen, die sich von System zu System unterscheiden, in denselben Verzeichnissen gespeichert wurden.

Der neue Dateibaum

Beim reorganisierten Dateibaum erhalten die Dateien neue Pfadnamen. Aus Gründen der Kompatibilität mit einem vorherigen Release-Stand bestehen beim neuen Dateibaum symbolische Link-Verbindungen zu den meisten Verzeichnissen und Dateien, die verschoben wurden. Dadurch ist die Reorganisation für die meisten Anwendungen transparent.

Der reorganisierte Dateibaum hat folgende Merkmale:

- Dateien, die von mehreren Systemen mit derselben Hardwarearchitektur gemeinsam verwendet werden können, befinden sich im Dateisystem **/usr**.
- Variable Klientendateien, z. B. SPOOL- und Postdateien, befinden sich im Dateisystem **/var**.
- Architekturunabhängige Textdateien, die von mehreren Systemen gemeinsam verwendet werden können, z. B. Handbuchseiten, befinden sich im Verzeichnis **/usr/share**.
- Das Stammdateisystem **/** enthält für den Systembetrieb wichtige Dateien und Verzeichnisse. Es enthält z. B. ein Einheitenverzeichnis, Programme zum Umladen des Systems sowie Mount-Punkte, an denen Dateisysteme an das Stammdateisystem angehängt werden können.
- Das Dateisystem **/home** ist der Mount-Punkt für Benutzerverzeichnisse.
- Bei einem Server enthält das Verzeichnis **/export** Dateien für den Seitenwechselbereich, Klientenstammdateisysteme (Stammdateisysteme, die nicht von mehreren Systemen verwendet werden können), Speicherauszugs- und Benutzerverzeichnisse, das Verzeichnis **/usr/share** für Klienten ohne Plattenspeicher sowie exportierte **/usr**-Verzeichnisse.

Nähere Informationen zum Inhalt bestimmter Dateisysteme oder Verzeichnisse sind den folgenden Abschnitten zu entnehmen:

- Einführung in das Stammdateisystem **/**
- Einführung in das Dateisystem **/usr**
- Einführung in das Verzeichnis **/usr/share**

- Einführung in das Dateisystem /var
- Einführung in das Verzeichnis /export

Einführung in das Stammdateisystem

Das folgende Diagramm "Reorganisiertes Stammdateisystem" zeigt einige der Unterverzeichnisse des reorganisierten Stammdateisystems:

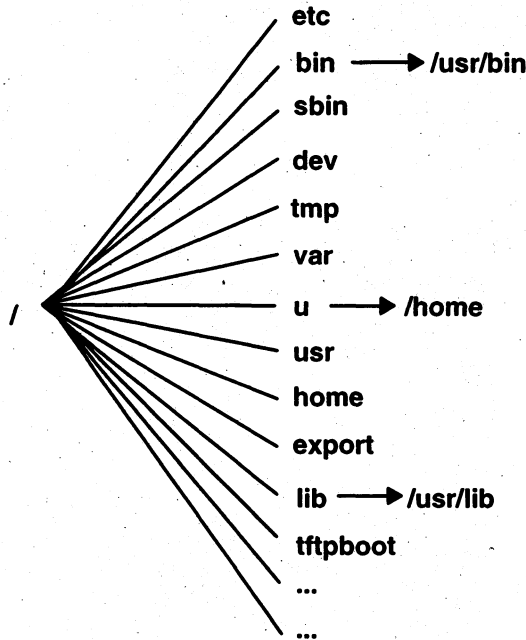


Abbildung 1. Reorganisiertes Stammdateisystem

Das Stammdateisystem ist die oberste Ebene des hierarchischen Dateibaums. Es enthält für den Systembetrieb wichtige Dateien und Verzeichnisse, darunter das Einheitenverzeichnis und Programme für das Umladen des Systems. Ferner enthält das Stammdateisystem Mount-Punkte, an denen Dateisysteme angehängt und damit in die Hierarchie des Stammdateisystems aufgenommen werden können.

Die folgende Liste enthält Informationen zum Inhalt einiger Unterverzeichnisse des Stammdateisystems (/):

/etc Enthält Konfigurationsdateien, die von System zu System unterschiedlich sind, z. B.:

- Datei **/etc/hosts**
- Datei **/etc/passwd**.

Das Verzeichnis **/etc** enthält allgemeine Dateien für die Systemverwaltung. Die meisten Befehle, die sich vorher im Verzeichnis **/etc** befanden, befinden sich bei der neuen Struktur im Verzeichnis **/usr/sbin**. Aus Gründen der Kompatibilität bestehen jedoch symbolische Link-Verbindungen zu den neuen Positionen einiger ausführbarer Dateien, z. B.:

- **/etc/chown** ist eine symbolische Link-Verbindung zu **/usr/bin/chown**.

- **/etc/exportvg** ist eine symbolische Link-Verbindung zu **/usr/sbin/exportvg**.

/bin	ist eine symbolische Link-Verbindung zum Verzeichnis /usr/bin . Bei vorherigen UNIX-Dateisystemen enthielt das Verzeichnis /bin Benutzerbefehle, die sich in der neuen Dateisystemstruktur im Verzeichnis /usr/bin befinden.
/sbin	Enthält Dateien, die zum URLaden des Systems und zum Anhängen des Dateisystems /usr benötigt werden. Die meisten der zum URLaden benötigten Befehle werden dem RAM-Dateisystem des URLadeabbilds entnommen, deshalb befinden sich nur wenige Befehle im Verzeichnis /sbin .
/dev	Enthält Einheitenknoten für Gerätedateien für lokale Einheiten. Außerdem befinden sich im Verzeichnis /dev Gerätedateien für Bandlaufwerke, Drucker, Plattenpartitionen und Datenstationen.
/tmp	Dient als Mount-Punkt für Dateisysteme, die vom System generierte temporäre Dateien enthalten. Dieses Dateisystem ist ein leeres Verzeichnis.
/var	Dient als Mount-Punkt für Dateien, die von System zu System unterschiedlich sind. Dieses Dateisystem ist als Dateisystem konfiguriert, da es Dateien enthält, die möglicherweise größer werden. Weitere Informationen sind dem Abschnitt "Einführung in das Dateisystem /var" auf Seite 2-11 zu entnehmen.
/usr	Enthält Dateien, die sich nicht ändern und von mehreren Systemen gemeinsam verwendet werden können (z. B. ausführbare Dateien und ASCII-Dokumentation). Eigenständige Systeme hängen den Stamm eines separaten lokalen Dateisystems an das Verzeichnis /usr an. Systeme ohne Plattenspeicher oder mit systemreservierter kleiner Platte hängen Verzeichnisse eines fernen Servers über das Dateisystem /usr an. Weitere Informationen sind dem Abschnitt "Einführung in das Dateisystem /usr" auf Seite 2-8 zu entnehmen.
/u	Symbolische Link-Verbindung zum Verzeichnis /home .
/home	Dient als Mount-Punkt für Dateisysteme, die Benutzerverzeichnisse enthalten. Dieses Dateisystem enthält Benutzerdateien und -verzeichnisse. Bei eigenständigen Systemen befindet sich das Verzeichnis /home in einem separaten Dateisystem, dessen Stamm an das Verzeichnis /home des Stammdateisystems angehängt ist. Innerhalb eines Netzes ist es möglich, daß sich auf einem Server Benutzerdateien befinden, auf die Zugriff von mehreren Systemen bestehen sollte. In diesem Fall ist das Verzeichnis /home des Servers an ein lokales Dateisystem /home fern angehängt.
/export	Enthält die Verzeichnisse und Dateien eines Servers, die für ferne Klienten bestimmt sind. Weitere Informationen zum Dateibaum unterhalb dieses Verzeichnisses sind dem Abschnitt "Einführung in das Verzeichnis /export" auf Seite 2-12 zu entnehmen.
/lib	Symbolische Link-Verbindung zum Verzeichnis /usr/lib . Nähere Informationen dazu befinden sich im Abschnitt "Einführung in das Dateisystem /usr" auf Seite 2-8.

/ftptboot

Enthält Urladeabbilder und -informationen für Klienten ohne Plattenspeicher.

Einführung in das Dateisystem /usr

Das Dateisystem **/usr** enthält ausführbare Dateien, die von mehreren Systemen gemeinsam verwendet werden können. Die wichtigsten Unterverzeichnisse des Verzeichnisses **/usr** sind im folgenden Diagramm dargestellt.

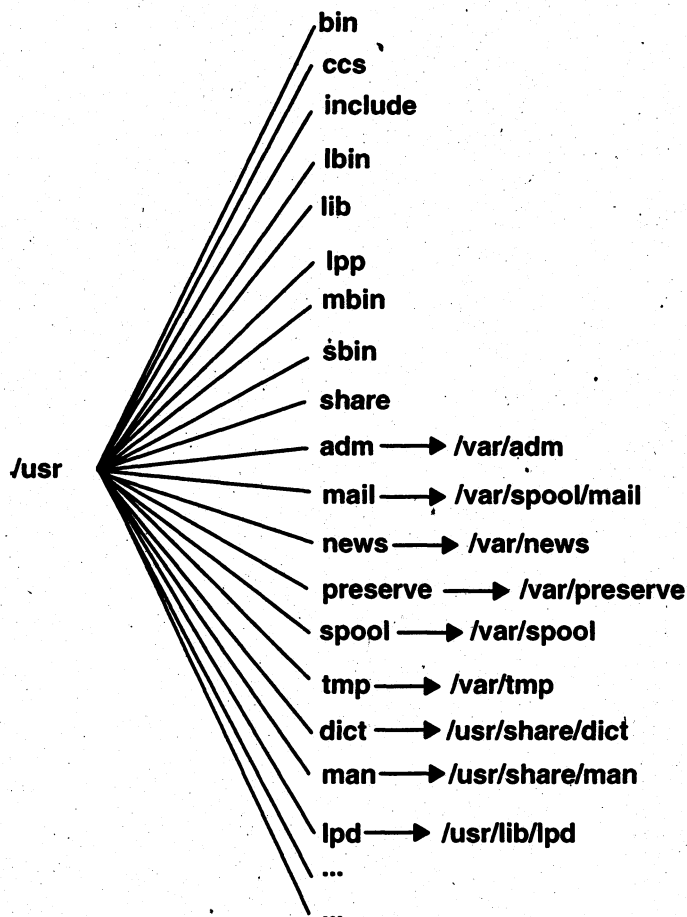


Abbildung 2. Reorganisiertes Dateisystem /usr

Bei eigenständigen Systemen ist das Dateisystem **/usr** ein separates Dateisystem (auf dem logischen Datenträger **/dev/hd2**). Bei Systemen ohne Plattenspeicher oder mit systemreservierter kleiner Platte wird ein Verzeichnis von einem fernen Server mit Leseberechtigung an das lokale Dateisystem **/usr** angehängt. Das Dateisystem **/usr** enthält Lesezugriffsbefehle, Bibliotheken und Daten.

Mit Ausnahme des Inhalts des Verzeichnisses **/usr/share** kann der Inhalt aller Dateien und Verzeichnisse im Dateisystem **/usr** von allen Systemen mit derselben Hardwarearchitektur gemeinsam verwendet werden.

Das Dateisystem **/usr** umfaßt folgende Verzeichnisse:

/usr/bin

Enthält normale Befehle und Shell-Prozeduren, z. B. das Verzeichnis **/usr/bin** enthält die Befehle **ls**, **cat** und **mkdir**.

/usr/ccs	Enthält Compiler-Ausgabeprogramme.
/usr/include	Enthält Anzeigeoptions- oder <i>Kopfdateien</i> .
/usr/sbin	Enthält ausführbare Dateien, die die Ausgabeprogramme für Befehle darstellen.
/usr/lib	Enthält architekturunabhängige Bibliotheken, deren Name das Format lib*.a hat. Das Verzeichnis /lib im Stammdateisystem (/) ist eine symbolische Link-Verbindung zum Verzeichnis /usr/lib , d. h. alle Dateien, die sich bei einem früheren Release-Stand im Verzeichnis /lib befanden, befinden sich jetzt im Verzeichnis /usr/lib . Aus Gründen der Kompatibilität gilt dies auch für einige Nicht-Bibliotheksdateien.
/usr/lpp	Enthält Zusatzprogramme.
/usr/sbin	Enthält die NLS-Versionen (Unterstützung nationaler Sprachelemente) einiger Befehle und Dienstprogramme im Verzeichnis /usr/sbin . Diese Versionen der Befehle und Dienstprogramme bearbeiten Mehrbytezeichen. Weitere Informationen zum NLS sind dem Abschnitt "The National Language Support (NLS) Overview for System Management" im Handbuch <i>System Management Guide</i> zu entnehmen. Hinweis: Dieses Verzeichnis sollte aus der Umgebungsvariablen PATH entfernt werden, wenn keine länderspezifischen Angaben mit Mehrbytezeichen verwendet werden.
/usr/sbin	Enthält Dienstprogramme, die bei der Systemverwaltung verwendet werden, darunter die SMI-Befehle (System Management Interface Tool). Die meisten der Befehle, die sich beim vorherigen Release-Stand im Verzeichnis /etc befanden, befinden sich jetzt im Verzeichnis /usr/sbin .
/usr/share	Enthält Dateien, die von Systemen mit unterschiedlicher Architektur gemeinsam verwendet werden können. Weitere Informationen dazu befinden sich im Abschnitt "Einführung in das Verzeichnis /usr/share " auf Seite 2-10.

Symbolische Link-Verbindungen zum Verzeichnis **/var**

/usr/adm	Symbolische Link-Verbindung zum Verzeichnis /var/adm .
/usr/mail	Symbolische Link-Verbindung zum Verzeichnis /var/spool/mail .
/usr/news	Symbolische Link-Verbindung zum Verzeichnis /var/news .
/usr/preserve	Symbolische Link-Verbindung zum Verzeichnis /var/preserve .
/usr/spool	Symbolische Link-Verbindung zum Verzeichnis /var/spool .
/usr/tmp	Symbolische Link-Verbindung zum Verzeichnis /var/tmp , da das Verzeichnis /usr , auf das Lesezugriff besteht, von mehreren Knoten gemeinsam verwendet werden kann.

Symbolische Link-Verbindungen zu den Verzeichnissen **/usr/share** und **/usr/lib**

/usr/dict	Symbolische Link-Verbindung zum Verzeichnis /usr/share/dict .
/usr/man	Symbolische Link-Verbindung zum Verzeichnis /usr/share/man .
/usr/lpd	Symbolische Link-Verbindung zum Verzeichnis /usr/lib/lpd .

Einführung in das Verzeichnis /usr/share

Die im Verzeichnis **/usr/share** enthaltenen Textdateien können unabhängig von der Architektur von allen Systemen gemeinsam verwendet werden.

In einer gemischten Architekturumgebung hängen Klienten ohne Plattenspeicher in der Regel ein Server-Verzeichnis über das systemeigene Verzeichnis **/usr** an und anschließend ein weiteres Verzeichnis über das Verzeichnis **/usr/share** an. Die Dateien, die unterhalb des Verzeichnisses **/usr/share** gespeichert sind, befinden sich in einem oder mehreren installierbaren Paketen. Aus diesem Grund kann bei einem Knoten das Verzeichnis **/usr** über einen Server zur Verfügung gestellt werden, während sich die anderen erforderlichen Teile des Verzeichnisses **/usr/share** auf dem lokalen System befinden.

Das folgende Diagramm "Reorganisiertes Verzeichnis /usr/share" zeigt einige der im Verzeichnis **/usr/share** enthaltenen Verzeichnisse und Dateien.

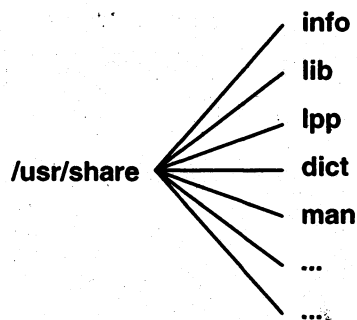


Abbildung 3. Reorganisiertes Verzeichnis /usr/share

Im folgenden werden einige dieser Verzeichnisse beschrieben:

/usr/share/man

Enthält Handbuchseiten, sofern diese geladen wurden.

/usr/share/dict

Enthält das Wörterverzeichnis und seine Indizes.

/usr/share/info

Enthält die InfoExplorer-Datenbankdateien.

/usr/share/lib Enthält architekturunabhängige Datendateien, darunter **terminfo**, **learn**, **tmac**, **me** und **macros**.

/usr/share/lpp Enthält Daten und Informationen über Zusatzprogramme auf dem System.

Einführung in das Dateisystem /var

Achtung: Die Größe des Dateisystems /var kann sich ändern, da es Unterverzeichnisse und Datendateien enthält, die von aktiven Anwendungen, wie z. B. Abrechnung, elektronische Post und Spooler, verwendet werden. Die Standardgröße für das Dateisystem /var ist 4 MB; bei intensiver Nutzung muß es jedoch vergrößert werden (Näheres hierzu befindet sich im Abschnitt "How to Change or Remove a File System" im Handbuch *System Management Guide*).

Folgende speziellen Dateien des Dateisystems /var garantieren periodische Überwachung:

/var/adm/wtmp

/var/adm/ras/errlog

Weitere Dateien aus dem Dateisystem /var, die überwacht werden können:

/var/adm/ras/trcfile Wenn die Ablaufverfolgungseinrichtung eingeschaltet ist.

/var/tmp/snmpd.log Wenn snmpd auf dem System läuft.

Das Diagramm "Reorganisiertes Verzeichnis /var" zeigt einige der im Dateisystem /var enthaltenen Verzeichnisse.

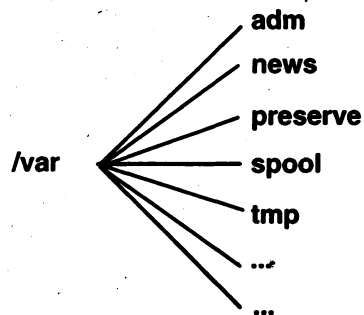


Abbildung 4. Reorganisiertes Verzeichnis /var

Im folgenden werden einige der im Dateisystem /var enthaltenen Verzeichnisse beschrieben:

/var/adm Enthält Anmelde- und Abrechnungsdateien des Systems.

/var/news Enthält Systemnachrichten.

/var/preserve Enthält gesicherte Daten aus unterbrochenen Editiersitzungen; ähnlich dem Verzeichnis **/usr/preserve** in vorherigen Release-Ständen.

/var/spool Enthält Dateien, die von Programmen verarbeitet werden, z. B. elektronische Post; ähnlich dem Verzeichnis **/usr/spool** in vorherigen Release-Ständen.

/var/tmp Enthält temporäre Dateien; ähnlich dem Verzeichnis **/usr/tmp** in vorherigen Release-Ständen. Das Verzeichnis **/usr/tmp** ist jetzt eine symbolische Link-Verbindung zum Verzeichnis **var/temp**.

Einführung in das Verzeichnis /export

Das Verzeichnis **/export** enthält Server-Dateien, die an Klienten, z. B. an Systeme ohne Plattenspeicher, mit systemreservierter Platte oder mit systemreservierter kleiner Platte, exportiert werden. Ein Server kann verschiedene Arten von Plattenspeichern exportieren, z. B. Pakete ausführbarer Programme, Seitenwechselbereich für Klienten ohne Plattenspeicher und Stammdateisysteme für Klienten mit systemreservierter (kleiner) Platte. Diese Art von Plattenspeicher befindet sich im Dateibaum standardmäßig im Verzeichnis **/export**. Das folgende Diagramm "Reorganisiertes Dateisystem /export" zeigt einige der Unterverzeichnisse des Verzeichnisses **/export**.

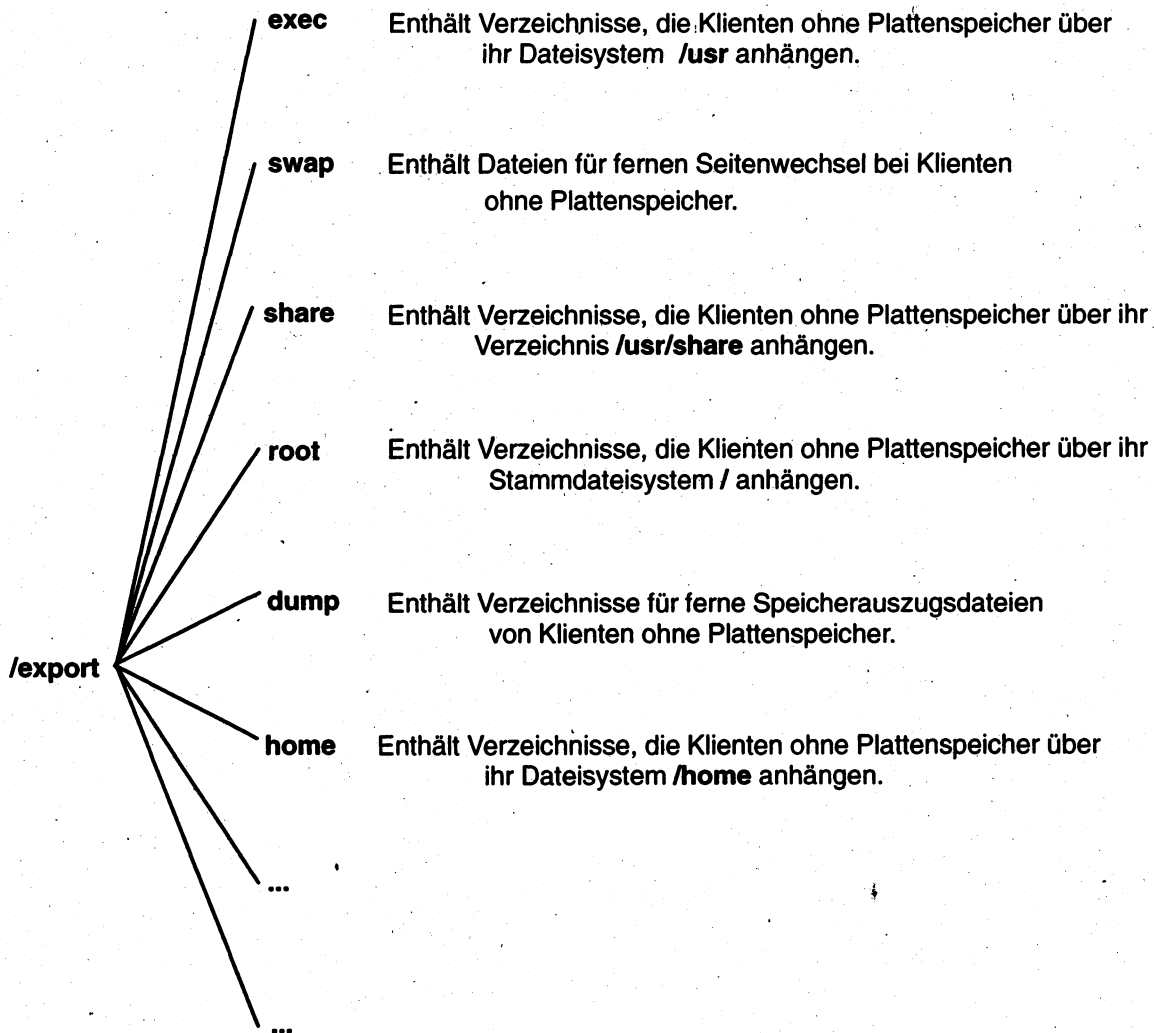


Abbildung 5. Reorganisiertes Dateisystem /export

Die Klientenressourcen für Befehle des Diskless Workstation Management befinden sich standardmäßig im Verzeichnis **/export**.

Die Klientenressourcen befinden sich nur auf dem Server im Verzeichnis **/export**. Da die Klienten die Ressourcen an ihren eigenen Dateibaum anhängen, scheinen diese Ressourcen für die Klienten dann an ihrer normalen Position im Dateibaum zu stehen. Es folgen die wichtigsten Unterverzeichnisse des Verzeichnisses **/export** und ihre entsprechenden Mount-Punkte in einem Klientendateibaum:

Verzeichnis /export/root

Dieses Verzeichnis wird über das Stammdateisystem (/) des Klienten angehängt. Klientenstammverzeichnisse befinden sich standardmäßig im Verzeichnis **/export/root** und erhalten den Host-Namen des Klienten.

Verzeichnis /export/exec oder SPOT (Shared Product Object Tree)

Dieses Verzeichnis wird über das Verzeichnis **/usr** des Klienten angehängt. SPOTs sind Versionen des im Verzeichnis **/export/exec** gespeicherten Dateisystems **/usr** und erhalten Namen, die ihren Release-Stand wiedergeben. Der Standardname ist **RISCAIX**.

Verzeichnis /export/share oder gemeinsam benutztes Verzeichnis (Share)

Dieses Verzeichnis wird über das Verzeichnis **/usr/share** des Klienten angehängt. Es enthält Daten, die von vielen Architekturen gemeinsam verwendet werden können. Die Standardposition ist **/export/share/AIX/usr/share**.

Verzeichnis /export/home

Dieses Verzeichnis wird über das Dateisystem **/home** des Klienten angehängt. Es enthält nach den Host-Namen der Klienten geordnete Benutzerverzeichnisse. Die Standardposition für Klientenbenutzerverzeichnisse ist das Verzeichnis **/export/home/export/home**.

Verzeichnis /export/swap oder Seitenwechselverzeichnis

Bei eigenständigen Systemen oder Systemen mit systemreservierter Platte wird der Seitenwechsel von einer lokalen Platte ermöglicht; bei Klienten ohne Plattenspeicher ermöglicht dies eine Datei auf einem Server. Die Datei wird nach dem Klienten-Host benannt und befindet sich standardmäßig im Verzeichnis **/export/swap**.

Verzeichnis /export/dump

Eigenständige Systeme verwenden eine lokale Platte als Speicherausgangseinheit, Klienten ohne Plattenspeicher verwenden eine Datei auf einem Server. Diese Datei befindet sich in einem Verzeichnis, das nach dem Klienten-Host benannt ist und standardmäßig im Verzeichnis **/export/dump** steht.

Mikrocodeverzeichnis

Dieses Verzeichnis enthält den Mikrocode für physische Einheiten. Seine Standardposition ist **/export/exec/RISCAIX/usr/lib/microcode**.

Überblick über Prozesse

Ein Programm oder Befehl, das/der gerade am System ausgeführt wird, wird als Prozeß bezeichnet. Am Betriebssystem können mehrere Prozesse gleichzeitig ausgeführt werden.

Prozesse sind in einer Vater-/Sohn-Hierarchie aufgegliedert. Ein Prozeß, der durch ein Programm oder einen Befehl gestartet wird, ist ein Vaterprozeß. Ein Sohnprozeß ist das Ergebnis eines Vaterprozesses. Ein Vaterprozeß kann mehrere Sohnprozesse haben, umgekehrt kann ein Sohnprozeß nur einen Vaterprozeß haben.

Jedem Prozeß wird vom System beim Starten eine Prozeßidentifikationsnummer (Prozeß-ID) zugeordnet. Wird derselbe Prozeß mehrmals gestartet, erhält er jedesmal eine andere Prozeß-ID.

Jeder Prozeß, der am System gestartet wird, verwendet einen Teil der verfügbaren Systemressourcen. Werden mehrere Prozesse gleichzeitig ausgeführt, teilt ein in das Betriebssystem integrierter Scheduler auf der Basis festgelegter Prioritäten jedem Prozeß einen Anteil an Computerzeit zu. Diese Prioritäten können über die Befehle **nice** und **renice** geändert werden.

Hinweis: Nur der Systemadministrator kann die Prioritäten erhöhen. Jeder Benutzer kann einem zu startenden Prozeß über den Befehl **nice** und einem bereits gestalteten Prozeß über den Befehl **renice** eine niedrigere Priorität geben.

Prozesse, die von einem Datensichtgerät aus gestartet werden und eine Interaktion des Benutzers mit dem Datensichtgerät erfordern, werden als Vordergrundprozesse bezeichnet. Prozesse, die unabhängig von dem Datensichtgerät ausgeführt werden, auf dem sie gestartet wurden, werden als Hintergrundprozesse bezeichnet. Programme und Befehle laufen standardmäßig als Vordergrundprozesse ab. Hintergrundprozesse werden aufgerufen, indem ein "&" an das Ende der Befehlszeile gestellt wird.

Bei der Initialisierung des Systems für normalen Betrieb wird vom Initialisierungsprozeß ein Shell-Prozeß zum Ausführen der Befehle in der Datei **/etc/rc** erstellt. Diese Befehle bereiten das System zum Anmelden von Benutzern vor. Dämonprozesse werden von der Datei **/etc/rc** aus gestartet.

Sie werden normalerweise beim Starten des Systems gestartet und solange ausgeführt, bis das System abgeschaltet wird. Dämonprozesse werden zur Ausführung von Systemdiensten verwendet und stehen jederzeit mehreren Benutzern und mehreren Tasks zur Verfügung. Dämonprozesse werden vom Benutzer, der als Root angemeldet ist, oder von der Root-Shell gestartet und können nur vom Root wieder beendet werden. Beispielsweise wird vom Prozeß **qdaemon** Zugriff auf Systemressourcen wie z. B. Drucker hergestellt.

Bei einem Geisterprozeß handelt es sich um einen Prozeß, der eine Prozeßidentifikationsnummer hat und in der Prozeßtabelle noch mit dieser Nummer geführt wird, dem aber weder im Benutzeradressbereich noch im Kernel Speicherbereich zugeordnet ist. Geisterprozesse sind beendete Prozesse, die noch so lange weiter existieren, bis der jeweilige Vaterprozeß beendet oder das System abgeschaltet und wieder erneut gestartet wird. Bei einigen Exemplaren kann es möglich sein, einen beendeten Prozeß erneut zu starten.

Prozesse starten

Vordergrundprozesse werden von einem Datensichtgerät aus durch Eingabe eines Programm- oder Befehlsnamens gestartet. Nach dem Starten des Prozesses finden während der Verarbeitungszeit Dialoge zwischen Datensichtgerät und Prozeß statt, bis dieser beendet ist. Andere Interaktionen (z. B. Eingabe eines anderen Befehls) können also nicht durchgeführt werden, bis der Prozeß beendet ist oder vom Benutzer unterbrochen wird.

Ein einzelner Benutzer kann standardmäßig bis zu 40 Prozesse gleichzeitig durchführen. Die Standardeinstellung kann mit Hilfe des Befehls **chdev** geändert werden.

Zum Ausführen eines Programms im Hintergrund den Namen des Befehls oder Programms mit allen gültigen Parametern und Markierungen gefolgt von einem Et-Zeichen (&) eingeben:

Befehlsname&

Programmname&

Während der Ausführung eines Hintergrundprozesses können weitere Aufgaben durch Eingabe der entsprechenden Befehle oder Programme am Datensichtgerät ausgeführt werden.

Im allgemeinen empfiehlt es sich, die Befehle als Hintergrundprozesse auszuführen, deren Ausführung einen längeren Zeitraum in Anspruch nimmt. Das System wird jedoch langsamer, da die Ausführung von Hintergrundprozessen Prozessorkapazität in Anspruch nimmt. Das kann sich unter Umständen störend auswirken, je nachdem, wie langsam das System wird und welche Prozesse außer den Hintergrundprozessen ausgeführt werden.

Hinweis: Über den Befehl **nice** kann ein Benutzer einem von ihm gestarteten Hintergrundprozeß eine niedrigere Priorität zuordnen.

Die meisten Prozesse, auch Hintergrundprozesse, leiten ihre Ausgabe an die Standardausgabe. Die Standardausgabe wird zum Datensichtgerät geleitet, sofern sie nicht umgeleitet wird. Um unerwünschte Arbeitsunterbrechungen aufgrund der Ausgabe von Hintergrundprozessen zu vermeiden, empfiehlt es sich, die Ausgabe von Hintergrundprozessen in eine Datei oder an einen Drucker umzuleiten. Dort ist die Ausgabe jederzeit verfügbar.

Hinweis: In bestimmten Fällen kann ein Prozeß die Ausgabe in einer anderen Reihenfolge generieren, wenn er im Hintergrund statt im Vordergrund ausgeführt wird. Benutzer mit Programmierkenntnissen können mit Hilfe der Unteroutine **fflush** sicherstellen, daß die Ausgabe bei Vordergrund- wie Hintergrundprozessen in der richtigen Reihenfolge erfolgt.

Während der Ausführung eines Hintergrundprozesses kann dessen Status über den Befehl **ps** (Prozeßstatus) abgefragt werden.

Prozesse abbrechen

Mit Hilfe der Unterbrechungstaste (Drücken der Tastenfolge Strg-C oder Strg-Rückschrittaste) kann ein bereits gestarteter Vordergrundprozeß wieder abgebrochen werden. Über den Befehl **stty** kann ermittelt werden, welche Tastenfolge für die Unterbrechungstaste gesetzt ist.

Hinweis: Mit Hilfe der Unterbrechungstaste (Strg-C) können keine Hintergrundprozesse abgebrochen werden. Zum Abbrechen eines Hintergrundprozesses muß der Befehl **kill** verwendet werden.

Die meisten einfachen Befehle werden so schnell verarbeitet, daß sie bereits beendet sind, bevor der Benutzer sie abbrechen kann. Daher wird in den Beispielen im folgenden Abschnitt ein Befehl mit längerer Verarbeitungszeit verwendet, nämlich der Befehl `find / -type f -print`. Über diesen Befehl können die Pfadnamen für alle Dateien am System angezeigt werden. Der Befehl **find** soll hier nicht als solches besprochen werden, sondern dient nur als Beispiel für das Arbeiten mit Prozessen.

Im nachfolgenden Beispiel wird über den Befehl **find** ein Prozeß gestartet. Dieser Prozeß kann nach einigen Sekunden mit der Unterbrechungstaste wie folgt abgebrochen werden:

```
$ find / -type f -print
/usr/sbin/acct/acctcms
/usr/sbin/acct/acctcon1
/usr/sbin/acct/acctcon2
/usr/sbin/acct/acctdisk
/usr/sbin/acct/acctmerg
/usr/sbin/acct/accton
/usr/sbin/acct/acctprc1
/usr/sbin/acct/acctprc2
/usr/sbin/acct/acctwtmp
/usr/sbin/acct/chargefee
/usr/sbin/acct/ckpacct
/usr/sbin/acct/dodisk
/usr/sbin/acct/fwtmp
/usr/sbin/acct/lastlogin
/usr/sbin/acct/monacct
/usr/sbin/acct/nulladm
/usr/sbin/acct/prctmp
/usr/sbin/acct/prdaily
/usr/sbin/acct/prtacct
/usr/sbin/acct/runacct
/usr/sbin/acct/sdisk
/usr/sbin/acct/shutacct INTERRUPT (Ctrl-C)
$ _
```

Die Eingabeaufforderung erscheint wieder in der Anzeige. Nun kann ein anderer Befehl eingegeben werden.

Gelingt es nicht, mit der Unterbrechungstaste einen Vordergrundprozeß abzubrechen, oder soll ein bereits gestarteter Hintergrundprozeß unterbrochen werden, kann der Prozeß mit dem Befehl **kill** abgebrochen werden. Dazu muß jedoch die Prozeß-ID bekannt sein.

Die benötigte Prozeß-ID kann bei Bedarf über den Befehl **ps** angezeigt werden (es erscheint eine Liste der Prozeß-IDs aller Prozesse).

Nachdem die Verarbeitung des Befehls **find** im nachfolgenden Beispiel begonnen hat, wird über den Befehl **ps**, der ohne die Markierung **-p** eingegeben wird, der Status aller Prozesse angezeigt:

```
# find / -type f -print >dir.paths &
[1] 21593
# ps
  PID  TTY  TIME  COMMAND
1627 pts3 0:00 ps
5461 pts3 0:00 ksh
17565 pts3 0:00 -ksh
21593 pts3 0:00 find / -type f -print
# kill 21593
# ps
  PID  TTY  TIME  COMMAND
1627 pts3 0:00 ps
5461 pts3 0:00 ksh
17565 pts3 0:00 -ksh
[1] + 21593 Terminated find / -type f -print > dir.paths #
```

Durch Eingabe des Befehls **kill 21593** wird der Hintergrundprozess **find** abgebrochen. Beim zweiten Eingeben des Befehls **ps** werden keine Statusinformationen über die Prozeß-ID mehr angezeigt. Das System gibt erst bei Eingabe des nächsten Befehls eine Abbruchnachricht aus, sofern es sich bei diesem Befehl nicht um den Befehl **cd** handelt.

Alle aktuellen Prozesse anzeigen

Bei eingeschaltetem System werden stets mehrere Prozesse ausgeführt. Mit dem Befehl `ps` können alle aktiven Prozesse sowie Informationen über diese Prozesse angezeigt werden. Mit dem Befehl `ps` können alle aktiven Prozesse auf folgende Weise angezeigt werden:

- Von der Befehlszeile aus
- Mit der Einrichtung SMIT.

Alle Prozesse von der Befehlszeile aus anzeigen

Der Befehl `ps` verfügt über mehrere Markierungen, mit deren Hilfe sich bestimmen läßt, welche Prozesse aufgelistet werden und welche Informationen über diese Prozesse angezeigt werden. Der Abschnitt zum Befehl `ps` enthält Informationen über die richtige Syntax.

Folgendes eingeben, um alle auf dem System aktiven Prozesse anzuzeigen:

```
ps -ef
```

Alle Prozesse mit der Einrichtung SMIT anzeigen

1. Den SMIT-Direktaufbefehl eingeben:

```
smit process
```

Mit diesem Befehl wird die Anzeige Prozesse angezeigt.

2. Die Option Aktuelle Prozesse anzeigen markieren und die Eingabetaste drücken. SMIT führt dann den Befehl `ps -ef` aus und zeigt seitenweise allen aktiven Prozesse an.

Prozeßstatus anzeigen

Bei eingeschaltetem System werden stets mehrere Prozesse ausgeführt. Mit dem Befehl **ps** können alle aktiven Prozesse sowie Informationen über diese Prozesse angezeigt werden. Mit dem Befehl **ps** können alle aktiven Prozesse auf folgende Weise angezeigt werden:

- Von der Befehlszeile aus
- Mit der Einrichtung SMIT.

Prozeßstatus von der Befehlszeile aus anzeigen

Folgendes eingeben, um den Status aller aktiven Prozesse anzuzeigen:

```
ps gv
```

Der Befehl **ps** verfügt über mehrere Markierungen, mit deren Hilfe sich bestimmen läßt, welche Prozesse aufgelistet werden und welche Informationen über diese Prozesse angezeigt werden. Das Handbuch "Befehlsreferenz Endbenutzerbefehle" enthält die gültige Syntax. Dieses Befehlsformat listet eine Reihe von Statistiken für jeden aktiven Prozeß auf. Folgendes oder Ähnliches wird mit diesem Befehl ausgegeben:

PID	TTY	STAT	TIME	PGIN	SIZE	RSS	LIM	TSIZ	TRS	%CPU	%MEM	COMMAND
0	-	S	0:44	7	8	8	xx	0	0	0.0	0.0	swapper
1	-	S	1:29	518	244	140	xx	21	24	0.1	1.0	/etc/init
514	-	R	1503:36	0	12	8	xx	0	0	0.0	0.0	kproc
771	-	S	1:22	0	16	16	xx	0	0	0.0	0.0	kproc
1028	-	S	0:00	10	16	8	xx	0	0	0.0	0.0	kproc
1503	-	S	0:33	127	16	8	xx	0	0	0.0	0.0	kproc
1679	-	S	1:03	282	192	12	32768	130	0	0.7	0.0	pcidossvr
2089	-	S	0:22	918	72	28	xx	1	4	0.0	0.0	/etc/sync
2784	-	S	0:00	9	16	8	xx	0	0	0.0	0.0	kproc
2816	-	S	5:59	6436	2664	616	8	852	156	0.4	4.0	/usr/lpp/
3115	-	S	0:27	955	264	128	xx	39	36	0.0	1.0	/usr/lib/
3451	-	S	0:00	0	16	8	xx	0	0	0.0	0.0	kproc
3812	-	S	0:00	21	128	12	32768	34	0	0.0	0.0	/usr/lpd/
3970	-	S	0:00	0	16	8	xx	0	0	0.0	0.0	kproc
4267	-	S	0:01	169	132	72	32768	16	16	0.0	0.0	/etc/sysl
4514	hft/0	S	0:00	60	200	72	xx	39	60	0.0	0.0	/etc/gett
4776	pts/3	S	0:02	250	108	280	8	303	268	0.0	2.0	-ksh
5050	-	S	0:09	1200	424	132	32768	243	56	0.0	1.0	/usr/sbin
5322	-	S	0:27	1299	156	192	xx	24	24	0.0	1.0	/etc/cron
5590	-	S	0:00	2	100	12	32768	11	0	0.0	0.0	/etc/writ
5749	-	S	0:00	0	208	12	xx	13	0	0.0	0.0	/usr/lpp/
6111	-	S	0:00	66	108	12	32768	47	0	0.0	0.0	/usr/lpp/

Prozeßstatus mit der Einrichtung SMIT anzeigen

1. Den SMIT-Direktaufrufbefehl eingeben:

```
smit monitors
```

Mit diesem Befehl wird die Anzeige Ressourcenstatus & Überwachungsprogramme angezeigt.

2. Die Option Prozeßstatus anzeigen markieren und die Eingabetaste drücken. SMIT führt dann den Befehl **ps gv** aus und zeigt seitenweise allen aktiven Prozesse an.

Prozesse löschen

Den Befehl **kill** verwenden, um einen Prozeß zu löschen. Der Befehl **kill** kann auf eine der folgenden Arten verwendet werden:

- Von der Befehlszeile aus
- Mit der Einrichtung SMIT.

Vorbedingung

Um einen Prozeß zu löschen, muß der Benutzer eine Berechtigung für das Stammverzeichnis besitzen oder den Prozeß selbst gestartet haben.

Prozesse von der Befehlszeile aus löschen

Folgendes eingeben, um einen Prozeß zu löschen:

```
kill PROZESS-ID
```

Mit diesem Befehl wird das Signal **SIGTERM** an den angegebenen Prozeß gesendet. Um andere Signale, wie z. B. das Signal **SIGKILL**, an den Prozeß zu senden, die Markierung **-s** oder die Signalnummer verwenden:

```
kill -s SIGKILL PROZESS-ID
```

oder

```
kill -9 PROZESS-ID
```

Mit Hilfe des Befehls **ps** kann die Prozeß-ID eines bestimmten Prozesses angezeigt werden.

Prozesse mit der Einrichtung SMIT löschen

1. Den SMIT-Direktaufbefehl eingeben:

```
smit kill
```

2. Um das Signal **SIGKILL** anstatt des Signals **SIGTERM** auszuwählen, durch Drücken der Funktionstaste 4 die Liste **SIGNALTYP** anzeigen. Mit der Abwärtspfeiltaste **SIGKILL** auswählen und anschließend die Eingabetaste drücken.

Überblick über Shells

Die Benutzerschnittstelle zum Betriebssystem wird als *Shell* bezeichnet.

Die Shell ist der am weitesten außen liegende Bereich des Betriebssystems. Sie enthält eine Programmiersprache, mit der sowohl Prozesse und Dateien gesteuert als auch andere Programme gestartet und gesteuert werden. Von der Shell wird die Interaktion zwischen dem Benutzer und dem Betriebssystem verwaltet, indem der Benutzer aufgefordert wird, eine Eingabe vorzunehmen, diese Eingabe für das Betriebssystem interpretiert und die daraus resultierende Ausgabe des Betriebssystems bearbeitet wird.

Weitere Informationen zu den Shells und ihrer Verwendung stehen in den folgenden Abschnitten:

- Einführung in die Shells
- Shell-Funktionen
- Zur Verfügung stehende Shells
- Definition von Shell-Begriffen

Einführung in die Shells

Eine Shell dient der Kommunikation zwischen einem Benutzer und dem Betriebssystem. Diese Kommunikation kann interaktiv (auf eine Eingabe über die Tastatur erfolgt unmittelbar eine Reaktion) oder über eine Shell-Prozedur erfolgen. Eine Shell-Prozedur ist eine Folge von Shell-Befehlen und Betriebssystembefehlen, die in einer Datei gespeichert sind.

Beim Anmelden lokalisiert das System den Namen des auszuführenden Shell-Programms. Nachdem das Programm ausgeführt wurde, zeigt die Shell eine Eingabeaufforderung an. Diese Eingabeaufforderung besteht normalerweise aus einem \$ (Dollarzeichen). Wird bei der Eingabeaufforderung ein Befehl eingegeben und anschließend die Eingabetaste gedrückt, wertet die Shell den Befehl aus und versucht, ihn auszuführen. Entsprechend den Anweisungen im Befehl schreibt die Shell die Befehlsausgabe entweder auf den Bildschirm oder leitet die Ausgabe um. Daraufhin zeigt sie wieder die Eingabeaufforderung an und wartet auf die Eingabe neuer Befehle.

Eine *Befehlszeile* ist die Zeile, die bei der Shell-Eingabeaufforderung eingegeben wird. Das Basisformat für jede Zeile ist:

Befehl Argumente

Für die Shell stellt das erste Wort einer Befehlszeile (bis zur ersten Leerstelle) den Befehl dar und das zweite die Argumente.

Shell-Funktionen

Es folgt eine Auflistung der wesentlichen Vorteile einer Shell als Schnittstelle zum Betriebssystem:

Ersetzen von Platzhalterzeichen in Dateinamen (Mustererkennung)

Befehle können durch Angabe eines Musters anstelle eines bestimmten Dateinamens an einer Gruppe von Dateien ausgeführt werden.

Hintergrundverarbeitung

Aufgaben, deren Ausführung erhöhten Zeitaufwand erfordern, können im Hintergrund verarbeitet werden. Dadurch steht die Datenstation während dieser Zeit für Dialogverarbeitung zur Verfügung.

Alias-Namen für Befehle

Einem Befehl oder einer Phrase können Alias-Namen zugeordnet werden. Wenn die Shell einen Alias-Namen in der Befehlszeile oder in einer Shell-Prozedur feststellt, ersetzt sie diesen durch den Text, der dem Alias-Namen zugeordnet ist.

Hinweis: Das Zuordnen von Alias-Namen für Befehle wird von der Bourne-Shell nicht unterstützt.

Befehlsprotokoll

Zeichnet die eingegebenen Befehle in der Protokolldatei auf. Diese Datei ermöglicht auf einfache Weise den Zugriff auf jeden aufgelisteten Befehl sowie dessen Änderung und Wiederausgabe.

Dateinamensubstitution

Erstellt automatisch eine Liste von Dateinamen auf einer Befehlszeile mit Hilfe von Zeichen, deren Muster übereinstimmen.

Umadressierung von Eingabe und Ausgabe

Die Eingabe kann von der Tastatur zu einer anderen Eingabeeinheit umgeleitet werden. Die Ausgabe kann anstatt an die Datenstation an eine andere Einheit umgeleitet werden. Die Eingabe für ein Programm könnte zum Beispiel von einer Datei zur Verfügung gestellt und an den Drucker oder eine andere Datei umgeleitet werden.

Verarbeitung im Pipelinesystem

Befehle können in beliebiger Zahl zu einem komplexen Programm miteinander verbunden werden. Die Standardausgabe eines Programms wird zur Standardeingabe für das nächste Programm.

Ersetzen von Shell-Variablen

Daten können in benutzerdefinierten Variablen und in vordefinierten Shell-Variablen gespeichert werden.

Zur Verfügung stehende Shells

Mit der vorliegenden Version des Betriebssystems werden folgende Shells bereitgestellt:

- Korn-Shell (kann mit dem Befehl **ksh** aufgerufen werden)
- Bourne-Shell (kann mit dem Befehl **bsh** aufgerufen werden)
- Eingeschränkte Shell (eine Untergruppe der Bourne-Shell, die mit dem Befehl **Rsh** aufgerufen werden kann)
- C-Shell (kann mit dem Befehl **cs** aufgerufen werden)
- Gesicherte Shell (kann mit dem Befehl **tsh** aufgerufen werden)
- Ferne Shell (kann mit dem Befehl **rsh** aufgerufen werden).

Als *Standard-Shell* wird die Shell bezeichnet, die dem Befehl **/usr/bin/sh** zugeordnet ist. Die Bourne-Shell wird als Standard-Shell eingerichtet.

Als *Anmelde-Shell* wird die Shell bezeichnet, die beim Anmelden am Datenverarbeitungssystem geladen wird. Die Anmelde-Shell wird in der Datei **/etc/passwd** festgelegt. Die Korn-Shell wird standardmäßig als Anmelde-Shell eingerichtet.

Definition von Shell-Begriffen

Folgende Definitionen erleichtern das Verständnis von Shells:

Leerzeichen Ein Tabulator oder ein Leerschritt.

Befehl	Eine Folge von Zeichen in der Syntax der Shell-Sprache. Von der Shell wird jeder Befehl gelesen und die gewünschte Operation entweder direkt oder durch Aufrufen separater Dienstprogramme ausgeführt.
Kommentar	Jedes mit einem Nummernzeichen (#) beginnende Wort. Alle nachfolgenden Wörter und Zeichen bis zum nächsten Zeilenvorschubzeichen werden ignoriert.
Kennung	Eine Folge von Buchstaben, Ziffern oder Unterstreichungszeichen, die mit einem Buchstaben oder einem Unterstreichungszeichen beginnt. Kennungen werden als Alias-Namen oder als Namen für Funktionen und benannte Parameter verwendet.
Liste	Eine Folge von einer oder mehreren Pipelines, die durch eines der folgenden vier Symbole voneinander getrennt werden: ; (Semikolon), & (Et-Zeichen), && (doppeltes Et-Zeichen) oder (doppelter vertikaler Strich). Die Liste kann wahlweise durch eines der folgenden Symbole beendet werden: ; (Semikolon), & (Et-Zeichen), oder & (vertikaler Strich Et-Zeichen). <ul style="list-style-type: none"> ; & & &&

- ;
- &
- |&
- &&
- ||

Die Symbole ; (Semikolon), & (Et-Zeichen) und |& (vertikaler Strich Et-Zeichen) haben eine niedrigere Priorität als das && (doppeltes Et-Zeichen) und || (doppelter vertikaler Strich). Die Symbole ;, &, und |& haben dieselbe Priorität, genauso wie die Symbole && und ||. Zum Abgrenzen von zwei Befehlen in einer Liste können anstelle eines Semikolons ein oder mehrere Zeilenvorschubzeichen verwendet werden.

Hinweis: Das Symbol |& ist nur in der Korn-Shell gültig.

Metazeichen

Alle Metazeichen haben eine Sonderbedeutung für die Shell und dienen zur Begrenzung eines Wortes, sofern sie nicht in Anführungszeichen stehen. Zu den Metazeichen zählen folgende Zeichen: | (vertikaler Strich), & (Et-Zeichen), ; (Semikolon), < (Kleiner-als-Zeichen), > (Größer-als-Zeichen), ((linke Klammer),) (rechte Klammer), \$ (Dollarzeichen), ' (linkes Anführungszeichen), \ (umgekehrter Schrägstrich), ' (rechtes Anführungszeichen), " (doppeltes Anführungszeichen), Zeilenvorschubzeichen, Leerzeichen und Tabulatorzeichen. Stehen diese

Zeichen zwischen Anführungszeichen, werden sie wie normaler ASCII-Text behandelt; ansonsten behalten sie ihre Sonderbedeutung bei. (Metazeichen werden in der C-Shell auch als *Parser-Metazeichen* bezeichnet.)

Parameterzuordnungsliste

Umfaßt eine oder mehrere Wörter im Format *Kennung=Wert* (*Kennung=Wert* für die C-Shell, in der die Leerzeichen erforderlich sind). Es müssen entweder führende oder folgende Leerzeichen angegeben oder beide weggelassen werden.

Hinweis: In der C-Shell erfolgt die Parameterzuordnung im Format *set Kennung = Wert*. Es sind führende und folgende Leerzeichen zum Gleichheitszeichen erforderlich.

Pipeline

Ein Befehl oder eine Folge von Befehlen, die durch das Pipe-Symbol, einen vertikalen Strich (|), voneinander getrennt sind. Jeder Befehl in der Pipeline wird als ein eigener Prozeß ausgeführt. Die Standardausgabe jedes Befehls in der Pipe wird zur Standardeingabe für den folgenden Befehl in der Pipeline. Der Rückkehrcode der Pipeline entspricht dem Rückkehrcode des letzten Befehls in der Folge. Steht eine Liste zwischen runden Klammern, kann sie als ein einfacher Befehl in einer anderen Subshell ausgeführt werden.

Hinweis: In früheren Versionen der Bourne-Shell wurde eine Pipe durch ein Winkelzeichen (^) gekennzeichnet.

Shell-Variable

Ein Name oder Parameter, dem ein Wert zugeordnet werden kann. Das Zuordnen der Variablen erfolgt durch Eingeben des Variablennamens, des Gleichheitszeichens und des Wertes. Der Variablenname kann dann ein vorangestelltes \$ (Dollarzeichen) anstelle des zugeordneten Wertes angegeben werden. Variablen dienen besonders als kurze Bezeichnung für einen langen Pfadnamen, wie z. B. **\$HOME** als Variable für das Benutzerverzeichnis. Einer vordefinierten Variablen wird von der Shell ein Wert zugeordnet. Unter einer benutzerdefinierten Variablen versteht man eine Variable, der vom Benutzer ein Wert zugeordnet wird.

Einfacher Befehl

Eine Folge von Wörtern, die durch Leerzeichen voneinander getrennt sind. Einem einfachen Befehl kann eine Parameterzuordnungsliste vorausgehen. Das erste Wort in einem einfachen Befehl ist der Name des auszuführenden Befehls. Werden weder Pipelines noch Listen verwendet, werden die weiteren Wörter in dem einfachen Befehl als Parameter an die aufgerufene Unterroutine übergeben. Der Befehlsname wird als Parameter 0 übergeben (entsprechend der Definition der Unterroutine **exec**). Der Wert eines einfachen Befehls ist gleich seinem Rückkehrcode, wenn er normal beendet wird, oder (oktal) 200+Status wenn er abnormal beendet wird. In der Dokumentation zu den Unterroutinen **sigaction**, **sigvec** und **signal** ist eine Liste mit Statuswerten enthalten.

Sonderbefehl

Ein Befehl, der von der Shell ausgeführt wird, ohne einen eigenen Prozeß zu starten. Wenn keine unerwünschten Ergebnisse eines Sonderbefehls dokumentiert sind, können die meisten von ihnen als eigene Dienstprogramme implementiert werden.

Subshell

Eine Shell, die als Sohnprozeß der Anmelde-Shell ausgeführt wird.

Platzhalterzeichen

Auch bekannt als Suchmustervergleichszeichen. Dieses Zeichen ist ein

	Sonderzeichen, dem von der Shell ein vorgegebener Wert zugeordnet wird. Platzhalterzeichen sind besonders geeignet für die Dateinamenssubstitution.
Wort	Eine Folge von Zeichen, die keine Leerzeichen enthält. Wörter werden durch ein oder mehrere, nicht zwischen Metazeichen stehende • Metazeichen voneinander getrennt.

Verwendung der Shell-Prozedur

Shell-Prozeduren bieten eine einfache Möglichkeit, komplexe Befehle, lange oder komplexe Befehlsfolgen und Routineaufgaben auszuführen. Als Shell-Prozedur wird eine Datei bezeichnet, die einen oder mehrere Shell-Befehle enthält. Die Befehlsfolge in einer Shell-Prozedur kann durch Eingabe des Namens der Shell-Prozedurdatei ausgeführt werden.

Zum Erstellen der Shell-Prozedurdatei einen Texteditor, z. B. `ed` oder `vi` verwenden. Die Prozedurdatei kann sowohl Betriebssystembefehle als auch Shell-Befehle enthalten.

Die Ausführungsberechtigung für die Prozedurdatei nach dem Erstellen und Sichern durch Eingabe des Befehls `chmod` setzen. Dieser Befehl erteilt allen zur Gruppe gehörenden Benutzern eine Ein- und Ausgabeberechtigung für die Datei.

Zum Ausführen der Shell-Prozedur den Namen der Shell-Prozedurdatei in die Befehlszeile eingeben.

Hinweis: Eine Shell-Prozedur kann ohne Ausführungsberechtigung ausgeführt werden, wenn ein Shell-Befehl (**ksh**, **bsh** oder **csh**) dem Shell-Prozedurdateinamen in der Befehlszeile vorangestellt ist. Um z. B. eine Datei ohne Ausführungsberechtigung mit dem Namen `Skript1` unter Steuerung der Korn-Shell auszuführen, folgendes eingeben:

```
ksh Skript1
```

Shell-Prozeduren, die zur Verwendung durch viele Benutzer eingerichtet werden, werden gemäß Vereinbarung im Verzeichnis `/home/bin` gespeichert.

Hinweis: `setuid` oder `setgid` werden vom Betriebssystem nicht in einer Shell-Prozedur unterstützt.

Eine Shell für eine Prozedurdatei angeben

Wenn eine ausführbare Datei entweder in der Korn-Shell oder in der Bourne-Shell ausgeführt wird, werden die Befehle in der Prozedur unter der Steuerung der aktuellen Shell (von der aus die Prozedur gestartet wird) ausgeführt, es sei denn, es wird eine andere Shell angegeben. Wird eine ausführbare Shell-Prozedur in der C-Shell ausgeführt, werden die Befehle in der Prozedur unter der Steuerung der Standard-Shell (die Shell, die der Datei `/usr/bin/sh` zugeordnet ist und mit dem Befehl `sh` gestartet wird) ausgeführt, es sei denn, es wird eine andere Shell angegeben.

Eine Shell-Prozedur kann in einer bestimmten Shell ausgeführt werden, indem die Shell mit in die Shell-Prozedur einbezogen wird. Soll eine ausführbare Shell-Prozedur in einer anderen Shell ausgeführt werden, `#! Pfad` in die erste Zeile der Shell-Prozedur eingeben. Bei den Zeichen `#!` handelt es sich um die Anzeiger des Dateityps. Mit der Variablen `Pfad` wird der Pfadname zu der Shell angegeben, von der aus die Shell-Prozedur ausgeführt werden soll. Lautet die erste Zeile in einer Shell-Prozedur z. B. `#! /usr/bin/bsh`, wird die Prozedur unter der Steuerung der Bourne-Shell ausgeführt.

Wird dem Namen einer Shell-Prozedurdatei ein Shell-Befehl vorangestellt, setzt die Angabe dieser Shell in der Befehlszeile die Angaben aller Shells in der Shell-Prozedur außer Kraft.

Daher wird bei Eingabe von `ksh meinDat` die Datei mit dem Namen `meinDat` unter Steuerung der Korn-Shell ausgeführt, selbst wenn die erste Zeile der Datei `meinDat` die Angabe `#!/usr/bin/csh` enthält.

Eine Shell-Prozedur erstellen und ausführen

Voraussetzungen

1. Der Benutzer muß Zugriff auf einen Texteditor haben.
2. Eine Shell muß geöffnet sein.

Erforderliche Schritte

Die folgenden Schritte dienen als allgemeine Richtlinien zum Schreiben von Shell-Prozeduren:

1. Mit Hilfe eines Texteditors, z. B. ed oder vi, eine Datei erstellen und sichern. Die Shell-Prozedurdatei kann jede beliebige Kombination von Shellbefehlen und Betriebssystembefehlen enthalten.
2. Die Ausführungsberechtigung für die Prozedurdatei mit dem Befehl **chmod** setzen. Wenn z. B. die Datei `meinDat` heißt, `chmod u=rwx meinDat` eingeben.
3. Zum Ausführen der Shell-Prozedur in der Befehlszeile den Namen der Shell-Prozedur eingeben.

Hinweis: Auf die Shell-Prozedurdatei kann auch ohne Ausführungsberechtigung zugegriffen werden, wenn ein Shell-Befehl (**ksh**, **bsh** oder **cs**) dem Shell-Prozedurdateinamen in der Befehlszeile vorangestellt wird.

Überblick über Ein- und Ausgabe

Die Ein- und Ausgabe (E/A) von Daten in das und aus dem Betriebssystem kann über spezielle E/A-Befehle und -Symbole gesteuert werden. Die Eingabe von Daten kann durch Angabe der Stelle, von der Daten geholt werden sollen, gesteuert werden. Die Eingabe kann z. B. über die Tastatur erfolgen (Standardeingabe) oder durch Lesen aus einer Datei. Die Ausgabe von Daten kann durch Angabe der Stelle erfolgen, an der Daten angezeigt oder gespeichert werden sollen. Die Ausgabe kann z. B. durch Anzeigen am Bildschirm (Standardausgabeeinheit) oder durch Schreiben in eine Datei erfolgen.

Ein- und Ausgabe mit Befehlen steuern

Ein- und Ausgabebefehle bearbeiten Nachrichten, die am Bildschirm angezeigt werden. Die Befehle **banner** und **echo** zeigen benutzerspezifische System- oder Fehlermeldungen am Bildschirm an. Der Befehl **clear** löscht die Bildschirmanzeige. Die Befehle **tee**, **script** und **capture** speichern am Bildschirm angezeigte Daten in einer Datei.

Ein- und Ausgabe mit Symbolen steuern

Die Eingabe für einen Befehl wird normalerweise über die Tastatur (Standardeingabeeinheit) vorgenommen, verarbeitet und als Ausgabe am Bildschirm (Standardausgabeeinheit) angezeigt. Die Art und Weise, wie Befehle die Ein- und Ausgabe von Daten handhaben, kann durch die Angabe von E/A-Symbolen bestimmt werden. Zum Beispiel kann die Eingabe für einen Befehl aus einer Datei gelesen, die Ausgabe in eine Datei gestellt oder ein anderer Befehl als Eingabe verwendet werden. Mit den folgenden Symbolen kann die Ein- und Ausgabe umgeleitet werden:

<	Liest die Standardeingabe aus einer Datei.
>	Schreibt die Standardausgabe in eine Datei.
>>	Fügt die Standardausgabe an das Ende einer Datei an.
	Leitet die Ausgabe von einem Befehl über eine Pipe zu einem zweiten Befehl.

Eingabe aus einer Datei lesen

Mit dem Symbol < (Kleiner-als-Zeichen) wird angegeben, daß die Eingabe aus einer Datei gelesen werden soll, z. B. aus der Datei3 in den Befehl **wc** (word count – Wörter zählen):

```
wc < Datei3
```

Mit dem Befehl **wc** werden die Zeilen, Wörter und Zeichen in Datei3 gezählt, und das Ergebnis wird an die Standardausgabeeinheit gesendet. Wird keine Eingabedatei eingegeben, erfolgt die Eingabe für den Befehl **wc** über die Tastatur.

Ausgabe in eine Datei umleiten

Das Symbol > (Größer-als-Zeichen) oder das Symbol >> wird verwendet, wenn die Ausgabe an eine Datei gesendet werden soll. Das Symbol > bewirkt, daß die Shell den Inhalt der Datei durch die Ausgabe des Befehls ersetzt; die Shell löscht den Inhalt der Originaldatei. Das Symbol >> bewirkt, daß die Ausgabe des Befehls an das Ende einer Datei angefügt wird. Wird das Symbol > oder >> verwendet, um die Ausgabe in eine nicht vorhandene Datei zu schreiben, erstellt die Shell die Datei.

Die Ausgabe von **ls** z. B. an die Datei temp senden:

```
ls > temp
```

Ist die Datei temp bereits vorhanden, ersetzt die Shell ihren Inhalt durch die Ausgabe des Befehls **ls**. Andernfalls wird die Datei von der Shell erstellt.

Im folgenden Beispiel fügt die Shell die Ausgabe von `ls` am Ende von Datei `temp` an:

```
ls >> temp
```

Andernfalls wird die Datei von der Shell erstellt.

Im folgenden Beispiel wird der Befehl `find` im Hintergrund ausgeführt. Die Ausgabe wird mit Hilfe des Symbols `>` an die Datei `dir.paths` geleitet:

```
find / -type f -print > dir.paths &
```

Die ID des Hintergrundprozesses wird angezeigt und die Cursorsteuerung wieder dem Benutzer übergeben.

Die Ausgabe sowohl in eine Datei als auch am Bildschirm erfolgt mit Hilfe des Befehls `tee`.

Ausgabe zu einem weiteren Befehl umleiten

Mit Hilfe einer Pipeline kann die Ausgabe eines Befehls von einem anderen Befehl als Eingabe eingelesen werden. Die Befehle werden durch den Operator `|` (Pipe) miteinander verbunden.

Der Befehl `li` z. B. zeigt den Inhalt des aktuellen Verzeichnisses in einem fortlaufenden Datenstrom an. Wenn der Datenstrom größer ist als der Bildschirmbereich, können einige Daten nicht gesehen werden. Im folgenden Beispiel werden die Befehle `li` und `pg` miteinander verbunden, um den Inhalt des Verzeichnisses seitenweise anzuzeigen:

```
li | pg
```

Im obigen Beispiel ist die Ausgabe des Befehls `li` die Eingabe für den Befehl `pg`.

Pipelines verlaufen nur in einer Richtung (von links nach rechts). Jeder Befehl in einer Pipeline läuft als separater Prozeß und alle Prozesse können gleichzeitig ablaufen. Ein Prozeß wird unterbrochen, wenn keine Eingabe zum Lesen vorhanden ist, oder wenn für die Pipe an den nächsten Prozeß keine weiteren Angaben mehr möglich sind.

Standardfehlerausgabe und andere Arten der Ausgabe umleiten

Zusätzlich zur Standardausgabe geben Prozesse häufig Fehler- oder Statusnachrichten aus, die sogenannte Diagnoseausgabe. Die Diagnoseausgabe wird ebenso wie die Standardausgabe am Bildschirm angezeigt, sofern sie nicht umgeleitet wird.

Im allgemeinen sind bereits drei Dateien geöffnet, wenn ein Befehl gestartet wird: `stdin` (Standardeingabe), `stdout` (Standardausgabe) und `stderr` (Standardfehler). Soll die Standardeingabe oder -ausgabe umgeleitet werden, kann das Symbol `<`, `>` oder `>>` verwendet werden. Soll jedoch die Standardfehlerausgabe (oder eine andere Ausgabe) umgeleitet werden, muß ein Dateideskriptor für die Standardfehlerausgabe verwendet werden. Dateideskriptoren können auch zum Umleiten der Standardeingabe und -ausgabe verwendet werden.

Jeder Datei, die von einem Befehl verwendet wird, wird eine Zahl, ein sogenannter Dateideskriptor, zugeordnet. Die folgenden Zahlen sind den Dateien Standardeingabe, -ausgabe und -fehlerausgabe zugeordnet:

0	Standardeingabeeinheit (Tastatur)
1	Standardausgabeeeinheit (Bildschirm)
2	Standardfehlerausgabeeeinheit (Bildschirm).

Zum Umleiten der Standardfehlerausgabe eine `2` (die Dateideskriptornummer) eingeben, bevor Ausgabeumadressierungssymbole (`>` und `>>`) und ein darauffolgender Dateiname (nach dem Symbol) eingegeben werden. Der folgende Befehl fügt die Standardfehlerausgabe des Befehls `cc` an die Datei `FEHLER` an:

```
cc testdatei.c 2>> FEHLER
```

Mit Hilfe der oben beschriebenen Methode zum Umleiten der Standardfehlerausgabe kann jede Ausgabe umgeleitet werden, der ein beliebiger Dateideskriptor von 0 bis 9 zugeordnet ist. Schreibt der Befehl `cmd` z. B. die Ausgabe an Dateideskriptor 9, kann diese Ausgabe mit dem folgenden Befehl an die Datei `Datensicherung` umgeleitet werden:

```
cmd 9> Datensicherung
```

Erfolgt die Ausgabe eines Befehls an mehrere verschiedene Dateideskriptoren, können diese unabhängig voneinander umgeleitet werden, wie das folgende Beispiel zeigt:

```
cmd > Standard 2> Fehler 9> Daten
```

Im obigen Beispiel wird die Standardausgabe (Dateideskriptor 1) an die Datei `Standard` geleitet, die Standardfehlerausgabe (Dateideskriptor 2) an die Datei `Fehler` und die Datenausgabe (Dateideskriptor 9) an die Datei `Daten`.

Empfohlene Dokumentation

Zugehörige Informationen

Im Abschnitt "Überblick über Befehle" auf Seite 2-41 wird beschrieben, wie Befehle vom Betriebssystem verarbeitet werden.

Im Abschnitt "Überblick über Dateien" auf Seite 3-3 sind Informationen über Dateien aufgeführt, die bei der Umadressierung hilfreich sind.

Im Kapitel "Überblick über Prozesse" auf Seite 2-14 wird erläutert, was Prozesse sind.

Im Kapitel "Überblick über Shells" auf Seite 2-21 wird beschrieben, was Shells sind, welche verschiedenen Arten von Shells es gibt und wie Shells die verschiedenen Möglichkeiten der Befehls-Interpretation beeinflussen.

Die Abschnitte zu den Befehlen **banner**, **capture**, **clear**, **echo**, **script** und **tee**.

Überblick über Profile

Die Shell verwendet zwei Arten von Profildateien beim Anmelden im Betriebssystem. Sie wertet die in den Dateien enthaltenen Befehle aus und richtet die Systemumgebung ein. Die Dateien haben ähnliche Funktionen, mit Ausnahme der Datei **/etc/profile**, die die Profilvariablen für alle Benutzer eines Systems steuert, und der Datei **.profile**, die es dem Benutzer erlaubt, seine eigene Umgebung anzupassen.

Datei **/etc/profile**

Die Datei **/etc/profile** wird zum Anmeldezeitpunkt als erste Datei eingesetzt. Sie steuert folgende systemweit gültigen Standardvariablen:

- Exportvariablen
- Dateierstellungsmaske (umask)
- Datenstationstypen
- Nachrichten der elektronischen Post, die anzeigen, wenn neue Post eingetroffen ist.

Der Systemadministrator konfiguriert die Datei **profile** für alle Benutzer im System. Die Datei **profile** kann nur vom Systemadministrator geändert werden.

Die Datei **.profile**

Die Datei **.profile** wird zum Anmeldezeitpunkt als zweite Datei verwendet. Die Datei **.profile** liegt im Benutzerverzeichnis (**\$HOME**) vor und erlaubt es dem Benutzer, seine eigene Umgebung anzupassen. Die Datei **.profile** überschreibt die Befehle und Variablen, die in der Datei **/etc/profile** festgelegt werden. Zum Auffinden der Datei **.profile**, bei der es sich um eine verdeckte Datei handelt, den Befehl **ls -a** verwenden. Mit der Datei **.profile** kann folgendes gesteuert werden:

- Was beim Systemstart gestartet wird (z.B. welche Shell oder Shells gestartet werden sollen)
- Welche Eingabeaufforderung angezeigt wird
- Umgebungsvariablen (z. B. Suchpfade)
- Ein- und Ausschalten des Tastaturgeräuschs.

Im folgenden Beispiel wird eine typische Profildatei **.profile** gezeigt:

```
PATH=/usr/bin:/etc:/home/bin:/usr/lpp/tps4.0/user:/home/gsc/bin::
epath=/home/gsc/e3:
export PATH epath
csh
```

Im obigen Beispiel hat der Benutzer zwei Pfade angelegt (PATH und epath), diese exportiert und eine C-Shell (csh) geöffnet.

Zum Festlegen der Variablen für die Anmelde-Shell wird die Profildatei **.profile** oder, falls diese nicht vorhanden ist, die Datei **profile** verwendet. Zusätzlich zu diesen Profilen, mit denen die Anmelde-Shell angepaßt wird, können Benutzer auch andere Shell-Umgebungen anpassen. Zum Beispiel werden die Dateien **.cshrc** und **.kshrc** verwendet, um eine C-Shell und Korn-Shell anzupassen, wenn der jeweilige Shell-Typ gestartet wird.

Empfohlene Dokumentation

Zugehörige Informationen

Der Abschnitt "Überblick über Shells" auf Seite 2-21.

Die Abschnitte zu den Befehlen **env** und **printenv**.

Der Befehl **ls -a** listet verdeckte Dateien wie die Datei **.profile** auf.

Das Dateiformat **profile** zeigt ein Beispiel für eine typische Profildatei **/etc/profile**.

Die Datei **environment**.

Überblick über das An- und Abmelden

Damit der Benutzer mit dem Betriebssystem arbeiten kann, muß das System aktiv und der Benutzer angemeldet sein. Wenn sich der Benutzer am Betriebssystem anmeldet, gibt er sich dem System zu erkennen und ermöglicht damit, daß das System die Betriebsumgebung für ihn einrichtet.

Die Anmeldung am Betriebssystem erfolgt über den Anmeldedialog. Es wird automatisch das Anmeldeverzeichnis (auch Benutzerverzeichnis genannt) aufgerufen. In diesem Verzeichnis befindet sich eine verdeckte Datei namens **.profile**. Bei der Anmeldung am Betriebssystem werden die in dieser Datei enthaltenen Befehle ausgeführt. Die Datei **.profile** (bzw. die Datei **.cshrc** in der csh-Shell) kann angepaßt werden, damit beim Anmelden bestimmte Befehle ausgeführt oder bestimmte Pfadnamen gesucht werden.

Bei einem Betriebssystem sind mehrere gültige Anmeldenamen möglich. Soll von einem Anmeldenamen zu einem anderen gewechselt werden, kann dies ohne Abmelden vom System erfolgen. Entweder können verschiedene Anmeldenamen in verschiedenen Shells gleichzeitig oder in derselben Shell nacheinander verwendet werden. Außerdem kann eine Anmeldung an jedem anderen System erfolgen, für das eine Anmelde-ID vorhanden ist, sofern das System des Benutzers an ein Netz angeschlossen ist. Dies wird als Fernanmeldung bezeichnet.

Nach Beenden der Arbeit mit dem Betriebssystem muß sich der Benutzer abmelden, um sicherzustellen, daß die Dateien und Daten geschützt sind. Die Abmeldung vom Betriebssystem erfolgt über die Tastenfolge für DATEIENDE (Strg-D) oder mit dem Befehl **logout**.

Anmelden an mehreren Systemen

Ist das eigene System an ein Netz angeschlossen, ist eine Fernanmeldung möglich. Die Fernanmeldung ermöglicht das Anmelden an allen Systemen des Netzes, für das eine Anmelde-ID vorhanden ist. Andere Benutzer, die eine Anmelde-ID für das System haben, können sich von fernen Systemen aus ebenfalls anmelden.

Es gibt mehrere Möglichkeiten der Fernanmeldung. Die Verbindung des eigenen Systems mit anderen Systemen im Netz wird u. a. über den Befehl **rlogin** hergestellt. Soll das eigene System mit einem anderen System verbunden und dabei ein spezieller Befehl ausgeführt werden, ist dies mit dem Befehl **rsh** möglich. Auch mit dem Befehl **telnet** ist eine Fernanmeldung möglich. Von der Methode der Fernanmeldung an das System hängt auch die der Abmeldung ab.

Häufig erweist sich die Kenntnis darüber, welche lokalen und fernen Benutzer gegenwärtig am System angemeldet sind, als hilfreich. Mit dem Befehl **who** erhält der Benutzer eine Liste der aktuellen Benutzer des Systems und deren Datenstationsnamen sowie Datum und Zeitpunkt der Anmeldung. Benutzer, die an mehreren Systemen angemeldet sind, sind in der Liste mehrmals aufgeführt.

Mehrfaches Anmelden

Im Betriebssystem können für mehrfaches Anmelden entweder derselbe oder verschiedene Anmeldenamen verwendet werden. Das ist sehr nützlich, wenn an mehreren Projekten gearbeitet wird und die Benutzerkonten getrennt verwaltet werden sollen. Bei jedem Betriebssystem gibt es eine maximale Anzahl von Anmeldenamen, die zu einer gegebenen Zeit aktiv sein können. Diese Zahl hängt von der jeweiligen Lizenz und Installation ab.

Bei mehrfacher Anmeldung geht möglicherweise der Überblick über die Anmeldenamen verloren. Der jeweils verwendete Anmelde-name kann mit dem Befehl **whoami** oder **id**

festgestellt werden. Es ist auch möglich, die Eingabeaufforderung der Shell so zu setzen, daß der aktuelle Anmeldename angezeigt wird.

Empfohlene Dokumentation

Grundlegende Informationen

Im Abschnitt "Überblick über Shells" auf Seite 2-21 wird beschrieben, was Shells sind, welche verschiedenen Arten von Shells es gibt und wie Shells die verschiedenen Möglichkeiten der Befehlsinterpretation beeinflussen.

Zugehörige Informationen

Der Abschnitt "Anmelden am Betriebssystem" auf Seite 2-35 enthält Anweisungen für das Anmelden.

Der Abschnitt "Abmelden am Betriebssystem" auf Seite 2-36 enthält Anweisungen für das Abmelden.

Im Abschnitt "Überblick über virtuelle Datenstationen" auf Seite 2-38 wird beschrieben, wie eine physische Datenstation als Mehrplatzdatenstation eingesetzt werden kann.

Im Abschnitt "Überblick über Profile" auf Seite 2-31 werden die verdeckten Dateien beschrieben.

Im Abschnitt zum Befehl **id** wird beschrieben, wie der aktuelle Anmeldename festgestellt werden kann.

Im Abschnitt zum Befehl **login** wird das Anmelden am Betriebssystem beschrieben.

Im Abschnitt zum Befehl **logout** wird das Abmelden am Betriebssystem beschrieben.

Im Abschnitt zum Befehl **rlogin** wird die Fernanmeldung beschrieben.

Im Abschnitt zum Befehl **rsh** werden die Fernanmeldung und das Ausführen eines Befehls beschrieben.

Im Abschnitt zum Befehl **su** wird beschrieben, wie man sich unter einem anderen Benutzernamen (einschließlich Superuser) am System anmelden kann.

Im Abschnitt zum Befehl **telnet** wird das Anmelden an einem fernen System beschrieben.

Im Abschnitt zum Befehl **who** wird beschrieben, wie festgestellt werden kann, welche Benutzer gegenwärtig am System angemeldet sind.

Im Abschnitt zum Befehl **whoami** wird beschrieben, wie der eigene Anmeldename festgestellt wird.

Anmelden am Betriebssystem

Bei nicht aktiviertem System

1. Die Hardwarekomponenten über die Netzschalter einschalten.
2. Weiter mit Schritt 1 im nächsten Abschnitt.

Bei aktiviertem System

1. Den Anmeldenamen bei der Eingabeaufforderung `login:` eingeben und die Eingabetaste drücken:

`login: AnmeldeName`

Wurde kein Kennwort festgelegt, kann anschließend mit dem Betriebssystem gearbeitet werden.

2. Erscheint die Aufforderung zur Kennworteingabe, das Kennwort eingeben. Wenn kein Kennwort zugeordnet wurde, erscheint auch keine entsprechende Eingabeaufforderung. Das Kennwort wird bei der Eingabe nicht angezeigt. Anschließend die Eingabetaste drücken:

`password: [Kennwort]`

Empfohlene Dokumentation

Zugehörige Informationen

Das Kapitel "Lerntext: Erste Schritte" auf Seite 1-1 enthält detaillierte Informationen zum Starten des Systems und zum Anmelden am Betriebssystem.

Der Abschnitt zum Befehl **login** enthält detaillierte Informationen zum Anmeldevorgang.

Der Abschnitt "Überblick über das An- und Abmelden" auf Seite 2-33 beschreibt das Anmelden am und Abmelden vom Betriebssystem.

Der Abschnitt zum Befehl **passwd** beschreibt das Erstellen und Ändern des Kennwortes.

Abmelden vom Betriebssystem

Abmelden und aktiviertes System verlassen

Folgende Tastenfolge drücken:

Strg-D (für DATEIENDE)

ODER bei der Eingabeaufforderung folgenden Befehl eingeben:

```
logout.
```

Nach dem Abmelden zeigt das System die Eingabeaufforderung `login` für den nächsten Benutzer an.

Betriebssystem stoppen

Benutzer mit Superuser-Berechtigung können mit dem Befehl **shutdown** das Betriebssystem stoppen. Das System darf nicht vor Ausführen des Befehls **shutdown** ausgeschaltet werden, da es sonst zu Datenverlust kommen kann. Wenn der Benutzer keine Berechtigung für den Befehl **shutdown** hat, meldet er sich einfach vom System ab und läßt es aktiviert.

Betriebssystem stoppen durch Eingabe von:

```
shut down
```

Ist die Ausführung des Befehls **shutdown** beendet und das Betriebssystem inaktiviert, wird folgende Nachricht ausgegeben:

```
....Halt beendet....
```

Empfohlene Dokumentation

Zugehörige Informationen

Informationen zur Tastenfolge Strg-D für DATEIENDE enthält der Abschnitt "7. Schritt: Spezielle Tastenfunktionen benutzen" auf Seite 1-7.

Im Abschnitt zum Befehl **shutdown** wird das Stoppen des Betriebssystems beschrieben.

Der Abschnitt zum Befehl **logout** enthält detaillierte Informationen zum Abmelden vom Betriebssystem.

Anmeldenachrichten unterdrücken

Diese Prozedur beschreibt, wie Anmeldenachrichten an den Benutzer durch Erstellen einer Datei **.hushlogin** im Benutzerverzeichnis unterdrückt werden.

Nach dem erfolgreichen Anmelden werden vom Befehl **login** Tag, Datum und Uhrzeit des letzten erfolgreichen und nicht erfolgreichen Anmeldeversuchs sowie die Gesamtzahl der nicht erfolgreichen Anmeldeversuche für dieses Konto seit der letzten Änderung der Daten für die Identifikationsüberprüfung (normalerweise ein Kennwort) angezeigt. Diese Nachrichten werden unterdrückt, wenn im Benutzerverzeichnis eine Datei **.hushlogin** vorhanden ist.

Erforderliche Schritte

1. Die Anmeldung für den Benutzer durchführen, für den Nachrichten unterdrückt werden sollen. Um z. B. Anmeldenachrichten für den Benutzer `kerry` zu unterdrücken, mit der Benutzer-ID `kerry` am System anmelden.
2. `cd /u/kerry` eingeben und die Eingabetaste drücken, um ins Benutzerverzeichnis zu wechseln.
3. `touch .hushlogin` eingeben und die Eingabetaste drücken, um die Datei **.hushlogin** im Benutzerverzeichnis zu erstellen.

Beim nächsten Anmelden des Benutzers werden alle Anmeldenachrichten unterdrückt. Das System kann so eingerichtet werden, daß alle Anmeldenachrichten außer dem Tag unterdrückt werden. Dazu entsprechend der verwendeten Shell eine der folgenden Dateien editieren:

- Bei Verwendung der C-Shell die Datei **.login** editieren.
- Bei Verwendung der Korn- oder der Bourne-Shell die Datei **.profile** editieren.

4. Anschließend mit Hilfe eines Systemeditors folgendes am Anfang der Datei **.login** bzw. **.profile** einfügen:

```
cat /etc/motd
```

Beim nächsten Anmelden des Benutzers werden alle Anmeldenachrichten außer dem Tag unterdrückt.

Überblick über virtuelle Datenstationen

Mit der Einrichtung "virtuelle Datenstation" stehen dem Benutzer mehrere logische Datenstationen auf einer physischen Datenstation zur Verfügung.

Bei einer regulären Anmeldesitzung wird ein Befehl eingegeben, den das Betriebssystem verarbeitet; danach kann ein anderer Befehl eingegeben werden. Das Betriebssystem kann jedoch mehrere Befehle gleichzeitig ausführen. Stehen an einem System mehrere physische Datenstationen zur Verfügung, kann an jeder Datenstation ein Befehl eingegeben werden, bis auf jeder Datenstation ein Befehl ausgeführt wird. Die Einrichtung "virtuelle Datenstation" ermöglicht bei der Anmeldesitzung das Öffnen von maximal 16 virtuellen Datenstationen an einer physischen Datenstation und führt Befehle auf jeder virtuellen Datenstation aus, ohne eine andere physische Datenstation zu benötigen. Jede virtuelle Datenstation stellt bei einer Anmeldung eine separate Datenstation dar. Mit dem Befehl **chnumvt** kann die Anzahl der virtuellen Datenstationen geändert werden.

Eine virtuelle Datenstation ist mit einem Fenster vergleichbar, über das mit dem Betriebssystem gearbeitet werden kann. In jedem geöffneten Fenster können Aufgaben ausgeführt werden. Einer virtuellen Datenstation sind eigene Prozesse zugeordnet, die die gesamte Datenstation belegen.

Mit dem Befehl **open ksh** wird eine virtuelle Datenstation mit dem standardmäßigen Befehls-Interpreter des Betriebssystems (die Shell) geöffnet. Soll eine virtuelle Datenstation mit einem anderen Befehls-Interpreter geöffnet werden, anstelle von **ksh** den Namen dieses Befehls-Interpreters angeben.

Hinweise:

1. Das Arbeiten mit virtuellen Datenstationen ist nur auf grafikfähigen Datenstationen möglich.
2. Einige Systemnachrichten, die eventuell bei einer Anmeldesitzung angezeigt werden, erscheinen nur an der Systemkonsole der physischen Datenstation.

Empfohlene Dokumentation

Zugehörige Informationen


Im Abschnitt "Mit virtuellen Datenstationen arbeiten" auf Seite 2-40 werden die notwendigen Schritte zum Öffnen, Verwenden und Schließen der virtuellen Datenstationen beschrieben.

Mit dem Befehl **chnumvt** kann die Zahl der virtuellen Datenstationen geändert werden.

Der Abschnitt zum Befehl **open** enthält Informationen zum Öffnen virtueller Datenstationen.

Tastenfunktionen des Betriebssystems AIX

Tastenfolge	Funktion
Strg-C	Unterbrechen.
Strg-D	Dateiende.
Strg-\	Verlassen.
Esc	Escape.
Rückschrittaste	Voriges Zeichen löschen.
Strg-W	Voriges Wort löschen.
Strg-U	Gesamte Zeile löschen.



Strg-S	Ausgabe stoppen.
Strg-Q	Ausgabe wiederaufnehmen.
Alt-Ausf	Zu nächster virtueller Datenstation.
Umsch-Ausf	Zu voriger virtueller Datenstation.
Strg-Z	Zeitweilig aussetzen.
Strg-Y	Aussetzen mit Verzögerung.
Strg-R	Fehlerhafte Eingabezeile erneut anzeigen.
Strg-O	Ausgabe bis zum nächsten Lesen löschen.
Strg-V	Nächstes Literal.

Mit virtuellen Datenstationen arbeiten

Erforderliche Schritte

1. Zum Starten einer virtuellen Datenstation folgendes eingeben:

```
open ksh
```

2. Nachdem eine virtuelle Datenstation geöffnet wurde, können Befehle wie gewohnt eingegeben werden. Sind mehrere virtuelle Datenstationen geöffnet, kann der Benutzer durch Drücken der Tastenfolge Alt-Ausf (NÄCHSTES FENSTER) von der aktuellen virtuellen Datenstation zur nächsten gelangen.

Zurückkehren zur vorigen virtuellen Datenstation mit der Tastenfolge Umsch-Ausf (Vorheriges Fenster).

Die Tastenfolgen NÄCHSTES FENSTER (Alt-Ausf) und VORIGES FENSTER (Umsch-Ausf) ermöglichen das Hin- und Herspringen zwischen virtuellen Datenstationen, und zwar in der Reihenfolge, in der sie geöffnet wurden (Datenstationsring). Auf dem System können maximal 16 virtuelle Datenstationen gleichzeitig geöffnet sein.

3. Zum Schließen (Stoppen) einer virtuellen Datenstation die Tastenfolge Strg-D (DATEIENDE) verwenden.

Hinweis: Vor dem Abmelden alle geöffneten virtuellen Datenstationen schließen.

Empfohlene Dokumentation

Der Abschnitt "Überblick über virtuelle Datenstationen" auf Seite 2-38.

Zugehörige Informationen

Der Abschnitt "Tastenfunktionen des Betriebssystems AIX" auf Seite 2-38 enthält Informationen zu den Tastenfolgen Alt-Ausf (NÄCHSTES FENSTER); Umsch-Ausf (VORHERIGES FENSTER) und Strg-D (DATEIENDE).

Der Abschnitt zum Befehl **open** enthält Informationen zum Öffnen virtueller Datenstationen.

Überblick über Befehle

Ein Befehl ist eine Anforderung zum Ausführen einer Operation oder eines Programms. Das Betriebssystem wird mit Befehlen angewiesen, bestimmte Aufgaben auszuführen. Eingegebene Befehle werden von einem Befehls-Interpreter (auch "Shell" genannt) entschlüsselt und die entsprechende Aufgabe wird ausgeführt.

Befehle können aus einem Wort bestehen oder aber als Kombination mehrerer Befehlsnamen eingegeben werden. Wird z. B. die Ausgabe eines Befehls als Eingabe für einen anderen Befehl verwendet, so ist dies eine sogenannte Pipe.

Durch Optionen können Befehlsoperationen noch genauer definiert werden. Eine Option wird zusammen mit dem Befehlsnamen in die Befehlszeile eingegeben. Vor einer Option wird gewöhnlich ein Minuszeichen angegeben.

Befehle können auch zusammengestellt und in einer Datei gespeichert werden. Solche Befehlsdateien werden als Shell-Prozeduren bezeichnet. Anstelle der einzelnen Befehle wird die Datei, die die Befehle enthält, ausgeführt.

Einige Befehle können mit der Einrichtung SMIT (System Management Interface Tool) aufgebaut werden. Mit Hilfe der SMIT-Menüs können Befehle zur Ausführung von Systemverwaltungsaufgaben aufgebaut werden.

Weitere Informationen zur Einrichtung SMIT können dem Abschnitt "System Management Interface Tool (SMIT) Overview" im Handbuch *System Management Guide* entnommen werden.

Syntaxdiagramme

Die Syntax von Befehlen wird durch Syntaxdiagramme und Verwendungsanweisungen dargestellt.

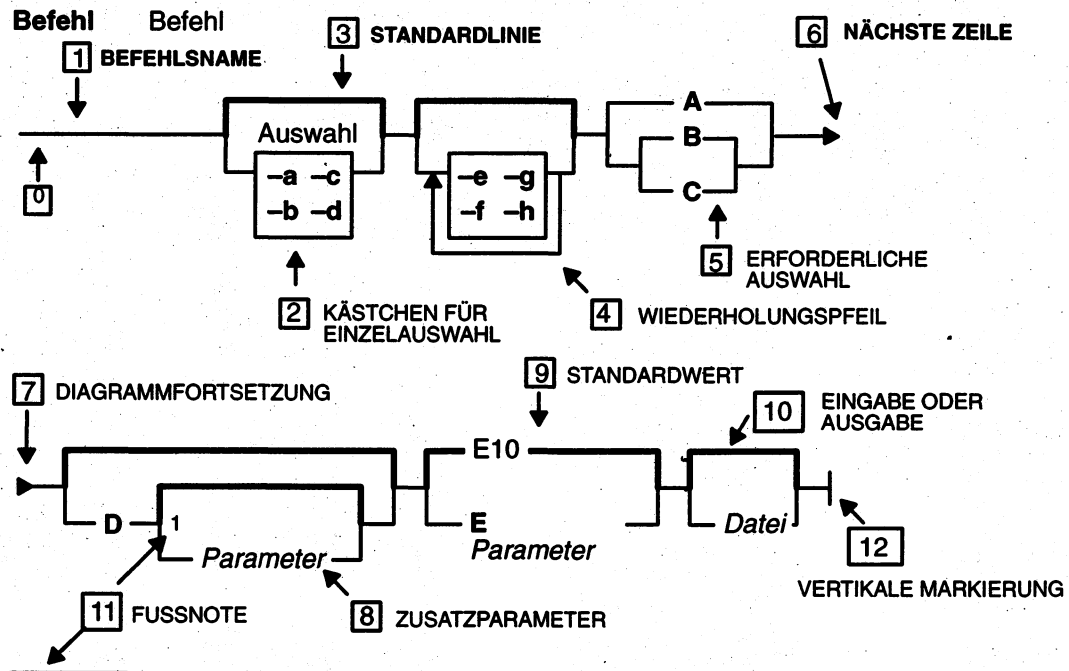
Syntaxdiagramme dienen zum Darstellen von Informationen über die Eingabe von Befehlen in die Befehlszeile. Das Syntaxdiagramm kann folgende Informationen enthalten:

- Optionen, die in die Befehlszeile eingegeben werden können
- Optionen, für die ein Parameter erforderlich ist
- Optionen, die einen Zusatzparameter haben
- Standardwerte der Optionen und Parameter (falls festgelegt)
- Optionen, die sich gegenseitig ausschließen
- Angaben über erforderliche oder wahlfreie Eingabe von Optionen oder Parametern
- Angaben über die Wiederholung von Options- und Parameterfolgen

Für die Syntaxdiagramme von Befehlen gelten folgende Konventionen:

- Diagrammelemente, die genau so in die Befehlszeile einzugeben sind, sind **fettgedruckt** (z. B. Befehlsnamen, Optionen und Literale).
- Diagrammelemente, die Variablen für einen Namen sind, werden in *Kursivschrift* dargestellt (z. B. Parameter, die nach Optionen und Parametern angegeben werden und vom Befehl gelesen werden, wie *Dateien* und *Verzeichnisse*).
- Standardwerte, deren Eingabe nicht erforderlich ist, werden in Normalschrift in der Standardpfadlinie (**Fettdruck**) dargestellt.

Folgendes Diagramm soll exemplarisch die in Syntaxdiagrammen verwendeten Konventionen verdeutlichen. Die einzelnen Teile des Diagramms sind bezeichnet und mit Nummern versehen und werden im Anschluß an das Diagramm erläutert.



¹ Zwischen diesen Angaben kein Leerzeichen eingeben.

Abbildung 6. Beispielsyntaxdiagramm

Erläuterungen zum Beispieldiagramm:

0 PFADLINIE Das Syntaxdiagramm beginnt mit der Pfadlinie.

1 BEFEHLSNAME

Das erste Element im Diagramm ist der Name des aufzurufenden Befehls. Es wird in Fettdruck dargestellt und muß daher unverändert eingegeben werden.

Nach dem Befehlsnamen verzweigt sich der Pfad. Eine dieser beiden Pfadalternativen ist auszuwählen. Der untere Pfad wird unter Punkt 2, der obere unter Punkt 3 beschrieben.

2 KÄSTCHEN FÜR EINZELAUSWAHL

Auf der unteren Pfadlinie befindet sich ein Kästchen mit der Überschrift *Auswahl*. Aus einem solchen Kästchen kann nur ein Parameter ausgewählt werden.

3 STANDARDLINIE

Bei Auswahl der oberen Pfadlinie wird das Auswahlkästchen umgangen und nichts eingegeben. Die fettgedruckte Linie oberhalb des Kästchens gibt den Standardpfad an, d. h. bei Auswahl dieses Pfades sind an dieser Stelle des Diagramms keine Eingaben erforderlich. Ausnahmen werden normalerweise unter "Beschreibung" erläutert. Eine wichtige Ausnahme, die leere Standardlinie um Eingabe- und Ausgabedateien, wird unter Punkt 10 erklärt.

4 WIEDERHOLUNGSPFEIL

Befindet sich auf der ausgewählten Pfadlinie ein von einem Pfeil eingerahmtes Kästchen, muß mindestens einer der möglichen Parameter im Kästchen ausgewählt werden. Danach kann entweder dem Wiederholungspfeil gefolgt und eine weitere Auswahl getroffen oder der

Eingabepfad fortgesetzt werden. Wird einem Pfeil gefolgt, der genau bis zum Anfang des Kästchens zurückführt (und nicht einem Pfeil, der mehrere Zweige des Diagramms umfaßt), darf die gleiche Auswahl nicht mehr als einmal getroffen werden.

5 ERFORDERLICHE AUSWAHL

Auf den Zweig mit dem Wiederholungspfeil folgt ein Zweig mit drei Auswahlmöglichkeiten, die nicht durch eine Standardpfadlinie umgangen werden können. Es muß also A, B oder C ausgewählt werden.

6 NÄCHSTE ZEILE

Ist ein Diagramm so lang, daß es nicht in eine Zeile paßt, gibt diese Kennzeichnung an, daß mit der nächsten Zeile des Diagramms die Eingabe der Befehlsfolge fortgesetzt wird. Das Ende des Diagramms wird erst durch die vertikale Markierung angezeigt.

7 DIAGRAMMFORTSETZUNG

Dieses Zeichen gibt an, wo das Diagramm nach der Unterbrechung in der vorherigen Zeile fortgesetzt wird.

8 ZUSATZPARAMETER

Wenn eine Option einen Parameter haben kann, aber nicht muß, verzweigt sich der Pfad nach der Option. Darf zwischen Option und Parameter kein Leerzeichen eingegeben werden, wird in einer Fußnote darauf hingewiesen.

9 STANDARDWERT

Oft sind einem Befehl Standardvorgaben oder -werte zugeordnet, die wirksam werden, wenn vom Bediener nichts anderes eingegeben bzw. keine Auswahl getroffen wird. Diese Standardwerte werden in Normalschriftart in der Standardlinie angegeben, wenn sie einer möglichen Bedienereingabe in der Befehlszeile entsprechen (z. B. einer Auswahl mit einem Wert). Kann der Standard nicht vom Bediener in der Befehlszeile eingegeben werden, wird er auch im Diagramm nicht angegeben.

Hinweis: Standardwerte werden zur Information des Bedieners in der Abbildung angegeben. Es ist nicht erforderlich, diese Werte in der Befehlszeile einzugeben.

10 EINGABE ODER AUSGABE

Bei Befehlen, die sowohl Eingabedateien als auch von der Standardeingabeeinheit lesen können, verläuft eine leere Standardpfadlinie oberhalb des Dateiparameters. Kann der Befehl seine Ausgabedaten in eine Ausgabedatei oder auf die Standardausgabeeinheit schreiben, zeigt das Diagramm für diesen Befehl ebenfalls eine leere Standardlinie über dem Ausgabedateiparameter.

Kann ein Befehl nur von der Standardeingabeeinheit lesen, ist im Diagramm keine Eingabedatei angegeben. Schreibt der Befehl nur auf die Standardausgabeeinheit, ist im Diagramm keine Ausgabedatei angegeben, da die Ausgabe standardmäßig auf die Standardausgabeeinheit erfolgt.

Muß für Eingabe oder Ausgabe ein Dateiname eingegeben werden, ist im Diagramm der Dateiparameter ohne leere Standardlinie darüber angegeben.

11 FUSSNOTE Gelten für einen Befehl spezielle Erfordernisse oder Einschränkungen, wird in einer Fußnote darauf hingewiesen.

12 VERTIKALE MARKIERUNG

Die vertikale Markierung kennzeichnet das Ende des Syntaxdiagramms.

Verwendungsanweisungen

Verwendungsanweisungen sind eine andere Art der Darstellung für die Befehlsyntax. Wie Syntaxdiagramme dienen Verwendungsanweisungen zur Erläuterung der Eingabe von Befehlen in der Befehlszeile. Verwendungsanweisungen werden nicht im Diagrammformat dargestellt, enthalten aber dieselben Informationen wie Syntaxdiagramme. Sie bestehen eher aus Zeichen, wie z. B. eckige Klammern ([]), geschweiften Klammern ({ }) und vertikalen Strichen (|). Als Beispiel folgt die Verwendungsanweisung für den Befehl **unget**:

```
unget [ -n ] [ -rSID ] [ -s ] { Datei ... | - }
```

Es gelten dieselben Konventionen bezüglich **Fettdruck** und *Kursivschrift* wie beim Syntaxdiagramm. Für die Befehlsbenutzungsanweisung gelten folgende zusätzliche Konventionen:

- Parameter in eckigen Klammern sind wahlfrei.
- Parameter in geschweiften Klammern sind erforderlich.
- Nicht in Klammern stehende Parameter sind erforderlich.
- Ein vertikaler Strich bedeutet, daß nur ein Parameter gewählt werden kann; z.B. bedeutet [a | b], daß a oder b ausgewählt werden *kann*; entsprechend bedeutet { a | b }, daß a oder b *erforderlich* ist.
- Auslassungszeichen (. . .) bedeuten, daß die Parameter in der Befehlszeile wiederholt werden können.
- Der Bindestrich (-) steht für die Standardeingabeeinheit.

Befehle im Hintergrund ausführen

Soll ein Befehl ausgeführt werden, dessen Verarbeitung sehr lange dauert, kann festgelegt werden, daß der Befehl im Hintergrund ausgeführt werden soll. Die Hintergrundverarbeitung empfiehlt sich bei Prozessen, die langsam ablaufen. Zum Ausführen eines Befehls im Hintergrund am Ende des Befehls das Zeichen & verwenden:

Befehl&

Wird der Befehl im Hintergrund ausgeführt, kann die Arbeit am System fortgesetzt werden, d. h. im Vordergrund können weitere Befehle eingegeben werden.

Neben der Verarbeitung im Hintergrund können Befehle auch auf einer separaten virtuellen Datenstation ausgeführt werden. Durch das Öffnen einer weiteren virtuellen Datenstation kann ein Befehl an einer virtuellen Datenstation verarbeitet werden, während der Benutzer an seiner Datenstation weiterarbeitet.

Es kann auch gelegentlich erforderlich sein, einen Befehl zu einem bestimmten Zeitpunkt (Uhrzeit oder Datum) auszuführen. Mit dem Dämon **cron** kann die automatische Ausführung von Befehlen zu einem festgelegten Zeitpunkt erfolgen. Mit den Befehlen **at** und **batch** können Befehle zu einem späteren Zeitpunkt, oder sobald die Systembelastung dies zuläßt, ausgeführt werden.

Befehle eingeben

Beim Arbeiten mit dem Betriebssystem werden Befehle gewöhnlich nach der Shell-Eingabeaufforderung in der Befehlszeile eingegeben. Die Shell-Eingabeaufforderung kann unterschiedlich aussehen. In den folgenden Beispielen ist das Dollarzeichen (\$) die Eingabeaufforderung.

Soll z. B. der Inhalt des aktuellen Verzeichnisses angezeigt werden, **ls** eingeben und die Eingabetaste drücken:

```
$ ls
```

Während der Ausführung eines Befehls wird die Shell-Eingabeaufforderung nicht angezeigt. Sobald die Befehlsausführung beendet ist, wird die Shell-Eingabeaufforderung erneut angezeigt. Dann können weitere Befehle eingegeben werden.

Das allgemeine Format für die Eingabe von Befehlen sieht folgendermaßen aus:

Befehl Option(en) Parameter

Optionen beeinflussen die Art der Befehlsausführung. Bei vielen Befehlen stehen mehrere Optionen zur Verfügung. Wird z. B. die Option **-l** (lang) mit dem Befehl **ls** eingegeben, zeigt das System zusätzliche Informationen über den Inhalt des aktuellen Verzeichnisses an. Es folgt ein Beispiel für die Verwendung des Befehls **ls**:

```
$ ls -l
```

Ein Parameter ist eine Zeichenfolge, die auf einen Befehl oder eine Option folgt. Mit ihr werden Daten angegeben, z. B. Namen von Dateien oder Verzeichnissen oder Werte. Im folgenden Beispiel ist das Verzeichnis **/usr/bin** ein Parameter:

```
$ ls -l /usr/bin
```

Beim Eingeben von Befehlen sind folgende Konventionen zu beachten:

- Befehle werden in Kleinbuchstaben eingegeben.
- Optionen wird ein - (Minuszeichen) vorangestellt.
- In der Befehlszeile können mehrere Befehle eingegeben werden, diese durch ein ; (Semikolon) getrennt werden.
- Lange Befehlsfolgen können in der nächsten Zeile fortgesetzt werden. Hierzu den umgekehrten Schrägstrich \ am Ende der ersten Zeile angeben. Die Platzierung des umgekehrten Schrägstrichs wird aus dem nachfolgenden Beispiel ersichtlich:

```
$ cat /usr/ust/mydir/mydata > \  
/usr/usts/yourdir/yourdata
```

Beim Eingeben einiger Befehle wird eine andere Eingabeaufforderung angezeigt. Da einige der Befehle (z. B. der Befehl **telnet**) eigentlich Programme sind, ändert sich die Eingabeaufforderung bei der Ausführung des Befehls. Alle innerhalb von Programmen eingegebenen Befehle werden Unterbefehle genannt. Nach Beenden des Programms wird erneut die Shell-Eingabeaufforderung angezeigt.

Das Betriebssystem kann mit verschiedenen Shells (z. B. Bourne, C oder Korn) arbeiten. Der Benutzer muß daher wissen, welche Shell verwendet wird, um die Befehle im richtigen Format eingeben zu können.

Befehle abbrechen

Wurde ein Befehl eingegeben, dessen Ausführung abgebrochen werden soll, die Tastenfolge Strg-C oder Alt-Pause verwenden. Nach dem erfolgreichen Abbrechen der Befehlsausführung wird die Shell-Eingabeaufforderung erneut angezeigt, und ein weiterer Befehl kann eingegeben werden.

Empfohlene Dokumentation

Zugehörige Informationen

Im Kapitel "Überblick über Shells" auf Seite 2-21 wird beschrieben, was Shells sind, welche verschiedenen Arten von Shells es gibt und wie Shells die verschiedenen Möglichkeiten der Befehls-Interpretation beeinflussen.

Im Abschnitt "Überblick über Prozesse" auf Seite 2-14 wird beschrieben, was Prozesse sind und wie sie gestartet und gestoppt werden können.

Im Abschnitt "Überblick über virtuelle Datenstationen" auf Seite 2-38 wird beschrieben, wie auf einer physischen Datenstation mehrere virtuelle Datenstationen geöffnet werden können.

Im Abschnitt "Überblick über Ein- und Ausgabe" auf Seite 2-28 wird beschrieben, wie das Betriebssystem Eingabe und Ausgabe verarbeitet.

Im Kapitel "System Management Interface Tool (SMIT) Overview" im Handbuch *System Management Guide* werden die Grundfunktionen der Schnittstelle SMIT beschrieben.

Im Abschnitt zum Befehl `at` wird beschrieben, wie Befehle zu einem späteren Zeitpunkt ausgeführt werden können.

Im Abschnitt zum Befehl `batch` wird beschrieben, wie Befehle ausgeführt werden können, wenn die Systembelastung dies zuläßt.

Im Abschnitt zum Dämon `cron` wird beschrieben, wie Befehle automatisch ausgeführt werden können.

Kapitel 3. Dateien, Verzeichnisse und allgemeine Aufgaben

Dieses Kapitel ist eine Einführung in Dateien und Verzeichnisse und erläutert anhand von Beispielen, wie mit ihnen gearbeitet wird. Außerdem enthält es Informationen und Prozeduren zum Sichern und Drucken sowie zu weiteren allgemeinen Aufgaben, die der Systembenutzer kennen sollte.

Unter anderem werden in diesem Kapitel folgende Themen behandelt:

- Dateien und Verzeichnisse
- Überblick über den Zugriff auf Dateien und Verzeichnisse
- Dateiprozeduren
 - Dateiverwaltung
 - Text in einer Datei bearbeiten
 - DOS-Dateien umwandeln
- Mit Verzeichnissen arbeiten
- Dateien sichern
- Drucken
- Allgemeine Verarbeitungsvorgänge

Das Handbuch *System Management Guide* enthält zusätzliche Informationen zur Dateiverwaltung.

Das Handbuch *General Programming Concepts* enthält Programmierhinweise zu verschiedenen Abschnitten.

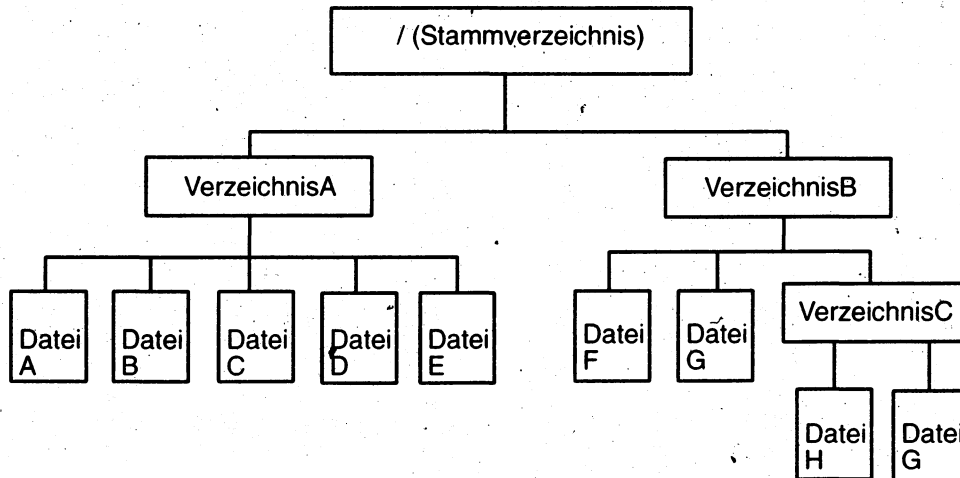
Überblick über Dateien und Verzeichnisse

Eine Datei ist eine Zusammenstellung von Daten, die gelesen werden kann oder in die Daten geschrieben werden können. Dateien enthalten entweder ASCII-Zeichen, die vom Benutzer gelesen werden können, oder binäre Zeichen, die nur vom Computer gelesen werden können. Dateien können ausführbare Programme, einfachen Text, Softwareanwendungen, Druckerdefinitionen und Befehle enthalten, die vom Betriebssystem auf gleiche Weise behandelt werden.

Dateien werden in Verzeichnissen gespeichert. Durch die Zusammenfassung werden Organisation, Zugriffszeit und Flexibilität verbessert. Verzeichnisse enthalten Dateien oder Unterverzeichnisse oder eine Kombination aus beidem. Jede Datei und jedes Verzeichnis in einem Dateisystem hat einen eindeutigen Namen.

Dateisysteme bestehen aus Gruppen von Verzeichnissen und den in den Verzeichnissen enthaltenen Dateien. Ein Dateisystem wird im allgemeinen als umgekehrte Baumstruktur dargestellt. Das Stammverzeichnis, dargestellt durch einen Schrägstrich (/), definiert ein Dateisystem und steht in der Baumstruktur eines Dateisystems ganz oben. Die Verzeichnisse verzweigen sich vom Stammverzeichnis aus nach unten und enthalten entweder Dateien oder Unterverzeichnisse oder beides. Durch die Verzweigung entsteht für jedes Objekt im Dateisystem ein eindeutiger Pfad.

Der Pfadname für jede Datei und jedes Verzeichnis besteht aus den Namen aller Verzeichnisse, die in der Hierarchie darüber liegen. Die Abbildung "Beispiel für ein Dateisystem" zeigt den Aufbau eines Dateisystems.



Beispiel für ein Dateisystem

Alle Pfade in einem Dateisystem gehen vom Stammverzeichnis aus. Daher hat jede Datei im Dateisystem einen eindeutigen Bezug zum Stammverzeichnis, der als absoluter Pfadname bezeichnet wird. Absolute Pfadnamen beginnen mit einem Schrägstrich (/). Der absolute Pfadname zur Datei **DateiH** im Beispiel ist **/VerzeichnisB/VerzeichnisC/DateiH**. Im obigen Beispiel sind zwei Dateien mit dem Namen **DateiG**. Da der absolute Pfad für jede dieser Dateien verschieden ist, nämlich **/VerzeichnisB/DateiG** bzw. **/VerzeichnisB/VerzeichnisC/DateiG**, hat jede Datei **DateiG** einen eindeutigen Pfadnamen im Dateisystem. Jeder Bestandteil eines Pfads außer dem letzten ist ein Verzeichnis. Der letzte Bestandteil eines Pfadnamens kann ein Dateiname sein.

Folgende Abschnitte enthalten weitere Informationen zu Dateien und Verzeichnissen:

- Im Abschnitt "Einführung in Dateien" (auf Seite 3-3) wird erklärt, welche verschiedenen Dateitypen es gibt, wie sie arbeiten und welche Konventionen beim Benennen von Dateien zu beachten sind.
- Der Abschnitt "Mit Dateien arbeiten" (auf Seite 3-5) bietet eine Einführung zu den Befehlen, mit denen Dateien bearbeitet werden.
- Im Abschnitt "Einführung in Verzeichnisse" (auf Seite 3-8) werden die Pfadnamen, die interne Darstellung von Verzeichnissen und die verschiedenen Verzeichnistypen erklärt.
- Der Abschnitt "Mit Verzeichnissen arbeiten" (auf Seite 3-11) bietet eine Einführung zu den Befehlen, mit denen Verzeichnisse bearbeitet werden.
- Der Abschnitt "Einführung in die Verbindungsfunktion" (auf Seite 3-11) bietet eine Einführung zum Verbinden von Dateien.
- Der Abschnitt "Einführung in den Zugriff auf Dateien und Verzeichnisse" (auf Seite 3-13) bietet eine Einführung zu den Berechtigungen und zur Ausgabe der Auflistungsbefehle **ll** und **ls**.

Das Kapitel "Understanding JFS i-nodes" im Handbuch *General Programming Concepts* beschreibt und erklärt die im Dateiindex enthaltenen Informationen.

Einführung in Dateien

Eine Datei ist eine Zusammenstellung von Daten, die gelesen werden kann oder in die Daten geschrieben werden können. Eine Datei kann aus einem erstellten Programm, geschriebenem Text, gesammelten Daten oder einer verwendeten Einrichtung bestehen. In Dateien sind Befehle, Druckerspools, Anschlüsse, Korrespondenz- und Anwendungsprogramme gespeichert. Das erlaubt dem Benutzer, auf verschiedene Elemente des Systems auf gleiche Weise zuzugreifen und verleiht dem Dateisystem größere Flexibilität. Näheres wird in den folgenden Abschnitten beschrieben:

- Einführung
- Dateitypen
- Konventionen für das Benennen von Dateien

Einführung

Dateien werden im Betriebssystem für jede Ein- und Ausgabe (E/A) von Informationen verwendet. Dadurch wird der Zugriff sowohl auf Hardware als auch auf Software standardisiert. Die Eingabe erfolgt, indem der Inhalt einer Datei geändert oder geschrieben wird. Die Ausgabe erfolgt, indem der Inhalt einer Datei gelesen oder zu einer anderen Datei übertragen wird. Um z. B. eine Datei auszudrucken, liest das System die Informationen aus der Textdatei und schreibt sie in die Druckerdatei.

Dateigruppen werden in Verzeichnissen gespeichert. Diese Dateigruppen stehen oft in Bezug zueinander; durch das Speichern in Verzeichnisstrukturen bleibt dieser Bezug erhalten.

Es gibt viele Möglichkeiten, um Dateien zu erstellen, sie zu verwenden und zu ändern. Der Abschnitt "Mit Dateien arbeiten" auf Seite 3-5 bietet eine Einführung in die Befehle, mit denen Dateien bearbeitet werden.

Dateitypen

Es gibt drei grundlegende Dateitypen:

reguläre Dateien

Speichern Daten (Text, binäre und ausführbare).

Verzeichnisdateien

Enthalten Informationen für den Zugriff auf andere Dateien.

Sonderdateien

Definieren eine FIFO- (First In/First Out) Datei oder eine physische Einheit.

Sämtliche vom System identifizierten Dateitypen fallen unter eine dieser Kategorien. Das Betriebssystem verwendet jedoch viele Variationen dieser grundlegenden Dateitypen.

Reguläre Dateien sind die am häufigsten verwendeten Dateien. Wird mit Hilfe eines Textverarbeitungsprogramms ein Dokument erstellt, ist sowohl das Programm als auch das Dokument in regulären Dateien enthalten.

Reguläre Dateien enthalten entweder ASCII- oder Binärzeichen. ASCII-(Text-)Dateien können vom Benutzer gelesen werden. Binärdateien können nur vom Computer gelesen werden. Binärdateien können ausführbare Dateien sein, die das System anweisen, einen Job auszuführen. In ausführbaren Binärdateien sind Befehle, Shell-Prozeduren und Programme gespeichert. ASCII-Text wird von Compilerprogrammen in Binärcode übersetzt.

Verzeichnisdateien enthalten Informationen, die das System für den Zugriff auf sämtliche Dateitypen benötigt, sie enthalten jedoch nicht den Dateiinhalt selbst. Folglich benötigen Verzeichnisdateien weniger Speicherplatz als eine reguläre Datei, wodurch die Dateisystemstruktur flexibler und umfangreicher sein kann. Jeder Verzeichniseintrag besteht entweder aus einer Datei oder einem Unterverzeichnis und enthält den Namen der Datei sowie ihre Dateiindexnummer (Dateiindex). Die Dateiindexnummer weist auf den eindeutigen Dateiindex hin, der der Datei zugeordnet ist. Der Dateiindex enthält Angaben zum Standort der in der Datei enthaltenen Daten. Verzeichnisdateien werden mit eigenen Befehlen erstellt und bearbeitet. Weitere Informationen enthält der Abschnitt "Einführung in Verzeichnisse" auf Seite 3-8.

Sonderdateien definieren die Einheiten für das System oder die von Prozessen erstellten temporären Dateien. Es gibt drei grundlegende Sonderdateitypen: FIFO-(First In/First Out), Block- und Zeichendateien. FIFO-Dateien werden auch Pipes genannt. Pipes werden von einem Prozeß erstellt und ermöglichen eine vorübergehende Übertragung zu einem anderen Prozeß. Diese Dateien sind nach Beendigung des ersten Prozesses nicht mehr vorhanden. Block- und Zeichendateien definieren physische Einheiten.

Jede Datei beinhaltet Berechtigungen (Zugriffsmodus), die festlegen, wer zum Lesen, Ändern und Ausführen einer Datei berechtigt ist. Näheres zu den Dateizugriffsmodi steht im Abschnitt "Einführung in den Zugriff auf Dateien und Verzeichnisse" auf Seite 3-13.

Konventionen beim Benennen von Dateien

Jede Datei muß innerhalb ihres Verzeichnisses über einen eindeutigen Namen verfügen. Dadurch wird sichergestellt, daß die Datei innerhalb des Dateisystems auch einen eindeutigen Pfadnamen besitzt. Folgende Richtlinien müssen beim Benennen von Dateien berücksichtigt werden:

- Ein Dateiname kann aus maximal 255 Zeichen bestehen und Buchstaben, Zahlen und Unterstreichungszeichen enthalten.
- Das Betriebssystem berücksichtigt die Groß- und Kleinschreibung, d. h. es unterscheidet zwischen Groß- und Kleinbuchstaben in Dateinamen. Daher handelt es sich bei DATEIA,

DaTeia und dateia um drei verschiedene Dateien, selbst wenn sie sich im selben Verzeichnis befinden.

- Dateinamen müssen so gewählt werden, daß so eindeutig wie möglich auf ihren Inhalt hingewiesen wird.
- Bei Verzeichnissen gelten dieselben Namenskonventionen wie bei Dateien.
- Manche Zeichen haben eine spezielle Bedeutung für das Betriebssystem und dürfen daher nicht verwendet werden. Dabei handelt es sich um folgende Zeichen:

`/\`" '* ; - ? [] () ~ ! $ { } < > # @ & |`

- Ein Dateiname ist bei der normalen Auflistung des Verzeichnisses verdeckt, wenn er mit einem Punkt (.) beginnt. Wird der Befehl `ll` mit der Option `-a` eingegeben, werden die verdeckten Dateien zusammen mit den regulären Dateien und Verzeichnissen aufgeführt.

Der Pfadname für jede Datei und jedes Verzeichnis besteht aus den Namen aller Verzeichnisse, die in der Hierarchie darüber liegen. Der letzte Bestandteil eines Pfadnamens kann ein regulärer Dateiname sein. Alle anderen Bestandteile eines Pfadnamens müssen Verzeichnisse sein. Pfadnamen können absolut oder relativ sein. Der Abschnitt "Einführung in Pfadnamen" auf Seite 3-9 enthält weitere Informationen zu Dateinamen im Dateisystem.

Mit Dateien arbeiten

Es gibt viele Möglichkeiten, mit Dateien auf dem System zu arbeiten. Einige Beispiele:

- Dateien erstellen, kombinieren und editieren
- Dateien anzeigen
- Dateien verschieben, kopieren und umbenennen
- Dateien löschen
- Dateien verbinden
- DOS-Dateien umsetzen
- Dateien sichern
- Andere Dateibefehle verwenden

Dateien erstellen, kombinieren und editieren

Das Betriebssystem stellt für die Erstellung und Änderung von Dateien mehrere Editoren zur Verfügung (`vi`, `ed` und andere). Mit diesen Editoren können Programm- oder Textdateien geschrieben, gesichert und überarbeitet werden. Weitere Informationen dazu beinhaltet das Kapitel "Überblick über Editoren" im Handbuch *Editierkonzepte und -prozeduren*.

Neue Dateien können auch durch Verschieben, Kopieren und Umbenennen bereits bestehender Dateien erstellt werden. Dies ist im Abschnitt "Dateien verschieben, kopieren und umbenennen" auf Seite 3-6 beschrieben.

Neue Dateien können auch durch Neuordnung von Dateiinhalten (vorhandene Dateien teilen oder verknüpfen) oder durch Umbenennen von Dateien erstellt werden. Der Befehl `cat` zusammen mit den Umleitungssymbolen (`>>`) kann verwendet werden, um den Inhalt einer Datei an das Ende einer weiteren Datei anzuhängen. Mit dem Befehl `csplit` kann eine Datei in mehrere Dateien aufgeteilt werden.

Dateien anzeigen

Mit dem Befehl `ll` oder `ls` kann in jedem Verzeichnis eine alphabetische Auflistung der Dateien und Unterverzeichnisse angezeigt werden. Durch das Verwenden von Optionen

können exakt Dateinamen, Dateityp, Zugriffsmodus, Dateieigner, Dateigröße und Zeitangaben, z. B. Zeit der letzten Änderung oder des letzten Zugriffs, angezeigt werden.

Mit den Befehlen **pg**, **more** und **page** kann der Inhalt von Dateien angezeigt und die Geschwindigkeit, mit der Dateien angezeigt werden, gesteuert werden. Der Befehl **cat** kann auch verwendet werden, um den Inhalt einer oder mehrerer Dateien auf dem Bildschirm anzuzeigen. Durch die Kombination des Befehls **cat** mit dem Befehl **pg** werden jeweils nur so viele Daten angezeigt, wie in einer Anzeige Platz haben.

Mit dem Befehl **bfs** können sehr große Dateien sehr schnell gesucht werden. Dieser Befehl beinhaltet auch Möglichkeiten zum Suchen von übereinstimmenden Ausdrücken. Mit dem Befehl **tail** können standardmäßig die letzten 10 Zeilen einer Datei angezeigt werden. Durch das Anfügen von Optionen kann die Zahl der angezeigten Zeilen eingestellt werden. Mit Hilfe dieser Funktion kann festgestellt werden, ob ein Prozeß beendet oder noch aktiv ist.

Dateien verschieben, kopieren und umbenennen

Mit dem Befehl **mv** oder **move** werden Dateien durch Umbenennen von einem Verzeichnis in ein anderes verschoben. Mit diesen Befehlen wird nur die Position der Datei im Dateisystem geändert. Nach dem Verschieben der Datei kann der ursprüngliche Dateiname nicht mehr verwendet werden.

Mit dem Befehl **cp** oder **copy** werden Dateien kopiert, ohne daß die ursprüngliche Datei verschoben wird. Dateien können innerhalb desselben Verzeichnisses in eine andere Datei oder in ein anderes Verzeichnis kopiert werden. Dieser Befehl ist vor allem beim Erstellen von Sicherungskopien wichtiger Dateien hilfreich. Ist bereits eine Datei mit gleichem Namen im Zielverzeichnis vorhanden, wird diese jedoch überschrieben. Daher ist es sinnvoll, der zu kopierenden Datei einen neuen Namen zu geben und sicherzustellen, daß im Zielverzeichnis nicht bereits eine Datei gleichen Namens vorhanden ist.

Dateien löschen

Wird eine Datei nicht mehr benötigt, kann sie mit dem Befehl **del**, **rm** oder **delete** gelöscht werden. Bei Eingabe des Befehls **del** muß der Benutzer bestätigen, daß die Datei wirklich gelöscht werden soll, damit keine versehentlichen Löschungen durchgeführt werden. Die Befehle **rm** und **delete** erfordern vor dem Löschen keine Bestätigung durch den Benutzer. Mit allen diesen Befehlen kann auch eine Gruppe von Dateien gelöscht werden, oder es können bestimmte zu löschende Dateien aus einer Liste ausgewählt werden.

Dateien verbinden

Durch das Verbinden von Dateien mit dem Befehl **ln** kann bequem mit gleichen Dateien gearbeitet werden, die an mehreren Stellen vorkommen. Link-Verbindungen werden hergestellt, indem Originaldateien mit verschiedenen Namen (Alias-Namen) versehen werden. Mit Hilfe von Link-Verbindungen können mehrere Programme auf eine große Datei, z. B. Datenbank oder Postliste, zugreifen, ohne daß von der großen Datei eine Kopie erstellt werden muß. Link-Verbindungen sparen nicht nur Speicherplatz; Änderungen, die in einer Datei vorgenommen werden, werden automatisch in allen Verzeichnissen wiedergegeben, in denen eine Link-Verbindung zu der Datei besteht. Weitere Informationen zum Verbinden von Dateien befinden sich im Abschnitt "Einführung in die Verbindungsfunktion" auf Seite 3-11.

DOS-Dateien umsetzen

Mit dem Befehl **dosread** werden DOS-Dateien-Dateien in das Betriebssystem importiert. Mit dem Befehl **doswrite** werden Dateien des Betriebssystems in das DOS-Dateien-Format umgesetzt. Mit dem Befehl **dosdel** werden DOS-Dateien-Dateien gelöscht. Mit dem Befehl **dosdir** wird das DOS-Dateien-Verzeichnis aufgelistet.

Dateien sichern

Dateien können relativ leicht (versehentlich oder absichtlich) geändert oder gelöscht werden. Deshalb sollten immer Sicherungskopien von Dateien auf Diskette oder Band erstellt werden. Der Befehl **backup** wird dabei am häufigsten verwendet. Wenn Daten auf dem System verlorengehen oder beschädigt werden, können sie von der Sicherungskopie zurückgeschrieben werden. Werden die Dateien mit dem Befehl **backup** auf Band oder Diskette kopiert, werden sie mit Hilfe des Befehls **restore** wieder zurückgeschrieben.

Andere Dateibefehle verwenden

Der Befehl **which** sucht nach Dateien, die ausführbare Programme oder Shell-Prozeduren enthalten, und listet die Standorte der Dateien in der Standardausgabeeinheit auf. Der Befehl **find** sucht Verzeichnisse, die wie Dateien aussehen, und sucht nach Verzeichnissen, die der Benutzer nach Kriterien spezifiziert hat. Der Befehl **touch** aktualisiert die Zeitmarke einer Datei. (Die Zeitmarke einer Datei gibt an, wann die letzte Änderung oder der letzte Zugriff auf eine Datei erfolgte.) Dieser Befehl ist hilfreich, wenn der Computer so installiert ist, daß er Dateien löscht, die innerhalb einer bestimmten Zeit nicht geändert wurden. Der Befehl **cmp** vergleicht den Inhalt von Dateien. Diese Funktion ist hilfreich beim Sichern von Dateien, da nur die Dateien neu gesichert werden müssen, die sich seit der letzten Sicherung geändert haben.

Einführung in Verzeichnisse

Ein Verzeichnis ist ein bestimmter Dateityp, der nur die Informationen enthält, die für den Zugriff auf Dateien und andere Verzeichnisse erforderlich sind. Folglich beansprucht ein Verzeichnis weniger Speicherbereich als andere Dateitypen. Dies ermöglicht eine flexiblere und umfangreichere Dateisystemstruktur. Verzeichnisse fassen Dateien und andere Verzeichnisse in Gruppen zusammen und ermöglichen dadurch einen modularen Aufbau des Dateisystems. Im Gegensatz zu anderen Dateitypen werden zum Arbeiten mit Verzeichnissen spezielle Befehle benötigt.

Verzeichnisse enthalten Verzeichniseinträge; jeder Eintrag enthält den Namen einer Datei oder eines Unterverzeichnisses sowie ihre Dateiindexnummer (Dateiindex). Um die Zugriffszeit und die Auslastung des Speicherbereichs zu verbessern, werden die Daten einer Datei im Speicher des Computers an verschiedenen Standorten gespeichert. Der Dateiindex enthält die Adressen, die zum Auffinden der verstreuten Datenblöcke notwendig sind. Der Dateiindex enthält zusätzliche Informationen zu den Dateien, wie Zeit der letzten Änderung und des letzten Zugriffs, Zugriffsmodi, Zahl der Link-Verbindungen, Dateieigner und Dateityp. Durch Erstellen von Verzeichniseinträgen mit dem Befehl `In` ist es möglich, mehrere Namen für eine Datei mit demselben Dateiindex zu verbinden.

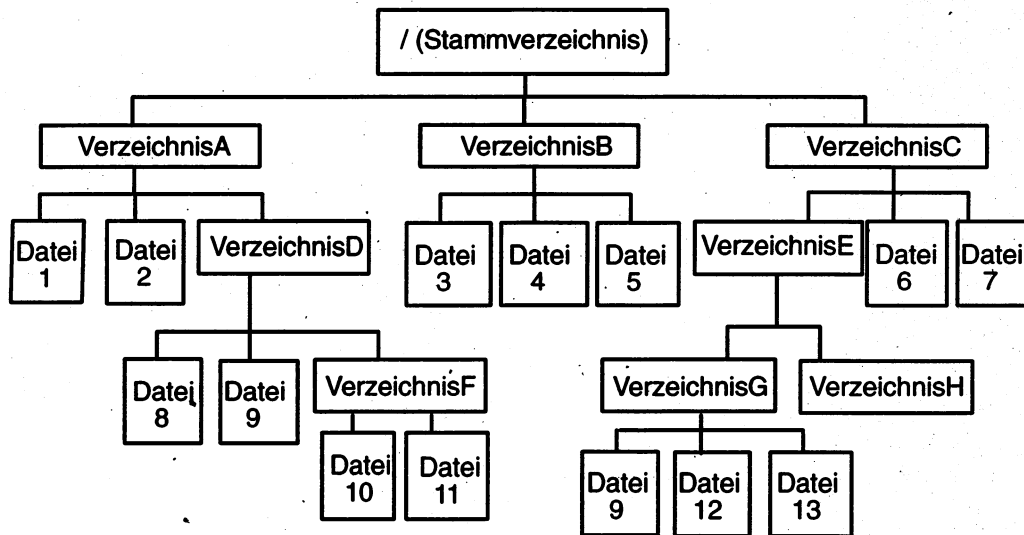
Da Verzeichnisse oft Informationen enthalten, die nicht allen Benutzern im System zugänglich sein sollten, ist es möglich, Verzeichnisse zu schützen. Durch Setzen der Berechtigung für ein Verzeichnis hat der Benutzer die Kontrolle über den Zugriff anderer Benutzer auf sein Verzeichnis. Der Benutzer kann auch festlegen, welche anderen Benutzer ggf. die Informationen in seinem Verzeichnis ändern können, oder ob dies überhaupt der Fall sein soll. Weitere Informationen sind dem Abschnitt "Einführung in den Zugriffsmodus auf Dateien und Verzeichnisse" auf Seite 3-13 zu entnehmen.

Einführung in den Aufbau von Verzeichnissen

Verzeichnisse enthalten Dateien oder Unterverzeichnisse oder eine Kombination aus beiden. Ein Unterverzeichnis ist ein Verzeichnis innerhalb eines Verzeichnisses. Das Verzeichnis, das das Unterverzeichnis enthält, ist das Väterverzeichnis.

Damit es vom Betriebssystem gefunden wird, enthält jedes Verzeichnis den Eintrag `..` (Punkt Punkt) für das Väterverzeichnis, in dem es erstellt wurde, und den Eintrag `.` (Punkt) für das Verzeichnis selbst. In den meisten Verzeichnislisten sind diese Dateien verdeckt.

Die Strukturen aus Väterverzeichnissen, Unterverzeichnissen und Dateien heißen Dateisysteme. Verzeichnisstrukturen werden oft mit einem umgekehrten Baum verglichen. Das Stammverzeichnis, symbolisiert durch einen `/` (Schrägstrich) und in der Verzeichnisbaumstruktur ganz oben dargestellt, ist die Basis. Unterverzeichnisse und Dateien verzweigen sich vom Stammverzeichnis aus nach unten. In der Abbildung "Beispiel für eine Verzeichnisstruktur" sind die Verzweigungen zwischen Stammverzeichnis, Verzeichnissen, Unterverzeichnissen und Dateien dargestellt.



Beispiel für eine Verzeichnisstruktur

Die Dateisystemstruktur von Verzeichnissen kann leicht komplex geraten. Daher sollte die Datei- und Verzeichnisstruktur so einfach wie möglich gehalten werden. Außerdem sollten Verzeichnis- und Dateinamen so gewählt werden, daß so eindeutig wie möglich auf ihren Inhalt hingewiesen wird, um das Arbeiten mit Dateien zu erleichtern.

Die Arbeit erfolgt immer innerhalb eines Verzeichnisses. Das Verzeichnis, in dem gerade gearbeitet wird, wird als aktuelles oder Arbeitsverzeichnis bezeichnet. Mit dem Befehl **pwd** wird der Name des Arbeitsverzeichnisses angezeigt. Mit dem Befehl **cd** kann das Arbeitsverzeichnis geändert werden. Beim Anmelden am System wird das Benutzer- oder Anmeldeverzeichnis aufgerufen. Dieses Verzeichnis wird vom Systemadministrator für jeden Benutzer definiert und angepaßt. Normalerweise sind Verzeichnisse, die der Benutzer für sich selbst erstellt, Unterverzeichnisse des Benutzerverzeichnisses. Zum Wechseln ins Benutzerverzeichnis den Befehl **cd** bei der Eingabeaufforderung eingeben.

Einführung in Pfadnamen

Jede Datei und jedes Verzeichnis kann durch einen eindeutigen Pfad, auch Pfadname genannt, in der Baumstruktur des Dateisystems gefunden werden. Der Pfadname gibt den Standort einer Datei oder eines Verzeichnisses im Dateisystem an. Ein Pfadname kann aus maximal 1023 Zeichen bestehen.

Im Dateisystem werden die folgenden zwei Arten von Pfadnamen verwendet:

Absolute Pfadnamen

Geben den Pfad vom Stammverzeichnis aus an. Absolute Pfadnamen beginnen immer mit einem / (Schrägstrich).

Relative Pfadnamen

Geben den Pfad zu einem Unterverzeichnis oder einer Datei vom aktuellen Verzeichnis aus an.

Ein absoluter Pfadname (auch vollständiger Pfadname genannt) steht für den vollständigen Namen eines Verzeichnisses oder einer Datei vom Stammverzeichnis aus. Unabhängig davon, an welcher Stelle im Dateisystem sich ein Benutzer befindet, kann ein bestimmtes Verzeichnis oder eine Datei durch Angabe des vollständigen Pfadnamens jederzeit aufgerufen werden. Absolute Pfadnamen beginnen immer mit einem / (Schrägstrich, dieses Symbol steht für das Stammverzeichnis). Im obigen Beispiel ist der Pfadname **/VerzeichnisA/VerzeichnisD/Datei9** der absolute Pfadname zur Datei **Datei9**. Der erste/

(Schrägstrich) steht für das Stammverzeichnis, der obersten Suchebene. Der verbleibende Teil des Pfadnamens zeigt an, daß für die Suche vom Stammverzeichnis in das Verzeichnis **VerzeichnisA**, dann in das Verzeichnis **VerzeichnisB** und anschließend in die Datei **Datei9** gewechselt wird.

In dem Beispiel, befinden sich zwei Dateien namens **Datei9**. Das ist nur möglich, weil durch den absoluten Pfadnamen jede Datei einen eindeutigen Namen im Dateisystem erhält. Die Pfadnamen **/VerzeichnisA/VerzeichnisD/Datei9** und **/VerzeichnisC/VerzeichnisE/VerzeichnisG/Datei9** bestimmen zwei Einzeldateien mit dem Namen **Datei9**.

Im Gegensatz zu vollständigen Pfadnamen geben relative Pfadnamen den Pfad zu einem Verzeichnis oder einer Datei vom aktuellen Arbeitsverzeichnis aus an. Aus diesem Grund beginnen sie auch nicht mit einem / (Schrägstrich). Mit einem relativen Pfadnamen kann der Name einer Datei im aktuellen Verzeichnis oder der Pfadname eines Verzeichnisses, das sich eine Ebene unterhalb des aktuellen Verzeichnisses befindet, angezeigt werden. Wenn im obigen Beispiel **VerzeichnisD** das aktuelle Verzeichnis ist, lautet der relative Pfadname zur Datei **Datei10VerzeichnisF/Datei10**, der absolute Pfadname jedoch immer **/VerzeichnisA/VerzeichnisD/VerzeichnisF/Datei10**.

Neben absoluten und relativen Pfadnamen kann auch die Schreibweise .. (Punkt Punkt) verwendet werden, um in die in der Dateisystemhierarchie unmittelbar übergeordnete Ebene zu gelangen. .. (Punkt Punkt) steht für das Vaterverzeichnis. Das aktuelle Verzeichnis kann auch durch . (Punkt) dargestellt werden. Diese Schreibweise wird gewöhnlich beim Ausführen von Programmen verwendet, die den aktuellen Verzeichnisnamen lesen.

Einführung in die Verzeichnistypen

Verzeichnisse können vom Betriebssystem, vom Systemadministrator oder vom Benutzer definiert werden. Vom System festgelegte Verzeichnisse enthalten bestimmte Systemdateien, wie z. B. Befehle. An oberster Stelle in der Dateisystemhierarchie steht das vom System festgelegte Stammverzeichnis. Das Stammverzeichnis enthält gewöhnlich folgende standardmäßige systembezogene Verzeichnisse:

/home	Enthält Anmeldeverzeichnisse für die Benutzer am System.
/usr	Enthält die Verzeichnisse lpp und include sowie Systemprogramme.
/usr/bin	Enthält Benutzerprogramme.
/dev	Enthält Gerätedateien für E/A-Einheiten.
/etc	Enthält Dateien für die Systeminitialisierung und die Systemverwaltung.
/tmp	Enthält Dateien, die nach einem angegebenen Zeitraum (in Tagen) gelöscht werden.

Zu den Verzeichnissen, die vom Systemadministrator definiert und angepaßt werden, gehören z. B. das Benutzerverzeichnis (**\$HOME**) und das Anmeldeverzeichnis des Benutzers. Bei der Anmeldung am Betriebssystem wird automatisch das Anmeldeverzeichnis des Benutzers aufgerufen. Dieses Verzeichnis wird ebenfalls aufgerufen, wenn beim Verzeichniswechsel mit dem Befehl **cd** kein Verzeichnisname angegeben wird.

Vom Benutzer erstellte Verzeichnisse werden als benutzerdefinierte Verzeichnisse bezeichnet. Mit diesen Verzeichnissen kann der Benutzer seine Dateien den jeweiligen Anforderungen entsprechend verwalten und pflegen.

Mit Verzeichnissen arbeiten

Es gibt viele verschiedene Möglichkeiten, mit Verzeichnissen und ihrem Inhalt zu arbeiten, z. B.:

- Verzeichnisse erstellen
- Den Inhalt von Verzeichnissen auflisten
- Den Inhalt eines Verzeichnisses in ein anderes Verzeichnis verschieben
- Verzeichnisse entfernen .

Neue Verzeichnisse werden mit dem Befehl **mkdir** erstellt. Ein neues Verzeichnis wird im aktuellen oder Arbeitsverzeichnis erstellt, wenn kein absoluter Pfad zu einem anderen Standort im Dateisystem angegeben wird. Der Name des aktuellen Verzeichnisses wird mit dem Befehl **pwd** angezeigt.

Jeder Benutzer kann sich den Inhalt der Verzeichnisse, auf die er Zugriff hat, mit dem Befehl **ls** oder **ls** anzeigen lassen. Dabei kann er durch die Kombination dieser Befehle mit Optionen auch zusätzliche Informationen zum jeweiligen Verzeichnis abfragen, z. B. Verzeichnistyp, Berechtigungen, Anzahl an Link-Verbindungen, Eigner, Gruppe, Größe und den Zeitpunkt der letzten Änderung. Der Befehl **diff** vergleicht die Dateinamen von zwei Verzeichnissen, listet die Dateinamen auf und vergleicht anschließend den Inhalt gemeinsam benutzter Dateien.

Mit dem Befehl **cd** kann von einem Verzeichnis in ein anderes gewechselt werden. Bei entsprechender Berechtigung kann auf jedes Verzeichnis im Dateisystem zugegriffen werden. Mit dem Befehl **cp** oder **copy** kann der Inhalt eines Verzeichnisses in ein anderes kopiert werden. Mit dem Befehl **mv** wird ein Verzeichnis verschoben (umbenannt).

Wird ein Verzeichnis nicht länger benötigt, kann es aus dem Dateisystem entfernt werden mit dem Befehl **rmdir** oder **rm**. Mit dem Befehl **rmdir** können nur Verzeichnisse gelöscht werden, die keine Dateien und Unterverzeichnisse mehr enthalten. Mit dem Befehl **rm -r** werden Verzeichnisse zusammen mit ihrem Inhalt gelöscht. Die Option **-r** steht für rekursives Löschen. Obwohl es mit diesem Befehl möglich ist, ein Verzeichnis zusammen mit seinem Inhalt zu entfernen, sollte das rekursive Löschen nur mit äußerster Vorsicht durchgeführt werden. Durch einen Fehler beim rekursiven Löschen könnten *alle* Dateien und Verzeichnisse im aktuellen Verzeichnis, auf die der Benutzer Zugriff hat, gelöscht werden.

Einführung in die Verbindungsfunktion

Link-Verbindungen sind Verbindungen zwischen einem Dateinamen und einer Dateiindexnummer (Dateiindex), der internen Darstellung einer Datei. Da Verzeichniseinträge Dateinamen mit Dateiindexen enthalten, ist jeder Verzeichniseintrag eine Link-Verbindung. Durch die Verwendung von Link-Verbindungen kann jede Datei bzw. jeder Dateiindex viele verschiedene Namen besitzen.

Link-Verbindungen werden mit dem Befehl **ln** erstellt. Es gibt folgende zwei Arten von Link-Verbindungen:

feste Link-Verbindung

Ermöglicht den Zugriff auf Daten einer Datei von einer neuen Datei aus. Feste Link-Verbindungen stellen sicher, daß eine Datei entsteht. Wenn die letzte feste Link-Verbindung gelöscht wird, wird der Dateiindex freigegeben, und es kann nicht mehr auf die Daten in der Datei zugegriffen werden.

Feste Link-Verbindungen können nur zwischen Dateien in demselben Dateisystem erstellt werden.

symbolische Link-Verbindung

Ermöglicht den Zugriff auf Daten in einem anderen Dateisystem von einer neuen Datei aus. Symbolische Link-Verbindungen sind reguläre Dateien, die einen Pfadnamen enthalten. Wenn ein Prozeß eine symbolische Link-Verbindung aufnimmt, wird der Pfadname an den vom Prozeß gesuchten Pfad angehängt und die Suche fortgesetzt. Symbolische Link-Verbindungen schützen eine Datei nicht davor, aus dem Dateisystem gelöscht zu werden.

Der Benutzer, der eine Datei erstellt hat, bleibt Eigner der Datei, ungeachtet der Zahl der erstellten Link-Verbindungen. Der Zugriffsmodus kann nur vom Eigner der Datei oder vom Root festgelegt werden. Von einer verbundenen Datei aus können jedoch mit dem geeigneten Zugriffsmodus Änderungen in einer Datei gemacht werden.

Dateiindex 798 enthält z. B. eine Kurzinformation zu den Verkäufen im Monat Juni in der Filiale Omaha. Gegenwärtig lautet der Verzeichniseintrag für diese Kurzinformation:

Dateiindexnummer Dateiname

798 **.Kurzinfo**

Da sich die Information auf Daten bezieht, die sich in den Verzeichnissen **Verkäufe** und **Omaha** befinden, werden die Informationen mit Hilfe einer Verbindungsfunktion gemeinsam benutzt. Mit dem Befehl **ln** werden Link-Verbindungen zu diesen Verzeichnissen erstellt. Nun besitzt die Datei drei Dateinamen:

Dateiindexnummer Dateiname

798 **Kurzinfo**

798 **Verkäufe/Juni**

798 **Omaha/Juniverkäufe**

Wenn mit dem Befehl **pg** oder **cat** der Inhalt einer der drei Dateien angesehen wird, wird jedesmal dieselbe Information angezeigt. Wird der Inhalt des Dateiindexes von einer der drei Dateien editiert, wird der Inhalt der neuen Ausgabe aus den von allen drei Dateien angezeigten Daten bestehen.

Eine Datei bzw. ein Verzeichnis besteht so lange, wie eine feste Link-Verbindung zum Dateiindex der Datei besteht. Mit dem Befehl **ls -l** wird die lange Liste mit der Zahl der Link-Verbindungen zu allen Dateien und Unterverzeichnissen angezeigt. Sämtliche festen Link-Verbindungen werden unabhängig vom Zeitpunkt der Erstellung vom Betriebssystem gleich behandelt. Mit dem Befehl **rm** wird die Link-Verbindung zu einer angegebenen Datei gelöscht. Wenn die letzte Link-Verbindung zu einem Dateiindex gelöscht ist, gehen die Daten verloren, da nicht mehr auf sie zugegriffen werden kann. Die Dateiindexnummer wird später einer neu erstellten Datei zugeordnet.

Einführung in den Zugriff auf Dateien und Verzeichnisse

Jede Datei hat einen Eigner. Bei neu erstellten Dateien ist dies der Benutzer, der sie erstellt hat. Der Eigner ordnet der Datei einen Zugriffsmodus zu. Ein Zugriffsmodus gibt anderen Systembenutzern die Berechtigung, eine Datei zu lesen, zu ändern oder auszuführen. Der Zugriffsmodus für eine Datei kann nur vom Eigner der Datei oder von Benutzern mit Root-Berechtigung für das Stammverzeichnis geändert werden.

Es gibt drei Kategorien von Benutzern (Benutzer/ Eigner, Gruppe von Benutzern, alle anderen Benutzer). Die Zugriffsberechtigung für diese Gruppen kann aus einer Kombination der drei Modi Lesen, Schreiben und Ausführen bestehen. Beim Erstellen einer neuen Datei erhält der Benutzer, der die Datei erstellt, standardmäßig Berechtigung zum Lesen, Schreiben und Ausführen. Die anderen beiden Gruppen haben Lese- und Ausführungsberechtigung. Die folgende Tabelle zeigt die standardmäßigen Dateizugriffsmodi für die drei Benutzergruppen:

	Lesen	Schreiben	Ausführen
Eigner	ja	ja	ja
Gruppe	ja	nein	ja
Andere	ja	nein	ja

Eignerstatus und Benutzergruppen

Zunächst wird der Eigner einer Datei durch die Benutzer-ID der Person gekennzeichnet, die die Datei erstellt hat. Der Eigner einer Datei legt die Berechtigung fest, d. h. wer zum Lesen, Schreiben execute (Ändern) oder Ausführen einer Datei berechtigt ist. Der Eignerstatus kann mit dem Befehl **chown** geändert werden.

Jede Benutzer-ID wird einer Gruppe mit eindeutiger Gruppen-ID zugeordnet. Der Systemmanager legt die Benutzergruppen beim Installieren des Systems fest. Wenn eine neue Datei erstellt wird, erteilt das Betriebssystem eine Berechtigung für die Benutzer-ID der Person, die die Datei erstellt hat, für die Gruppen-ID der Gruppe, der der Eigner angehört und für eine Gruppe namens Andere, der alle übrigen Benutzer angehören. Mit dem Befehl **id** werden die Benutzer-ID, die Gruppen-ID und die Namen aller Gruppen, denen der Benutzer angehört, angezeigt.

In Dateiberichten (z. B. Berichte, die mit dem Befehl **ll** oder **ls** angezeigt werden) werden die drei Benutzergruppen stets in folgender Reihenfolge aufgeführt: Benutzer, Gruppe und Andere. Mit dem Befehl **groups** können die Namen aller zu einer Benutzer-ID gehörenden Gruppen angezeigt werden.

Dateizugriffsmodi

Dateien können gelesen (r), beschrieben (w) oder ausgeführt (x) werden. Das System legt fest, welche Benutzer Berechtigung haben, und auf welcher Berechtigungsstufe sie für jeden dieser Vorgänge stehen. Zugriffsmodi werden im Betriebssystem auf zwei Arten dargestellt: symbolisch und numerisch.

Symbolische Darstellung der Zugriffsmodi

Zugriffsmodi werden folgendermaßen symbolisch dargestellt:

r Gibt Lesezugriff an, d. h. der Benutzer kann sich den Inhalt einer Datei anzeigen lassen.

- w** Gibt Schreibzugriff an, d. h. der Benutzer kann den Inhalt einer Datei ändern.
- x** Gibt Ausführungsberechtigung an. Ausführungsberechtigung für ausführbare Dateien (gewöhnliche Dateien, die Programme enthalten) bedeutet, daß das Programm ausgeführt werden kann. Ausführungsberechtigung für Verzeichnisse bedeutet, daß der Inhalt eines Verzeichnisses gesucht werden kann.

Bei einer Datei mit dem Zugriffsmodus `rwxr-xr-x` (symbolische Darstellung der Standardeinstellung) besteht Lese- und Ausführungsberechtigung für alle drei Gruppen, Schreibzugriff jedoch nur für den Eigner der Datei.

Über den Befehl `ls` in Verbindung mit der Option `-l` erhält der Benutzer eine ausführliche Auflistung des Inhalts des aktuellen Verzeichnisses. Die ersten zehn Zeichen in der Auflistung zeigen den Dateityp und die Berechtigungen für jede der drei Gruppen an. Mit dem Befehl `ls -l` werden auch der Eigner und die Gruppe zu jeder Datei und zu jedem Verzeichnis angezeigt.

Das erste Zeichen gibt den Dateityp an. Die übrigen neun Zeichen enthalten Informationen zu den Berechtigungen der drei Benutzerklassen. Folgende Zeichen werden für die Angabe des Dateityps verwendet:

- reguläre Dateien
- d** Verzeichnis
- b** blockorientierte Gerätedateien
- c** zeichenorientierte Gerätedateien
- p** Pipe-orientierte Gerätedateien
- l** symbolische Verbindungen
- s** Sockets.

Das folgende Beispiel zeigt eine Auflistung mit dem Befehl `ls -l`:

```
-rwxrwxr-x      2      janet  acct   512 Mär 01 13:33 meinDat
```

Das erste - (Minuszeichen) zeigt eine reguläre Datei an. Die Zeichen 2 bis 4, `rwx`, stehen für die Berechtigung des Benutzers (Lesen, Schreiben und Ausführen). Die Zeichen 5 bis 7, `rwx`, stehen für die Berechtigungen der Gruppe (Lesen, Schreiben und Ausführen). Die Zeichen 8 bis 10, `r-x`, stehen für die Lese- und Ausführungsberechtigung aller übrigen Benutzer. Ein Minuszeichen bei einem dieser neun Zeichen gibt an, daß keine Berechtigung besteht.

Die Zahl 2 gibt an, wieviele Link-Verbindungen zur Datei bestehen. `janet` ist der Name des Dateieigners. `acct` ist der Name der Gruppe des Dateieigners Janet. 512 ist die Größe der Datei in Byte, Mär 01 13:33 ist das Datum und die Uhrzeit der letzten Änderung und `meinDat` ist der Dateiname.

Numerische Darstellung der Zugriffsmodi

Die numerische Darstellung ist 4 für Lesezugriff, 2 für Schreibzugriff und 1 für Ausführungsberechtigung. Die Gesamtsumme zwischen 1 und 7 gibt die Berechtigung für jede Gruppe an (Benutzer, Gruppe und Andere).

Mit der folgenden Tabelle können die Werte für jede Zugriffsebene ermittelt werden:

Gesamtsumme	Lesen	Schreiben	Ausführen
0	-	-	-
1	-	-	1
2	-	2	-
3	-	2	1
4	4	-	-
5	4	-	1
6	4	2	-
7	4	2	1

Wenn eine Datei erstellt wird, lautet die Berechtigung standardmäßig 755, d. h. der Benutzer hat Lese-, Schreib- und Ausführungsberechtigung (4+2+1=7), die Gruppe hat Lese- und Ausführungsberechtigung (4+1=5) und alle anderen Benutzer haben Lese- und Ausführungsberechtigung (4+1=5).

Berechtigungen können für eigene Dateien mit dem Befehl **chmod** geändert werden.

Empfohlene Dokumentation

Zugehörige Informationen

Im Abschnitt "Überblick über Dateisysteme" auf Seite 2-2 werden die Dateitypen und Verarbeitungsvorgänge zu Dateisystemen beschrieben.

Das Kapitel "Files, Directories, and File Systems Programmers Overview" im Handbuch *General Programming Concepts* bietet eine Einführung in den Dateiindex, die Zuordnung des Dateibereichs und die Unterroutinen, die Dateien und Dateisysteme steuern.

Im Abschnitt "Understanding JFS i-nodes" im Handbuch *General Programming Concepts* werden Dateiindizes mit Hilfe von grafischen Darstellungen näher erläutert.

Im Abschnitt "Überblick über Shells" auf Seite 2-21 wird beschrieben, was Shells sind, welche verschiedenen Arten von Shells es gibt und wie Shells die verschiedenen Möglichkeiten der Befehlsinterpretation beeinflussen.

Im Abschnitt "Überblick über die Möglichkeiten der Sicherung" auf Seite 3-40 wird erläutert, wie wichtig es ist, Sicherungskopien von Dateien auf Band oder Diskette zu erstellen.

Der Abschnitt "Überblick über die Ein- und Ausgabe" auf Seite 2-28 bietet eine Einführung in allgemeine Verarbeitungsvorgänge und die Grundlagen der Umadressierung.

Im Abschnitt "Überblick über Befehle" auf Seite 2-41 werden Befehle, Befehlssyntax, Befehlsanweisungen und Hintergrundprozesse erklärt.

Dateiverwaltung

Für jeden der folgenden Dateiverwaltungsvorgänge werden ein Befehl und ein Beispiel angeführt:

- Dateien löschen
- Dateien verschieben
- Dateien kopieren
- Dateien drucken
- Inhalt von Dateien anzeigen
- Dateien umbenennen
- Programmausgabe anzeigen und in eine Datei kopieren
- Dateien verbinden
- Dateien suchen
- Zeichenfolgen suchen
- Wörter, Zeilen und Zeichen zählen
- Teil einer Datei anzeigen
- Dateityp anzeigen
- Im Dateisystem verfügbaren Speicherbereich anzeigen

Dateien löschen

Dateien können mit den Befehlen **del** und **rm** gelöscht werden.

Verwenden des Befehls **del**

Achtung: Mit dem Befehl **del** kann der Eigner auch schreibgeschützte Datei löschen, da dieser Befehl den Dateizugriffsschutz ignoriert. Der Benutzer muß jedoch Schreibzugriff auf das Verzeichnis haben, in dem sich die Datei befindet. Da durch das Drücken der Eingabetaste ebenfalls mit "ja" bestätigt werden kann, sollte der Benutzer sehr vorsichtig sein, damit er nicht unbeabsichtigt Dateien löscht.

Der Befehl **del** zeigt die zu löschenden Dateien an und verlangt eine Bestätigung, bevor er die Dateien löscht. Zum Antworten mit "ja" (Dateien löschen) die Eingabetaste drücken oder *y* (bzw. das Äquivalent für *y* in der jeweiligen Landessprache) eingeben. Jede andere Antwort wird als "nein" gewertet (Dateien nicht löschen).

Mit dem Befehl **del** können keine Verzeichnisse gelöscht werden. Der Abschnitt zum Befehl **rmdir** enthält Informationen zum Löschen von Verzeichnissen.

Beispiel

Zum Löschen einer Datei eingeben:

```
del chap1.bak
```

Daraufhin wird eine Nachricht mit folgendem Wortlaut angezeigt:

```
del : chap1.bak
löschen? Ja eingeben oder die Eingabetaste drücken.
Beliebige Taste für nein drücken.
```

Zum Antworten kann entweder die Eingabetaste gedrückt oder *y* bzw. das Äquivalent der Landessprache eingegeben werden. Mit jeder anderen Taste wird das Löschen abgebrochen. Siehe Sicherheitshinweis **Achtung**.

Verwenden des Befehls rm

Der Befehl **rm** löscht die Einträge für die angegebene(n) Datei(en) aus einem Verzeichnis. Dabei benötigt der Benutzer keine Lese- oder Schreibberechtigung für die zu löschenden Dateien. Er muß jedoch Schreibzugriff auf das Verzeichnis haben, das die Datei(en) enthält.

Beispiele

1. Zum Löschen einer Datei eingeben:

```
rm myfile
```

2. Zum einzelnen Löschen von Dateien eingeben:

```
rm -i mydir/*
```

Nachdem ein Dateiname angezeigt wird, jeweils **y** eingeben, um die Datei zu löschen oder die Eingabetaste drücken, um sie zu behalten.

Dateien verschieben

Achtung: Der Befehl **mv** kann zahlreiche Dateien überschreiben, wenn er ohne die Option **-i** eingegeben wird. Bei Verwendung der Option **-i** muß der Benutzer erst bestätigen, bevor eine Datei überschrieben wird. Wenn die Optionen **-f** und **-i** zusammen angegeben werden, hat die als letzte angegebene Option Vorrang.

Der Befehl **mv** (**move**) verschiebt Dateien und Verzeichnisse von einem Verzeichnis in ein anderes oder benennt eine Datei oder ein Verzeichnis um. Wird eine Datei oder ein Verzeichnis in ein neues Verzeichnis verschoben, bleibt der Basisdateiname unverändert. (Eine Datei kann nur an eine andere Position verschoben werden.)

Beispiele

1. Umbenennen einer Datei und Verschieben in ein anderes Verzeichnis:

```
mv intro manual/chap1
```

Damit wird `intro` in `manual/chap1` verschoben. Der Name `intro` wird aus dem aktuellen Verzeichnis gelöscht, und dieselbe Datei erscheint als `chap1` im Verzeichnis `manual`. Siehe Sicherheitshinweis **Achtung**.

2. Verschieben einer Datei in ein anderes Verzeichnis unter Beibehaltung des Dateinamens:

```
mv chap3 manual
```

Damit wird `chap3` in `manual/chap3` verschoben. Siehe Sicherheitshinweis **Achtung**.

Dateien kopieren

Der Befehl **cp** kopiert den Inhalt einer in der *Quelldatei* oder im *Quellenverzeichnis* angegebenen Datei bzw. eines Verzeichnisses in eine in der *Zieldatei* oder im *Zielverzeichnis* angegebene Datei bzw. ein Verzeichnis. Besteht die als *Zieldatei* angegebene Datei bereits, werden die darin enthaltenen Daten überschrieben. Wenn mehrere *Quelldateien* kopiert werden, muß das Ziel ein Verzeichnis sein.

Um eine Kopie der *Quelldatei* in ein Verzeichnis zu stellen, einen Pfad zu einem bestehenden Verzeichnis im *Zielverzeichnis* anlegen. Dateien behalten beim Kopieren in ein Verzeichnis ihren Namen bei, wenn am Ende des Pfads kein neuer Dateiname angegeben wird. Der Befehl **cp** kopiert ganze Verzeichnisse in andere Verzeichnisse, wenn die Option **-r** oder **-R** angegeben wird.

Es ist auch möglich, Gerätedateien zu kopieren. Die dafür am besten geeignete Option ist **-R**. Durch die Angabe von **-R** werden Gerätedateien unter einem anderen Pfadnamen neu erstellt. Durch die Angabe der Option **-r** setzt der Befehl **cp** Gerätedateien in reguläre Dateien um.

Beispiele

1. Erstellen einer Kopie von einer Datei im aktuellen Verzeichnis:

```
cp prog.c prog.bak
```

Damit wird `prog.c` in `prog.bak` kopiert. Existiert die Datei `prog.bak` noch nicht, wird sie durch **cp** erstellt. Existiert sie bereits, ersetzt sie der Befehl **cp** durch eine Kopie der Datei `prog.c`.

2. Kopieren einer Datei vom aktuellen Verzeichnis in ein anderes Verzeichnis:

```
cp müller /home/liesl/kunden
```

Damit wird die Datei `müller` nach `/home/liesl/kunden/müller` kopiert.

3. Kopieren aller Dateien eines Verzeichnisses in ein neues Verzeichnis:

```
cp /home/liesl/kunden/* /home/sepp1/klienten
```

Damit werden die Dateien im Verzeichnis `Kunden` in das Verzeichnis `Klienten` kopiert.

4. Kopieren bestimmter Dateien in ein anderes Verzeichnis:

```
cp müller meier schmidt /home/liesl/kunden
```

Damit werden die Dateien `müller`, `meier` und `schmidt` vom aktuellen Arbeitsverzeichnis in das Verzeichnis `/home/liesl/kunden` kopiert.

5. Kopieren von Dateien mit Angabe von Suchmusterzeichen:

```
cp programme/*.c
```

Damit werden die Dateien im Verzeichnis `programme`, die mit `.c` enden, in das aktuelle Verzeichnis (`.`) kopiert. Zwischen `c` und dem letzten Punkt ist ein Leerzeichen einzugeben.

Dateien drucken

Der Befehl **lpr** verwendet zum Drucken der angegebenen *Datei* einen SPOOL-Dämon, sobald die notwendigen Einrichtungen verfügbar sind. Wenn keine Dateien angegeben werden, liest der Befehl **lpr** von der Standardeingabeeinheit.

Beispiele

1. Drucken von drei Exemplaren der Dateien `new.index.c`, `print.index.c` und `more.c`:

```
lpr -#3 new.index.c print.index.c more.c
```

Damit werden für die Dateien `new.index.c`, `print.index.c` und `more.c` jeweils drei Exemplare gedruckt.

2. Drucken von drei Exemplaren der verknüpften Dateien `new.index.c`, `print.index.c` und `more.c`:

```
cat new.index.c print.index.c more.c | lpr -#3
```

Inhalt von Dateien anzeigen

Der Inhalt von Dateien kann mit Hilfe der Befehle **pg**, **cat** und **more** angezeigt werden.

Verwenden des Befehls **pg**

Der Befehl **pg** liest die im Parameter *Datei* angegebenen Dateinamen und schreibt sie seitenweise in die Standardausgabe. Wenn für den Parameter *Datei* ein - (Minuszeichen) angegeben oder der Befehl **pg** ohne Optionen ausgeführt wird, liest der Befehl **pg** von der Standardeingabeeinheit. Am Ende jeder angezeigten Seite steht eine Eingabeaufforderung. Nach Drücken der Eingabetaste wird eine weitere Seite angezeigt. Wenn der Befehl **pg** zusammen mit Unterbefehlen verwendet wird, können Seiten überprüft werden, die bereits angezeigt wurden.

Beispiel

Seitenweises Anzeigen des Inhalts einer Datei:

```
pg Dateiname
```

Verwenden der Befehle **more** und **page**

Der Befehl **more** zeigt fortlaufenden Text seitenweise an. Am Ende jeder Seite wird das Wort *Weitere* angezeigt. Nach Drücken der Eingabetaste wird eine weitere Zeile angezeigt. Nach Drücken der Leertaste wird eine weitere Seite mit Text angezeigt.

Hinweis: Bei einigen Datenstationsmodellen wird bei Verwendung des Befehls **more** zuerst die Anzeige gelöscht, bevor die nächste Seite angezeigt wird.

Beispiele

1. Anzeigen einer Datei mit dem Namen *meinDat*:

```
more meinDat
```

Zum Anzeigen der nächsten Seite die Leertaste drücken.

2. Anzeigen mit dem Befehl **ps**:

```
ps -ef | more
```

Verwenden des Befehls **cat**

Der Befehl **cat** liest die im Parameter *Datei* angegebenen Dateien der Reihe nach und schreibt sie in die Standardausgabe.

Beispiel

Anzeigen einer Datei an der Datenstation:

```
cat notes
```

Damit werden die Daten in der Datei *notes* angezeigt. Wenn die Datei mehr als 23 Zeilen enthält, werden nur die letzten 23 Zeilen angezeigt. Zum seitenweisen Anzeigen einer Datei den Befehl **pg** verwenden.

Dateien umbenennen

Achtung: Der Befehl **mv** kann zahlreiche Dateien überschreiben, wenn er ohne die Option **-i** eingegeben wird. Bei Verwendung der Option **-i** muß der Benutzer erst bestätigen, bevor eine Datei überschrieben wird. Wenn die Optionen **-f** und **-i** zusammen angegeben werden, hat die als letzte angegebene Option Vorrang.

Der Befehl **mv** (**move**) verschiebt Dateien und Verzeichnisse von einem Verzeichnis in ein anderes oder benennt sie um. Wird eine Datei oder ein Verzeichnis in ein neues Verzeichnis verschoben, bleibt der Basisdateiname unverändert. (Eine Datei kann nur an eine andere Position verschoben werden.)

Beispiel

Umbenennen einer Datei:

```
mv appendix apndx.a
```

Damit wird `appendix` in `apndx.a` umbenannt. Wenn die Datei `apndx.a` bereits besteht, wird ihr Inhalt mit dem der Datei `appendix` überschrieben. Siehe Sicherheitshinweis **Achtung**.

Programmausgabe anzeigen und in eine Datei kopieren

Der Befehl **tee** liest von der Standardeingabeeinheit und schreibt anschließend die Ausgabe eines Programms in die Standardausgabeeinheit, wobei er die Ausgabe gleichzeitig in die angegebene(n) Datei(en) kopiert.

Beispiele

1. Gleichzeitiges Anzeigen und Sichern der Ausgabe eines Befehls:

```
ps -ef | tee program.ps
```

Damit wird die Standardausgabe des Befehls `ps -ef` an der Datenstation angezeigt und gleichzeitig in der Datei `program.ps` gesichert. Wenn die Datei `program.ps` bereits besteht, wird sie ersetzt.

2. Anzeigen und Sichern der Ausgabe eines Befehls in eine bestehende Datei:

```
ls -l | tee -a program.ls
```

Damit wird die Standardausgabe des Befehls `ls -l` an der Datenstation angezeigt und gleichzeitig an das Ende der Datei `program.ls` angehängt. Wenn die Datei `program.ls` noch nicht besteht, wird sie erstellt.

Dateien verbinden

Der Befehl **ln** verbindet die im Parameter *Quellendatei* angegebene Datei mit der im Parameter *Zieldatei* angegebenen Datei oder mit einer Datei selben Namens in einem anderen im Parameter *Zielverzeichnis* angegebenen Verzeichnis. In der Standardeinstellung erstellt der Befehl **ln** harte Verbindungen. Um mit dem Befehl **ln** symbolische Verbindungen zu erstellen, die Option `-s` angeben.

Wenn eine Datei mit einer Datei anderen Namens verbunden wird, kann nur eine Datei angegeben werden. Wenn sie mit einem Verzeichnis verbunden wird, können mehrere Dateien angegeben werden.

Der Parameter *Zieldatei* kann wahlweise verwendet werden. Wenn keine Zieldatei angegeben wird, erstellt der Befehl **ln** im aktuellen Verzeichnis eine neue Datei. Die neue Datei übernimmt den Namen der im Parameter *Quellendatei* angegebenen Datei (siehe Beispiel 5).

Hinweis: Dateien in verschiedenen Dateisystemen können ohne die Option `-s` nicht verbunden werden.

Beispiel

1. Erstellen einer weiteren (Alias-) Verbindung zu einer Datei:

```
ln -f chap1 intro
```

Damit wird `chap1` mit dem neuen Namen `intro` verbunden. Wenn `intro` noch nicht besteht, wird der Dateiname erstellt. Wenn `intro` bereits besteht, wird die Datei durch eine Verbindung zu `chap1` ersetzt. Dadurch beziehen sich die Dateinamen `chap1` und `intro` auf dieselbe Datei. Jede Änderung in einer der Dateien erscheint auch in der anderen. Wenn eine Datei mit dem Befehl `del` oder `rm` gelöscht wird, bleibt sie unter dem anderen Namen bestehen.

2. Verbinden einer Datei mit einer Datei gleichen Namens in einem anderen Verzeichnis:

```
ln index manual
```

Damit wird `index` mit dem neuen Namen `manual/index` verbunden.

Hinweis: In Beispiel 1 ist `intro` der Name einer Datei; in Beispiel 2 ist `manual` ein Verzeichnis, das bereits besteht.

3. Verbinden mehrerer Dateien mit Namen in einem anderen Verzeichnis:

```
ln chap2 jim/chap3 /home/manual
```

Damit wird `chap2` mit dem neuen Namen `/home/manual/chap2` und `jim/chap3` mit `/home/manual/chap3` verbunden.

4. Den Befehl `ln` zusammen mit Suchmusterzeichen verwenden:

```
ln manual/* .
```

Damit werden alle Dateien im Verzeichnis `manual` mit dem aktuellen Verzeichnis verbunden und erhalten durch den Punkt (`.`) dieselben Namen, die sie im Verzeichnis `manual` haben.

Hinweis: Zwischen dem Stern und dem Punkt muß eine Leerstelle eingegeben werden.

5. Erstellen einer symbolischen Verbindung:

```
ln -s /tmp/toc toc
```

Damit wird im aktuellen Verzeichnis die symbolische Verbindung `toc` erstellt. Die Datei `toc` zeigt auf die Datei `/tmp/toc`. Wenn die Datei `/tmp/toc` bereits besteht, listet der Befehl `cat toc` ihren Inhalt auf.

Erstellen einer symbolischen Verbindung ohne Angabe des Parameters *Zieldatei*:

```
ln -s /tmp/toc
```

Dateien suchen

Der Befehl `find` sucht das Verzeichnis nach jedem angegebenen *Pfad* ab und sucht nach Dateien, die mit einem Ausdruck übereinstimmen, der entsprechend den im folgenden Text beschriebenen Termini eingegeben wurde. Die Ausgabe des Befehls `find` ist abhängig von den im Parameter *Ausdruck* angegebenen Termini.

Die BSD-Schnellsuchsyntax Version 4.3 wird vom Befehl `find` nicht unterstützt.

Beispiele

1. Auflisten aller Dateien im Dateisystem, die einen Basisdateinamen haben:

```
find / -name .profile -print
```

Damit wird das gesamte Dateisystem abgesucht und die vollständigen Pfadnamen aller Dateien mit dem Namen `.profile` werden aufgelistet. Der Schrägstrich (`/`) gibt dem

Befehl **find** an, daß das Stammverzeichnis samt aller Unterverzeichnisse abgesucht werden soll. Um Zeit zu sparen empfiehlt es sich, die Suche auf die Verzeichnisse zu beschränken, in denen die Dateien vermutet werden.

2. Auflisten der Dateien im aktuellen Verzeichnis, die einen bestimmten Berechtigungscode haben:

```
find . -perm 0600 -print
```

Damit werden die Namen der Dateien aufgelistet, die *nur* Benutzer-Lese- und Benutzer-Schreib-Berechtigung haben. Der Punkt (.) gibt dem Befehl **find** an, daß das aktuelle Verzeichnis samt aller Unterverzeichnisse abgesucht werden soll. Der Abschnitt zum Befehl **chmod** enthält eine Erklärung der Berechtigungscode.

3. Suchen nach Dateien mit bestimmten Berechtigungscode in mehreren Verzeichnissen:

```
find manual clients proposals -perm -0600 -print
```

Damit werden die Namen der Dateien aufgelistet, die Benutzer-Lese- und Benutzer-Schreib- und andere Berechtigungen haben und sich in den Verzeichnissen `manual`, `clients` und `proposals` sowie deren Unterverzeichnissen befinden. Im ersten Beispiel werden durch den Parameter `-perm 0600` nur die Dateien ausgewählt, deren Berechtigungscode exakt mit `0600` übereinstimmt. In diesem Beispiel werden durch den Parameter `-perm -0600` die Dateien ausgewählt, deren Berechtigungscode Zugriffe zuläßt, die mit `0600` und darüber angegeben werden. Dadurch stimmen auch die Berechtigungscode `0622` und `2744` mit dem Parameter überein.

4. Auflisten aller Dateien im aktuellen Verzeichnis, die in den letzten 24 Stunden geändert wurden:

```
find . -ctime 0 -print
```

5. Suchen nach regulären Dateien mit mehreren Verbindungen:

```
find . -type f -links +1 -print
```

Damit werden die Namen der gewöhnlichen Dateien (`-type f`) aufgelistet, die mehr als eine Verbindung (`-links +1`) haben.

Hinweis: Jedes Verzeichnis hat mindestens zwei Verbindungen: den Eintrag in seinem Elternverzeichnis und seinen eigenen Eintrag `.` (Punkt). Im Abschnitt zum Befehl **ln** werden Mehrdateienverbindungen erklärt.

6. Drucken der Pfadnamen aller Dateien in oder unterhalb des aktuellen Verzeichnisses mit Ausnahme der Verzeichnisse `SCCS` oder der Dateien in den Verzeichnissen `SCCS`:

```
find . -name SCCS -prune -o -print
```

Drucken der Pfadnamen aller Dateien in oder unterhalb des aktuellen Verzeichnisses einschließlich der Namen der Verzeichnisse `SCCS`:

```
find . -print -name SCCS -prune
```

7. Suchen nach allen Dateien, die exakt 414 Byte groß sind:

```
find . -size 414c -print
```

Zeichenfolgen suchen

Der Befehl **grep** sucht nach dem im Parameter *Muster* angegebenen Muster und schreibt jede übereinstimmende Zeile in die Standardausgabereinheit. Die Muster sind begrenzte reguläre Ausdrücke, wie sie bei den Befehlen **ed** und **egrep** verwendet werden. Der Befehl **grep** verwendet einen kompakten nicht-deterministischen Algorithmus.

Wenn im Parameter *Datei* mehrere Namen angegeben werden, zeigt der Befehl **grep** den Namen der Datei an, die die übereinstimmende Zeile enthält. Zeichen, die eine besondere Bedeutung für die Shell haben (\$, *, [, |, ^, (,), \, \), müssen im Parameter *Muster* zwischen Anführungszeichen stehen. Wenn es sich beim Parameter *Muster* nicht um eine einfache Zeichenfolge handelt, muß in der Regel das gesamte Muster zwischen einfache Anführungszeichen gesetzt werden. Bei einem Ausdruck [a-z], bedeutet das Minuszeichen "bis" entsprechend der aktuellen Sortierfolge. Eine Sortierfolge kann Äquivalenzklassen für die Verwendung in Zeichenbereichen definieren.

Beispiele

1. Verwenden eines Musters, das einige der Suchmusterzeichen *, ^, ?, [,], \{, \}, \{ und \} enthält:

```
grep "^[a-zA-Z]" pgm.s
```

Damit werden alle Zeilen in `pgm.s` angezeigt, die mit einem Buchstaben beginnen.

2. Anzeigen aller Zeilen, die nicht mit einem Muster übereinstimmen:

```
grep -v bubble sort.c
```

Damit werden alle Zeilen in der Datei `sort.c` angezeigt, die nicht das Wort "bubble" enthalten.

3. Anzeigen der Zeilen, die mit der Zeichenfolge "staff" übereinstimmen, in der Ausgabe des Befehls `ls`:

```
ls -l | grep staff
```

Wörter, Zeilen und Zeichen zählen

Der Befehl **wc** zählt standardmäßig die Anzahl der Zeilen, Wörter und Byte der im Parameter *Datei* angegebenen Datei. Wenn im Parameter *Datei* keine Datei angegeben wird, wird die Standardeingabe gezählt. Der Befehl schreibt die Ergebnisse in die Standardausgabereinheit und zeichnet eine Gesamtsumme für alle angegebenen Dateien auf. Wenn Optionen angegeben werden, bestimmt die Reihenfolge der Optionen die Reihenfolge der Ausgabe. Ein Wort ist definiert als eine Zeichenfolge, die von Leerzeichen, Tabulatorzeichen oder Zeichen in der nächsten Zeile begrenzt wird.

Wenn in der Befehlszeile Dateien angegeben werden, werden deren Namen zusammen mit den Ergebnissen ausgegeben.

Beispiele

1. Anzeigen der Anzahl der Zeilen, Wörter und Byte einer Datei:

```
wc chap1
```

Damit wird die Anzahl der Zeilen, Wörter und Byte in der Datei `chap1` angezeigt.

2. Anzeigen der Anzahl der Wörter und Byte:

```
wc -cw chap*
```

Damit wird die Anzahl der Byte und Wörter jeder Datei angezeigt, deren Name mit `chap` beginnt; die Gesamtsummen werden ebenfalls angezeigt.

Teile von Dateien anzeigen

Teile von Dateien können mit Hilfe der Befehle **head** und **tail** angezeigt werden.

Verwenden des Befehls **head**

Der Befehl **head** schreibt die ersten Zeilen oder Byte jeder angegebenen Datei bzw. der Standardeingabeeinheit in die Standardausgabeeinheit. Wenn zusammen mit dem Befehl **head** keine Option angegeben wird, werden standardmäßig die ersten 10 Zeilen angezeigt.

Beispiel

Anzeigen der ersten fünf Zeilen der Datei `Test`:

```
head -5 Test
```

ODER

```
head -n 5 Test
```

Verwenden des Befehls **tail**

Der Befehl **tail** schreibt die im Parameter *Datei* angegebene Datei von einer angegebenen Stelle aus in die Standardausgabeeinheit.

Beispiele

1. Anzeigen der letzten 10 Zeilen der Datei `notes`:

```
tail notes
```

2. Angeben der Anzahl der Zeilen, die vom Ende der Datei `notes` gelesen werden sollen:

```
tail -20 notes
```

3. Seitenweises Anzeigen der Datei `notes`, beginnend beim zweihundertsten Byte:

```
tail -c +200 notes | pg
```

4. Verfolgen des Wachstums einer Datei:

```
tail -f accounts
```

Damit werden die letzten 10 Zeilen der Datei `accounts` angezeigt. Der Befehl **tail** zeigt weiterhin die Zeilen so an, wie sie in der Datei `accounts` hinzugefügt wurden. Die Anzeige wird so lange fortgesetzt, bis sie mit der Tastenfolge Strg-C beendet wird.

Den Dateityp anzeigen

Der Befehl **file** liest die im Parameter *Datei* bzw. **-f Dateiliste** angegebenen Dateien, führt mit jeder Datei eine Reihe von Tests durch und versucht, sie nach Dateitypen zu klassifizieren. Anschließend schreibt er die Dateitypen in die Standardausgabeeinheit.

Handelt es sich um eine ASCII-Datei, überprüft der Befehl **file** die ersten 512 Byte und stellt ihre Sprache fest. Handelt es sich nicht um eine ASCII-Datei, versucht der Befehl **file** weiter, eine Binärdatei von einer Textdatei mit erweiterten Zeichen zu unterscheiden.

Wird im Parameter *Datei* eine ausführbare oder Objektmoduldatei mit einer Versionsnummer größer 0 angegeben, zeigt der Befehl **file** die Versionsnummer an. Im Abschnitt zum Befehl **ld** wird die Verwendung von **a.out**-Dateien erklärt.

Der Befehl **file** verwendet die Datei `/etc/magic`, um Dateien zu identifizieren, die eine Art Dateitypanzeiger haben; d. h., jede Datei mit einer numerischen oder Zeichenfolgenkonstanten, die den Dateityp angibt.

Beispiele

1. Anzeigen der Informationsart, die eine Datei enthält:

```
file meinDat
```

Damit wird der Dateityp von `meinDat` (wie z. B. Verzeichnis, Daten, ASCII-Text, C-Programmquelle oder Archiv) angezeigt.

2. Anzeigen des Dateityps jeder in einer Liste mit Dateinamen aufgeführten Datei:

```
file -f Dateinamen
```

Damit wird der Dateityp jeder Datei in der Liste `Dateinamen` angezeigt. Jeder Dateiname muß in einer separaten Zeile stehen.

Zum Erstellen der Liste `Dateinamen`

```
ls >Dateinamen
```

geben und anschließend `Dateinamen` beliebig editieren.

Verfügbaren Speicherbereich in einem Dateisystem anzeigen

Der Befehl `df` zeigt Informationen über den gesamten und den verfügbaren Speicherbereich in einem Dateisystem an. Der Parameter *Dateisystem* gibt den Namen der Einheit an, auf der sich das Dateisystem befindet, das Verzeichnis, an das das Dateisystem angehängt ist oder den relativen Pfadnamen eines Dateisystems. Wenn der Parameter *Dateisystem* nicht angegeben wird, zeigt der Befehl `df` Informationen zu allen derzeit angehängten Dateisystemen an. Wenn eine Datei oder ein Verzeichnis angegeben wird, zeigt der Befehl `df` Informationen zu dem Dateisystem an, in dem sich die Datei befindet.

Der Befehl `df` verwendet in der Regel die Anzahl der verfügbaren Blöcke im Superblock. In einigen Ausnahmefällen können diese Zahlen falsch sein. Wird beispielsweise ein Dateisystem während der Durchführung des Befehls `df` aktiv geändert, ist die Anzahl der verfügbaren Blöcke unter Umständen nicht genau.

Hinweis: Bei einigen fernen Dateisystemen, wie z. B. Network File Systems (NFS), sind die Spalten leer, wenn der Server keine Informationen liefert.

Beispiele

1. Anzeigen von Informationen über alle angehängten Dateisysteme:

```
df
```

Wenn das System so konfiguriert ist, daß sich die Verzeichnisse `/`, `/usr`, `/site` und `/usr/venus` in separaten Dateisystemen befinden, ist die Ausgabe des Befehls `df` ähnlich dem folgenden:

Filesystem	TotalKB	free	%used	iused	%iused	Mounted on
/dev/hd0	19368	9976	48%	4714	5%	/
/dev/hd1	24212	4808	80%	5031	19%	/usr
/dev/hd2	9744	9352	4%	1900	4%	/site
/dev/hd3	3868	3856	0%	986	0%	/usr/venus

2. Anzeigen des verfügbaren Speicherbereichs in dem Dateisystem, in dem sich das aktuelle Verzeichnis befindet:

```
df .
```

Text in einer Datei bearbeiten

In den folgenden Abschnitten wird das Bearbeiten von Text in einer Datei beschrieben. Für jeden der folgenden Textbearbeitungsvorgänge werden ein Befehl und ein Beispiel angeführt:

- Dateien verknüpfen
- Abschnitte aus Dateien ausschneiden
- Abschnitte in eine Datei einfügen
- Zeilen in einer Datei numerieren
- Spalten aus einer Datei löschen
- Daten in einer Datei komprimieren
- Gepackte Dateien entpacken

Dateien verknüpfen

Der Befehl `cat` liest die durch den Parameter *Datei* beschriebenen Dateien nacheinander und schreibt sie auf die Standardausgabereinheit.

Achtung: Ausgabe in eine Eingabedatei nicht mit dem Symbol `>` (Umadressierung) umleiten, da sonst die Originaldaten in der Eingabedatei verlorengehen, weil die Shell sie abschneidet, bevor der Befehl `cat` sie lesen kann. Der Abschnitt "Eingabe und Ausgabe in der Korn Shell umadressieren" auf Seite A-35 enthält weitere Informationen.

Beispiele

1. Zur Verkettung mehrerer Dateien folgendes eingeben:

```
cat section1.1 section1.2 section1.3 >section1
```

Damit wird eine Datei mit dem Namen `section1` erstellt, in der die Dateien `section1.1`, `section1.2` und `section1.3` nacheinander aufgeführt werden.

2. Zum Anhängen einer Datei an das Ende einer anderen Datei folgendes eingeben:

```
cat section1.4 >>section1
```

Damit wird eine Kopie von `section1.4` an das Ende von `section1` angehängt. Soll die Datei ersetzt werden, ist `>` anzugeben.

Abschnitte aus Dateien ausschneiden

Der Befehl `cut` schneidet angegebene Byte, Zeichen oder Felder aus jeder Zeile einer Datei aus.

Beispiel

Anzeigen mehrerer Felder in jeder Zeile einer Datei:

```
cut -f1,5 -d: /etc/passw d
```

Damit werden die Felder mit den Anmeldenamen und vollständigen Benutzernamen der Systemkennwortdatei angezeigt. Dies sind das erste und fünfte Feld (`-f1,5`), getrennt durch Doppelpunkte (`-d:`).

Wenn also die Datei `/etc/passwd` wie folgt aussieht:

```
root:*:0:0:User with special privileges:/:usr/bin/sh
daemon:*:1:1:/:etc:
bin:*:2:2:/:usr/bin:
sys:*:3:3:/:usr/src:
adm:*:4:4:System Administrator:/var/adm:/usr/bin/sh
pierre:*:200:200:Pierre Harper:/home/pierre:/usr/bin/sh
joan:*:202:200:Joan Brown:/home/joan:/usr/bin/sh
```

gibt der Befehl `cut` folgendes aus:

```
root:User with special privileges
daemon:
bin:
sys:
adm:System Administrator
pierre:Pierre Harper
joan:Joan Brown
```

Abschnitte in eine Datei einfügen

Der Befehl `paste` verknüpft die Zeilen mehrerer Dateien oder nachfolgende Zeilen zu einer Datei.

Beispiele

1. Zusammenfügen mehrerer Datenspalten:

```
paste names places dates > npd
```

Eine Datei mit dem Namen `npd` wird erstellt. Diese Datei enthält die Daten der Datei `names` in einer Spalte, die Daten aus `places` in einer weiteren und die Daten aus `dates` in einer dritten. Wenn der Inhalt der Dateien `names`, `places` und `dates` wie folgt aussieht:

names	places	dates
rachel	New York	February 5
jerry	Austin	March 13
mark	Chicago	June 21
marsha	Boca Raton	July 16
scott	Seattle	November 4

dann enthält `npd` folgende Daten:

rachel	New York	February 5
jerry	Austin	March 13
mark	Chicago	June 21
marsha	Boca Raton	July 16
scott	Seattle	November 4

Ein Tabulatorzeichen trennt Name, Stadt und Datum in jeder Zeile. Die Spalten sind nicht immer ausgerichtet, da die Tabulatorstopps in jeder achten Spalte gesetzt sind.

2. Trennung der Spalten durch andere Zeichen:

```
paste -d"!@" names places dates > npd
```

Die Zeichen `!` und `@` werden abwechselnd als Trennzeichen verwendet. Wenn Beispiel 1 zugrunde liegt, enthält `npd` :

```
rachel!New York@February 5
jerry!Austin@March 13
mark!Chicago@June 21
marsha!Boca Raton@July 16
scott!Seattle@November 4
```

3. Anzeige der Standardeingabe in mehreren Spalten:

```
ls | paste - - - -
```

Das aktuelle Verzeichnis wird in vier Spalten aufgeteilt. Bei jedem - (Minuszeichen) erstellt der Befehl **paste** eine Spalte mit Daten, die aus der Standardeingabe stammen. Die erste Zeile wird in die erste Spalte gestellt, die zweite Zeile in die zweite Spalte usw.

Zeilen in einer Datei numerieren

Der Befehl **nl** liest die angegebene Datei (Standardeingabe), numeriert die Zeilen in der Eingabeeinheit und schreibt die numerierten Zeilen in die Standardausgabeeinheit.

Beispiele

1. Zeilen ohne Leerstellen numerieren:

```
nl chap1
```

Zeigt eine Auflistung von `chap1` an, in der nur die Zeilen ohne Leerstellen in den Hauptabschnitten numeriert sind.

2. Alle Zeilen numerieren:

```
nl -ba chap1
```

Damit werden alle Zeilen in den Hauptabschnitten numeriert. Diese Verwendung des Befehls **nl** ist in den meisten Fällen angebracht.

Spalten aus einer Datei löschen

Mit dem Befehl **colrm** werden ausgewählte Spalten aus einer Datei gelöscht. Die Eingabe stammt aus der Standardeingabeeinheit. Die Ausgabe wird an die Standardausgabeeinheit gesendet.

Wird mit dem Befehl ein Parameter aufgerufen, werden die Spalten jeder Zeile entfernt, beginnend bei der durch den Parameter angegebenen Spalte. Werden zwei Parameter angegeben, werden die Spalten von der ersten angegebenen bis zur letzten angegebenen Spalte entfernt.

Die Spaltennumerierung beginnt mit 1.

Beispiel

Spalten aus `text.fil` löschen:

```
colrm 6 < text.fil
```

Enthält `text.fil`:

```
123456789
```

zeigt der Befehl **colrm** folgendes an:

```
12345
```

Daten in einer Datei komprimieren

Dateien können mit dem Befehl **compress** oder **pack** komprimiert werden.

Der Befehl **compress**

Der Befehl **compress** verkleinert Dateien unter Verwendung der adaptiven Lempel-Zev-Codierung. Jede mit dem Parameter *Datei* angegebene Originaldatei wird entfernt und durch eine komprimierte Datei mit der Erweiterung **.Z** ersetzt. Die komprimierte Datei behält Eignerstatus und Modi sowie den Zeitpunkt des letzten Zugriffs bzw. der letzten Änderung der Originaldatei bei. Werden keine Dateien angegeben, wird die Standardeingabe in die Standardausgabe komprimiert. Kann eine Datei durch Komprimieren nicht verkleinert werden, wird eine Nachricht in die Standardfehlerausgabe geschrieben und die Originaldatei nicht ersetzt.

Der Befehl **uncompress** stellt die Originaldateien wieder her, die mit dem Befehl **compress** komprimiert wurden. Jede mit dem Parameter *Datei* angegebene komprimierte Datei wird entfernt und durch eine erweiterte Kopie ersetzt. Die erweiterte Datei hat denselben Namen wie die komprimierte Version, allerdings ohne die Erweiterung **.Z**. Die erweiterte Datei behält Eignerstatus und Modi sowie den Zeitpunkt des letzten Zugriffs bzw. der letzten Änderung der Originaldatei bei. Werden keine Dateien angegeben, wird die Standardeingabe in die Standardausgabe erweitert.

Ähnlich wie der Befehl **uncompress** ist der Befehl **zcat**, der jedoch die erweiterte Ausgabe immer in die Standardausgabe schreibt.

Das Ausmaß der Komprimierung hängt von der Größe der Eingabe, der durch die Variable *Bit* angegebenen Zahl an Bit pro Code sowie von der Verteilung allgemeiner Unterzeichenfolgen ab. Quellencode oder englischer Text wird in der Regel um 50-60% komprimiert. Die Komprimierung mit dem Befehl **compress** wird im allgemeinen kompakter und erfordert weniger Zeit als die Komprimierung mit Hilfe der Huffman-Codierung (wie sie im Befehl **pack** verwendet wird) oder der angepaßten Huffman-Codierung.

Beispiel

Die Datei `foo` komprimieren und den Prozentsatz der Komprimierung in die Standardfehlerausgabe schreiben:

```
compress -v foo
```

Der Befehl **pack**

Der Befehl **pack** speichert die durch den Parameter *Datei* angegebene(n) Datei(en) in komprimierter Form ab. Die Eingabedatei wird durch eine gepackte Datei ersetzt und unter einem Namen abgespeichert, der von der Originaldatei abgeleitet ist (*Datei.z*). Zugriffsmodus, Zugriffs- und Änderungsdaten bleiben gegenüber der ursprünglichen Datei unverändert. Der Name der Eingabedatei darf nicht mehr als 253 Byte enthalten, damit noch Platz für das angefügte Suffix **.z** bleibt. Wurde der Befehl **pack** erfolgreich ausgeführt, wird die ursprüngliche Datei gelöscht. Komprimierte Dateien können mit den Befehlen **unpack** und **pcat** wieder auf ihre ursprüngliche Form gebracht werden.

Kann mit dem Befehl **pack** keine kleinere Datei erstellt werden, wird die Verarbeitung abgebrochen und die Meldung angezeigt, daß kein Speicherplatz eingespart werden kann. (Im allgemeinen tritt dieser Fall bei kleinen Dateien oder bei Dateien mit gleichmäßiger Zeichenverteilung auf.) Die Größe des eingesparten Speicherplatzes hängt von der Größe der Eingabedatei und der Häufigkeitsverteilung der Zeichen ab. Da jede Datei **.z** mit einer Entschlüsselungsbaumstruktur beginnt, kann im allgemeinen mit Dateien, die kleiner als drei Blöcke sind, kein Speicherplatz eingespart werden. Normalerweise wird der Speicherplatz für Textdateien um 25 bis 40 Prozent verringert.

Der Rückkehrcode des Befehls **pack** ist die Anzahl der Dateien, die der Befehl nicht packen konnte. Unter den folgenden Voraussetzungen kann nicht gepackt werden:

- Die Datei ist bereits gepackt.
- Der Name der Eingabedatei umfaßt mehr als 253 Byte.

- Die Datei weist Link-Verbindungen auf.
- Die Datei ist ein Verzeichnis.
- Die Datei kann nicht geöffnet werden.
- Durch das Packen werden keine Speicherblöcke eingespart.
- Die entsprechende *Datei.z* besteht bereits
- Die Datei *.z* kann nicht erstellt werden.
- Bei der Verarbeitung tritt ein E/A-Fehler auf.

Beispiel

Zum Verdichten von Dateien:

```
pack chap1 chap2
```

Damit werden *chap1* und *chap2* komprimiert und durch die Dateien mit den Namen **chap1.z** und **chap2.z** ersetzt. Der Befehl **pack** zeigt den Prozentsatz der Abnahme jeder Datei an.

Gepackte Dateien entpacken

Der Befehl **unpack** entpackt Dateien, die mit dem Befehl **pack** erstellt wurden. Für jede angegebene Datei sucht der Befehl **unpack** nach der *Datei.z*. Handelt es sich dabei um eine gepackte Datei, wird sie mit dem Befehl **unpack** durch die entpackte Version ersetzt. Der Befehl **unpack** benennt die neue Datei wieder mit dem ursprünglichen Namen, indem er dazu das Suffix *.z* von der *Datei* entfernt. Zugriffsmodi, Zugriffs- und Änderungsdaten, sowie der Eigenerstatus der neuen und der gepackten Datei entsprechen sich.

Der Befehl **unpack** kann nur an Dateien ausgeführt werden, die auf *.z* enden. Wenn der angegebene Dateiname nicht auf *.z* endet, wird von **unpack** dieses Suffix hinzugefügt und das Verzeichnis nach Dateinamen mit diesem Suffix durchsucht.

Der Rückkehrcode entspricht der Anzahl der Dateien, die der Befehl **unpack** nicht entpacken konnte. Eine Datei kann nicht entpackt werden, wenn folgendes auftritt:

- Der Dateiname (außer *.z*) besteht aus mehr als 253 Byte.
- Die Datei kann nicht geöffnet werden.
- Bei der Datei handelt es sich nicht um eine gepackte Datei.
- Es gibt bereits eine Datei mit dem Namen der entpackten Datei.
- Die entpackte Datei kann nicht erstellt werden.

Hinweis: Der Befehl **unpack** gibt eine Warnung auf der Standardausgabeeinheit aus, wenn die zu entpackende Datei Verbindungen zu anderen Dateien hat. Die neu entpackte Datei hat einen anderen Dateiindex als die gepackte Datei, aus der sie erstellt wurde. Jedoch sind noch alle anderen Dateien, die eine Verbindung zum ursprünglichen Dateiindex der gepackten Datei haben, in gepacktem Zustand vorhanden.

Beispiel

Entpacken gepackter Dateien:

```
unpack chap1.z chap2
```

Dadurch werden die gepackten Dateien *chap1.z* und *chap2.z* entpackt und durch die Dateien *chap1* und *chap2* ersetzt. An den Befehl **unpack** können auch Dateinamen ohne das Suffix *.z* übergeben werden.

Mit Verzeichnissen arbeiten

Für jede der folgenden Verzeichnisfunktionen werden ein Befehl und ein Beispiel angeführt:

- Verzeichnisse erstellen
- Verzeichnisse verschieben
- Aktuelles Verzeichnis ermitteln
- Aktuelles Verzeichnis wechseln
- Verzeichnisse kopieren
- Inhalt von Verzeichnissen anzeigen
- Verzeichnisse löschen bzw. entfernen
- Eignerstatus für Verzeichnisse ändern
- Inhalt von Verzeichnissen vergleichen.

Verzeichnisse erstellen

Der Befehl **mkdir** legt ein oder mehrere im Parameter *Verzeichnis* angegebene(s) Verzeichnis(se) an. Jedes neue Verzeichnis enthält die Standardeinträge `.` (Punkt) und `..` (Punkt Punkt). Die Berechtigungen für neue Verzeichnisse können mit der Option **-m** *Modus* angegeben werden. Mit der Unteroutine **umask** kann die Option *Modus* standardmäßig an den Befehl **mkdir** angefügt werden.

Beispiele

1. Ein neues Verzeichnis `Test` mit Standardberechtigungen im aktuellen Arbeitsverzeichnis erstellen:

```
mkdir Test
```

2. Ein neues Verzeichnis `Test` mit den Berechtigungen `rwxr-xr-x` im zuvor erstellten Verzeichnis `/home/demo/sub1` erstellen:

```
mkdir -m 755 /home/demo/sub1/Test
```

3. Ein neues Verzeichnis `Test` mit Standardberechtigungen im Verzeichnis `/home/demo/sub2` erstellen:

```
mkdir -p /home/demo/sub2/Test
```

Die Option **-p** erstellt die Verzeichnisse `/home`, `/home/demo` und `/home/demo/sub2`, soweit diese nicht bereits existieren.

Verzeichnisse verschieben

Achtung: Der Befehl **mv** kann viele vorhandene Dateien überschreiben, wenn nicht die Option **-i** angegeben wird. Bei Verwendung der Option **-i** muß der Benutzer vor dem Überschreiben einer Datei bestätigen. Wenn die Optionen **-f** und **-i** kombiniert werden, hat die als letzte angegebene Option Vorrang.

Der Befehl **mv** (**move**) verschiebt Dateien und Verzeichnisse von einem Verzeichnis in ein anderes oder benennt eine Datei oder ein Verzeichnis um. Wird eine Datei oder ein Verzeichnis in ein neues Verzeichnis verschoben, bleibt der ursprüngliche Dateiname unverändert. (Eine Datei kann nur an eine andere Position verschoben werden.)

Beispiel

Verschieben eines Verzeichnisses:

```
mv book manual
```

Damit werden alle Dateien und Verzeichnisse in `book` in das Verzeichnis `manual` verschoben, sofern dieses vorhanden ist. Ist es nicht vorhanden, wird das Verzeichnis `book` in `manual` umbenannt. Siehe Sicherheitshinweis **Achtung**.

Aktuelles Verzeichnis ermitteln

Der Befehl `pwd` gibt den vollständigen Pfad des aktuellen Verzeichnisses (vom Stammverzeichnis ausgehend) auf der Standardausgabeeinheit aus. Alle Verzeichnisse werden durch einen / (Schrägstrich) voneinander getrennt. Das Stammverzeichnis ist das erste durch einen / (Schrägstrich) angegebene Verzeichnis. Das als letztes angegebene Verzeichnis ist das aktuelle Verzeichnis.

Beispiel

Das aktuelle Verzeichnis anzeigen:

```
pwd
```

Es wird etwa folgender vollständiger Pfadname zum aktuellen Verzeichnis angezeigt:

```
/home/thomas
```

Aktuelles Verzeichnis wechseln

Der Befehl `cd` wechselt vom aktuellen in ein anderes Verzeichnis. Für das angegebene Verzeichnis muß der Benutzer Ausführungsberechtigung (Suchberechtigung) besitzen.

Wird kein *Verzeichnis* angegeben, wechselt der Befehl `cd` in das jeweilige Anmeldeverzeichnis (**\$HOME** in den Umgebungen **ksh** und **bsh** oder **\$home** in der Umgebung **csh**). Wird das Verzeichnis in Form eines vollständigen Pfadnamens angegeben, so wird es zum aktuellen Verzeichnis. Ein vollständiger Pfadname beginnt mit / (Schrägstrich = Stammverzeichnis), einem . (Punkt = aktuelles Verzeichnis) oder einem .. (Punkt Punkt = Vaterverzeichnis). Wird das Verzeichnis nicht in Form eines vollständigen Pfadnamens angegeben, sucht der Befehl `cd` in einem in der Shellvariablen **\$CDPATH** (oder der Variablen **\$cdpath csh**) angegebenen entsprechenden Pfad nach dem Verzeichnis. Diese Variable hat die gleiche Syntax und eine ähnliche Semantik wie die Shell-Variablen **\$PATH** (oder die Variable **\$path csh**).

Beispiele

1. Wechseln in das Benutzerverzeichnis:

```
cd
```

2. Wechseln in ein beliebiges Verzeichnis, z. B. `/usr/include`:

```
cd /usr/include
```

Damit wird `/usr/include` zum aktuellen Verzeichnis.

3. In der Verzeichnisstruktur eine Ebene nach unten gehen:

```
cd sys
```

Wenn `/usr/include` das aktuelle Verzeichnis ist und ein Unterverzeichnis mit dem Namen `sys` enthält, wird dadurch `/usr/include/sys` zum aktuellen Verzeichnis.

4. In der Verzeichnisstruktur eine Ebene nach oben gehen:

```
cd ..
```

Der spezielle Dateiname `..` (Punkt Punkt) bezieht sich auf das Verzeichnis, das dem aktuellen Verzeichnis direkt übergeordnet ist.

Verzeichnisse kopieren

Der Befehl **cp** kopiert den Inhalt einer in der *Quelldatei* oder im *Quellenverzeichnis* angegebenen Datei bzw. eines Verzeichnisses in eine in der *Zieldatei* oder im *Zielverzeichnis* angegebene Datei oder ein Verzeichnis. Besteht die als *Zieldatei* angegebene Datei bereits, werden die darin enthaltenen Daten überschrieben. Wenn mehr als eine *Quelldatei* kopiert wird, muß das Ziel ein Verzeichnis sein.

Um eine Kopie der *Quelldatei* in ein Verzeichnis zu stellen, einen Pfad zu einem bestehenden Verzeichnis im *Zielverzeichnis* anlegen. Dateien behalten beim Kopieren in ein Verzeichnis ihren Namen bei, wenn am Ende des Pfads kein neuer Dateiname angegeben wird. Der Befehl **cp** kopiert ganze Verzeichnisse in andere Verzeichnisse, wenn die Option **-r** oder **-R** angegeben wird.

Beispiele

1. Kopieren aller Dateien eines Verzeichnisses in ein neues Verzeichnis:

```
cp /home/liesl/kunden/* /home/seppl/klienten
```

Damit werden die Dateien im Verzeichnis `kunden` in das Verzeichnis `klienten` kopiert.

2. Kopieren eines Verzeichnisses samt seiner Dateien und Unterverzeichnisse in ein anderes Verzeichnis:

```
cp -R /home/seppl/kunden /home/seppl/klienten
```

Damit wird das Verzeichnis `kunden`, seine Dateien, seine Unterverzeichnisse und die Dateien in den Unterverzeichnissen in das Verzeichnis `klienten/kunden` kopiert.

Informationen zum Kopieren von Verzeichnissen in andere Verzeichnisse ohne die Angabe einer zusätzlichen Unterverzeichnisebene enthält der Abschnitt zum Befehl **cp** im Handbuch *Commands Reference*.

Inhalt von Verzeichnissen anzeigen

Jeder Benutzer kann sich den Inhalt der Verzeichnisse, auf die er Zugriff hat, mit dem Befehl **li** oder **ls** anzeigen lassen.

Der Befehl li

Der Befehl **li** listet Informationen zu allen angegebenen *Dateien* und den Dateien in jedem angegebenen *Verzeichnis* auf. Ist *Datei* eine Archivdatei und wird die Option **-Rq** angegeben, listet der Befehl **li** die Dateien im Archiv auf.

Der Befehl **li** listet reguläre Dateien vor Verzeichnissen auf und sortiert Datei- und Verzeichnisnamen. Steuerzeichen werden in Dateinamen in erweiterter Form angezeigt (z. B. `^ D, \177`). Die Sortierreihenfolge der Dateinamen und das Format von Datums- und Zeitangaben sind von der jeweiligen Landessprache abhängig. Das Kapitel "National Language Support Overview for System Management" im Handbuch *System Management Guide* enthält weitere Angaben.

Beispiele

1. Auflisten der Dateien im aktuellen Verzeichnis in alphabetischer Reihenfolge:

```
li
```

2. Auflisten aller Dateien im aktuellen Verzeichnis einschließlich der Dateinamen, die mit einem `.` (Punkt) beginnen:

```
li -a
```

3. Anzeigen detaillierter Informationen:

```
li -l chap1 .profile
```

Eine ausführliche Auflistung mit detaillierten Informationen über `chap1` und die Datei `.profile` wird angezeigt. Alle möglicherweise benötigten Informationen werden angezeigt. Durch die Kombination des Befehls `li` mit der Option `-x` können weitere Informationen angezeigt werden.

4. Anzeigen detaillierter Informationen zu einem Verzeichnis:

```
li -d -l . manual manua l/chap1
```

Eine ausführliche Auflistung für die Verzeichnisse `.` (Punkt) und `manual` und für die Datei `manual/chap1` wird angezeigt.

5. Auflisten des Inhalts aller Verzeichnisse in einer Verzeichnisbaumstruktur:

```
li -R manual
```

Die Namen in jedem Unterverzeichnis der Verzeichnisbaumstruktur, die mit `manual` beginnt, werden aufgelistet.

Der Befehl ls

Der Befehl `ls` gibt den Inhalt jedes angegebenen *Verzeichnisses* oder den Namen jeder angegebenen *Datei* zusammen mit allen anderen Informationen, die mit den Optionen angefordert werden, auf der Standardausgabeeinheit aus. Wird keine *Datei* und kein *Verzeichnis* angegeben, zeigt `ls` den Inhalt des aktuellen Verzeichnisses an.

Standardmäßig zeigt `ls` alle Informationen in alphabetischer Reihenfolge nach Dateinamen an. Die Sortierfolge wird durch die Umgebungsvariable **LANG** oder **LC_COLLATE** festgelegt. Das Kapitel "National Language Support Overview for Programming" im Handbuch *System Management Guide* enthält weitere Einzelheiten.

Wenn der Befehl `ls` den Inhalt eines Verzeichnisses angibt, zeigt er keine Einträge zu Dateien an, die mit einem `.` (Punkt) beginnen, es sei denn, der Befehl wird mit der Option `-a` oder `-A` verwendet. Wird der Befehl vom Root ausgeführt, wird die Option `-A` standardmäßig verwendet.

Die drei wichtigsten Arten der Ausgabeformatierung sind folgende:

- Pro Zeile wird ein Eintrag aufgelistet.
- Die Einträge werden in mehreren Spalten aufgelistet, wenn entweder die Option `-C` oder `-x` angegeben wird. Die Option `-C` ist das Standardformat, wenn die Ausgabe auf einer nichtgrafikfähigen Datenstation erfolgt.
- Die Einträge werden in einer Reihe, getrennt durch Kommas, aufgelistet, wenn die Option `-m` angegeben wird.

Der Befehl `ls` verwendet die Umgebungsvariable **COLUMNS**, um die Zahl der Zeichenpositionen in der Ausgabezeile zu bestimmen. Wenn diese Variable nicht angegeben wird, liest der Befehl die Datei `terminfo`. Wenn der Befehl `ls` die Anzahl der Zeichenpositionen mit keiner der beiden Methoden bestimmen kann, verwendet er den Standardwert 80.

Der mit Hilfe der Optionen `-e` und `-l` angezeigte Modus wird folgendermaßen interpretiert:

Das erste Zeichen bedeutet:

- | | |
|----------|--|
| d | Der Eintrag ist ein Verzeichnis. |
| b | Der Eintrag ist eine blockorientierte Gerätedatei. |
| c | Der Eintrag ist eine zeichenorientierte Gerätedatei. |

- l** Der Eintrag ist eine symbolische Verbindung und es wurde entweder die Option **-N** angegeben oder die symbolische Verbindung hat nicht auf eine vorhandene Datei hingewiesen.
- p** Der Eintrag ist eine FIFO-Geräte-datei.
- s** Der Eintrag ist ein lokaler Socket.
- Der Eintrag ist eine gewöhnliche Datei.

Die nächsten neun Zeichen werden in drei Gruppen zu je drei Zeichen aufgeteilt. Die ersten drei Zeichen geben die Berechtigung des Eigners an. Die nächsten drei Zeichen geben die Berechtigung der anderen Benutzer in der Gruppe an. Die letzten drei Zeichen geben die Berechtigung aller anderen Benutzer an, die auf die Datei zugreifen können. Die drei Zeichen innerhalb der einzelnen Gruppen geben die Lese-, Schreib- und Ausführungsberechtigung der Datei an. Die Ausführungsberechtigung für ein Verzeichnis ermöglicht die Suche nach einer angegebenen Datei in diesem Verzeichnis.

Die Berechtigungen werden wie folgt angezeigt:

- r** Lesen
- w** Schreiben (Editieren)
- x** Ausführen (Suchen)
- Keine Zugriffsberechtigung.

Das Zeichen für Gruppenausführungsberechtigung ist **s**, wenn für die Datei der Modus set-group-ID (Gruppen-ID setzen) gesetzt ist. Das Zeichen für Benutzerausführungsberechtigung ist **s**, wenn für die Datei der Modus set-user-ID (Benutzer-ID setzen) gesetzt ist. Das letzte Zeichen des Modus (normalerweise **x** oder **-**) ist **T**, wenn das Bit 01000 (oktal) des Modus aktiviert ist. Nähere Informationen zur Bedeutung dieses Modus enthält der Abschnitt zum Befehl **chmod**. Die Meldungen von Modus set-ID und Bit 01000 des Modus stehen in Großbuchstaben (**S** bzw. **T**), wenn die entsprechende Ausführungsberechtigung nicht gesetzt wurde.

Der Modus, der mit der Option **-e** angezeigt wird, entspricht dem mit der Option **-l**, mit Ausnahme eines zusätzlich angezeigten 11. Zeichens, das wie folgt interpretiert wird:

- +** Gibt an, daß eine Datei erweiterte Sicherheitsinformationen enthält. Es liegen im Modus z. B. die erweiterten Attribute für **ACL**, **TCB** oder **TP** vor.
Das Dateiattribut **ACL** (Access Control Information) wird mit dem Befehl **aclget** angezeigt. Der Wert der Attribute **TCB** und **TP** wird mit dem Befehl **chtcb** angezeigt.
- Gibt an, daß eine Datei keine erweiterten Sicherheitsinformationen enthält.

Wenn die Größe der Dateien in einem Verzeichnis aufgelistet wird, zeigt der Befehl **ls** die Gesamtzahl der Blöcke einschließlich indirekter Blöcke an.

Beispiele

1. Auflisten aller Dateien im aktuellen Verzeichnis:

```
ls -a
```

Alle Dateien werden aufgelistet, einschließlich **.** (Punkt), **..** (Punkt Punkt) und andere Dateien, deren Namen mit einem Punkt beginnen.

2. Anzeigen detaillierter Informationen:

```
ls -l chap1 .profile
```

Eine ausführliche Liste mit detaillierten Informationen über **chap1** und **.profile** wird angezeigt.

3. Anzeige detaillierter Informationen zu einem Verzeichnis:

```
ls -d -l . manual manual/chap1
```

Eine ausführliche Auflistung für die Verzeichnisse `.` und `manual` sowie für die Datei `manual/chap1` wird angezeigt. Ohne die Option `-d` werden nur die Dateien in den Verzeichnissen `.` und `manual` anstelle der detaillierten Informationen über die Verzeichnisse selbst angezeigt.

Verzeichnisse löschen bzw. entfernen

Der Befehl `rmdir` löscht das im Parameter *Verzeichnis* angegebene Verzeichnis aus dem System. Um ein Verzeichnis zu löschen, darf es keine Dateien und Unterverzeichnisse enthalten, und der Benutzer muß Berechtigung im Vaterverzeichnis haben. Mit dem Befehl `ls-al` kann überprüft werden, ob das Verzeichnis leer ist.

Beispiele

1. Dateien und Unterverzeichnisse aus einem Verzeichnis entfernen und ein Verzeichnis löschen:

```
rm mydir/* mydir/*
rmdir mydir
```

Damit wird der Inhalt von `mydir` entfernt und anschließend das leere Verzeichnis gelöscht. Der Befehl `rm` zeigt eine Fehlermeldung an, daß die Verzeichnisse `.` (Punkt) und `..` (Punkt Punkt) gelöscht werden, und anschließend löscht der Befehl `rmdir` die Verzeichnisse.

Dabei ist zu beachten, daß `rm mydir/* mydir/*` zuerst alle Dateien löscht, deren Namen nicht mit einem Punkt beginnen, und anschließend alle Dateien, deren Namen mit einem Punkt beginnen. Der Benutzer kann nicht erkennen, ob das Verzeichnis Dateinamen enthält, die mit einem Punkt beginnen, weil der Befehl `ls` sie normalerweise nicht auflistet, es sei denn, die Option `-a` wird verwendet.

2. Die Verzeichnisse `/home`, `/home/demo` und `/home/demo/mydir` löschen:

```
rmdir -p /home/demo/mydir
```

Damit wird zuerst das Verzeichnis `/mydir` und anschließend werden der Reihe nach die Verzeichnisse `/demo` und `/home` gelöscht. Wenn ein Verzeichnis noch Dateien oder Unterverzeichnisse enthält, oder auf das Verzeichnis kein Schreibzugriff besteht, wird der Befehl abgebrochen.

Eignerstatus für Verzeichnisse ändern

Der Befehl `chown` ändert den Eignerstatus der mit dem Parameter *Datei* angegebenen Datei. Der mit dem Parameter *Eigner* angegebene Benutzer wird neuer Dateieigner. Für den Parameter *Eigner* kann entweder eine Benutzer-ID oder ein Anmeldename angegeben werden, der in der Datei `/etc/passwd` enthalten ist. Wahlweise kann auch eine Gruppe angegeben werden. Die Gruppe kann entweder als Gruppen-ID oder als Gruppenname angegeben werden, der in der Datei `/etc/group` enthalten ist.

Bei Angabe der Option `-R` durchläuft der Befehl `chown` das angegebene Verzeichnis rekursiv von oben nach unten. Wenn symbolische Link-Verbindungen mit einbezogen werden, wird der Eignerstatus einer Datei oder eines Verzeichnisses geändert, auf die bzw. das in der Link-Verbindung hingewiesen wird; der Eignerstatus der symbolischen Link-Verbindung ändert sich nicht.

Hinweis: Nur der Root kann den Eignerstatus einer Datei ändern. Bei Angabe der Option `-f` erfolgt keine Fehlermeldung.

Beispiele

1. Den Eigner der Datei `program.c` ändern:

```
chown jim program.c
```

Die Benutzerzugriffsberechtigungen für `program.c` gelten jetzt für `jim`. Als Eigner kann `jim` mit dem Befehl `chmod` anderen Benutzern Zugriffsberechtigung für `program.c` erteilen oder nicht.

2. Eigner und Gruppe aller Dateien im Verzeichnis `/tmp/src` in Eigner `john` und Gruppe `build` ändern:

```
chown -R john:build /tmp/src
```

Inhalt von Verzeichnissen vergleichen

Der Befehl `dircmp` vergleicht die Verzeichnisse *Verzeichnis1* und *Verzeichnis2* und schreibt Informationen über ihren Inhalt auf die Standardausgabeeinheit. Zuerst vergleicht der Befehl `dircmp` die Dateinamen in jedem Verzeichnis. Wenn beide Verzeichnisse Dateien mit gleichem Namen enthalten, vergleicht der Befehl `dircmp` den Inhalt beider Dateien.

Als Ausgabe führt der Befehl `dircmp` Dateien auf, die nur in einem Verzeichnis vorhanden sind. Anschließend werden die Dateien aufgeführt, die in beiden Verzeichnissen mit identischem Namen, aber mit unterschiedlichem Inhalt vorhanden sind. Ohne Option werden auch die Dateien aufgeführt, die in beiden Verzeichnissen sowohl einen identischen Inhalt als auch einen identischen Namen haben.

Beispiele

1. Zusammenfassen der Unterschiede zwischen Dateien zweier Verzeichnisse:

```
dircmp proj.ver1 proj.ver2
```

Damit wird eine Zusammenfassung der Unterschiede zwischen den Verzeichnissen `proj.ver1` und `proj.ver2` angezeigt. Die Zusammenfassung zeigt getrennte Listen der Dateien an, die nur in dem einen oder dem anderen Verzeichnis gefunden wurden, und derjenigen, die in beiden gefunden wurden. Wird eine Datei in beiden Verzeichnissen gefunden, stellt der Befehl `dircmp` fest, ob die beiden Kopien identisch sind oder nicht.

2. Anzeigen von Einzelheiten der Unterschiede zwischen beiden Dateien:

```
dircmp -d -s proj.ver1 proj.ver2
```

Die Option `-s` unterdrückt Informationen über identische Dateien. Die Option `-d` erstellt eine Auflistung `diff` aller unterschiedlichen Dateien aus beiden Verzeichnissen.

DOS-Dateien umwandeln

In den folgenden Beispielen wird die Umwandlung von DOS-Dateien in Betriebssystemdateien und umgekehrt beschrieben:

- DOS-Dateien in AIX-Dateien kopieren
- AIX-Dateien in DOS-Dateien kopieren
- DOS-Dateien löschen
- Inhalt des DOS-Verzeichnisses auflisten

DOS-Dateien in AIX-Dateien kopieren

Der Befehl **dosread** kopiert die angegebene DOS-Datei auf die Standardausgabereinheit oder in die angegebene AIX-Datei2. Wenn für Datei kein Pfad angegeben wird, wird die DOS-Datei in das Stammverzeichnis kopiert. Wenn keine AIX-Datei angegeben wird, wird die DOS-Datei auf die Standardausgabereinheit geschrieben.

Hinweise:

1. Im Gegensatz zur Shell haben bei diesem Befehl die Platzhalterzeichen * und ? (Stern und Fragezeichen) keine Sonderbedeutung. Wird keine Dateinamenerweiterung angegeben, wird der Dateiname beim Vergleichen so behandelt, als sei eine Erweiterung mit Leerzeichen angegeben worden.
2. Dieser Befehl muß **dosread** heißen.

Beispiele

1. Kopieren einer Textdatei von einer DOS-Diskette in das AIX-Dateisystem:

```
dosread -a chap1.doc chap1
```

Damit wird die DOS-Textdatei \CHAP1.DOC von **/dev/fd0** in die AIX-Datei chap1 im aktuellen Verzeichnis kopiert.

2. Kopieren einer Binärdatei von einer DOS-Diskette in das Dateisystem AIX:

```
dosread /survey/test.dta /home/fran/testdata
```

Damit wird die DOS-Datendatei \SURVEY\TEST.DTA von **/dev/fd 1** in die AIX-Datei /home/fran/testdata kopiert.

AIX-Dateien in DOS-Dateien kopieren

Der Befehl **doswrite** kopiert die angegebene AIX-Datei in die angegebene DOS-Datei.

Hinweise:

1. Im Gegensatz zur Shell haben bei diesem Befehl die Platzhalterzeichen * und ? (Stern und Fragezeichen) keine Sonderbedeutung. Wird keine Dateinamenerweiterung angegeben, wird der Dateiname beim Vergleichen so behandelt, als sei eine Erweiterung mit Leerzeichen angegeben worden.
2. Dieser Befehl muß **doswrite** heißen.

Beispiele

1. Kopieren einer Textdatei vom AIX-Dateisystem auf eine DOS-Diskette:

```
doswrite -a chap1 chap1.doc
```

Damit wird die AIX-Datei chap1 im aktuellen Verzeichnis in die DOS-Textdatei \CHAP1.DOC auf **/dev/fd0** kopiert.

2. Kopieren einer Binärdatei vom AIX-Dateisystem auf eine DOS-Diskette:

```
doswrite /home/fran/testdata /survey/test.dta
```

Damit wird die AIX-Datendatei /home/fran/testdata in die DOS-Datei \SURVEY\TEST.DTA auf /dev/fd1 kopiert.

DOS-Dateien löschen

Der Befehl **dosdel** löscht die angegebene DOS-Datei.

Die Namenskonventionen der DOS-Dateien werden bis auf eine Ausnahme beibehalten. Da das Zeichen \ (umgekehrter Schrägstrich) im Betriebssystem eine Sonderbedeutung annehmen kann, muß als Begrenzer zwischen den Unterverzeichnissen bei einem DOS-Pfadnamen ein / (Schrägstrich) verwendet werden. Der Befehl **dosdel** wandelt Kleinbuchstaben im Datei- oder Verzeichnisnamen in Großbuchstaben um, bevor er die Platte überprüft. Da davon ausgegangen werden muß, daß es sich bei allen Pfadnamen um vollständige und nicht um relative Pfadnamen handelt, muß der erste Schrägstrich (/) nicht angegeben werden.

Beispiel

Löschen einer DOS-Datei auf der Standardeinheit (/dev/fd0):

```
dosdel file.ext
```

Inhalt des DOS-Verzeichnisses auflisten

Der Befehl **dosdir** zeigt Informationen über die angegebenen DOS-Dateien oder -Verzeichnisse an.

Die Namenskonventionen der DOS-Dateien werden bis auf eine Ausnahme beibehalten. Da das Zeichen \ (umgekehrter Schrägstrich) im Betriebssystem eine Sonderbedeutung annehmen kann, muß als Begrenzer zwischen den Unterverzeichnissen bei einem DOS-Pfadnamen ein / (Schrägstrich) verwendet werden. Der Befehl **dosdir** setzt Kleinbuchstaben im Datei- oder Verzeichnisnamen in Großbuchstaben um, bevor die Platte überprüft wird. Da davon ausgegangen werden muß, daß es sich bei allen Pfadnamen um vollständige und nicht um relative Pfadnamen handelt, muß der erste Schrägstrich (/) nicht angegeben werden.

Beispiel

Lesen eines Verzeichnisses der DOS-Dateien auf /dev/fd0:

```
dosdir
```

Anschließend werden die Namen der Dateien und Informationen über den Plattenspeicherplatz aufgeführt.

```
PG3-25.TXT  
PG4-25.TXT  
PG5-25.TXT  
PG6-25.TXT  
Free space: 312320 bytes
```

Überblick über Möglichkeiten der Sicherung

Das Betriebssystem bietet verschiedene Möglichkeiten, Dateisysteme, Verzeichnisse und Dateien zu sichern. Das Erstellen von Dateien und Verzeichnissen erfordert einen beachtlichen Aufwand an Zeit und Mühe. Dateien können relativ leicht (versehentlich oder absichtlich) geändert oder gelöscht werden. Durch systematische Vorgehensweise bei der Sicherung der Dateisysteme sollte es immer möglich sein, die aktuellste Version einer Datei oder eines Dateisystems ohne größere Schwierigkeiten zurückzuschreiben.

Hinweis: Wenn durch den Ausfall einer Festplatte Daten auf der Platte verlorengehen, besteht die einzige Möglichkeit, zerstörte Daten wiederherzustellen, darin, die Information von der Sicherungskopie zurückzuschreiben.

Einführung in die Möglichkeiten der Sicherung

Es gibt verschiedene Möglichkeiten, Daten zu sichern. Am häufigsten wird eine Standardsicherung in Form einer Kopie des Dateisystems, des Verzeichnisses oder der Datei verwendet. Diese Sicherung wird für die Dateiübertragung oder für den Fall angefertigt, daß die Originaldaten unabsichtlich geändert oder zerstört werden. Eine andere Möglichkeit ist die Teilsicherung durch Kopieren der Daten, auf die seit Erstellen der letzten Sicherungsversion zugegriffen (ausgenommen Lesezugriff) wurde. Ferner besteht die Möglichkeit der Archivsicherung. Bei dieser Methode wird eine Kopie einer oder mehrerer Dateien oder einer gesamten Datenbank zur Verwendung als Referenzmaterial, als Protokoll oder für das Wiederherstellen angefertigt. Diese Kopie wird im Fall der Beschädigung oder des Verlusts der Originaldaten herangezogen. Ein Archiv wird gewöhnlich verwendet, wenn die entsprechenden Daten von dem System entfernt werden.

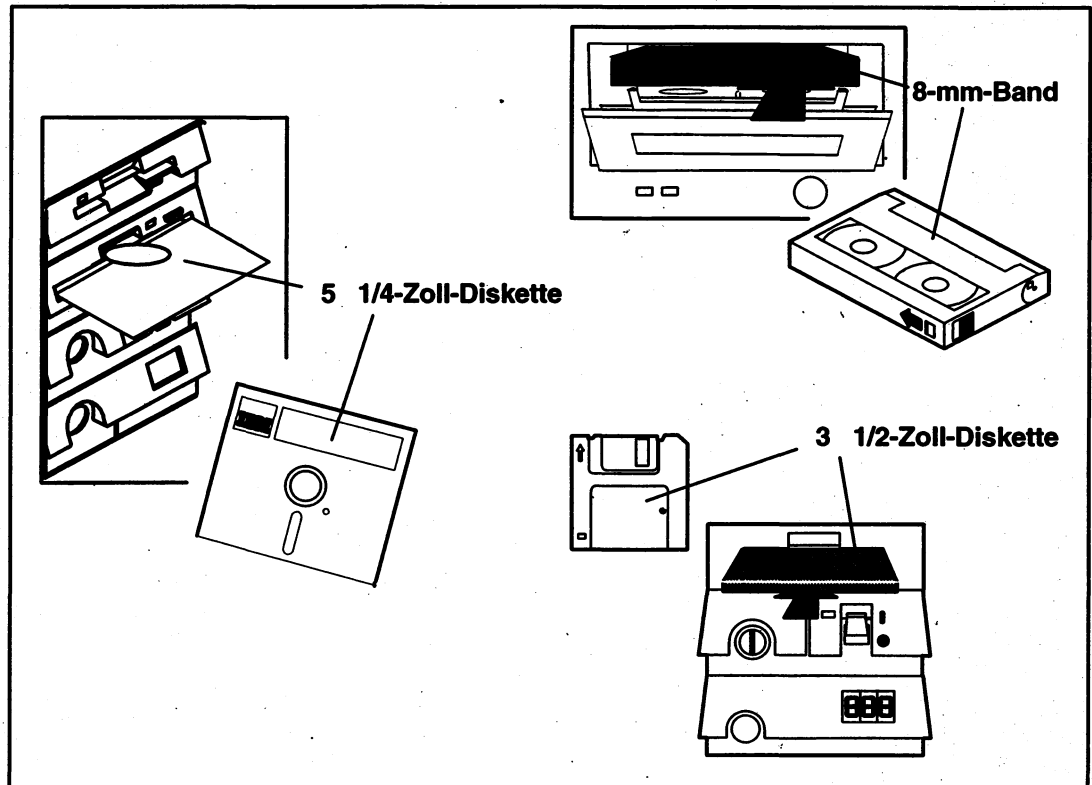
Mit dem Befehl **pack** können Dateien für die Speicherung verdichtet werden. Durch die Anwendung des Befehls **unpack** werden die zurückgeschriebenen Dateien wieder entpackt. Der Vorgang des Packens und Entpackens von Dateien beansprucht zwar Zeit, gepackte Dateien benötigen aber weniger Speicherbereich auf dem Sicherungsdatenträger.

Zum Erstellen von Sicherungen und Archiven stehen mehrere Befehle zur Verfügung. Aus diesem Grund muß für die gesicherten Daten vermerkt werden, welcher Befehl zum Sichern verwendet wurde und ob die Sicherung nach Namen oder Dateiindex erfolgte. Der Befehl **backup** wird am häufigsten verwendet, aber andere Befehle können für bestimmte Zwecke besser geeignet sein. Zum Beispiel:

backup	Sichert Dateien nach Namen oder Dateiindex.
cpio	Kopiert Dateien in und aus Archivierungsspeichern.
dd	Konvertiert und kopiert eine Datei.
tar	Verwaltet Archivsicherungen.
rdump	Sichert Dateien nach Dateiindex auf eine ferne Einheit.
pax	Erstellt und löscht Archive der Befehle tar und cpio .

Arten von Sicherungsdatenträgern

Sicherungen können auf verschiedenen Sicherungsdatenträgern erstellt werden. Welche Sicherungsdatenträger für die jeweilige Systemkonfiguration des Benutzers zur Verfügung stehen, hängt sowohl von der Hard- als auch von der Software ab. Am häufigsten verwendet werden die 5 1/4-Zoll-Diskette, das 8-mm-Band, das 9-Spur-Band und die 3 1/2-Zoll-Diskette.



Standardmäßig werden Disketten als Sicherungsträger verwendet. Die Ausgabe des Befehls **backup** wird automatisch auf die Einheit **/dev/rfd0** (das Diskettenlaufwerk) geschrieben, es sei denn mit dem Befehl **backup -f** wird eine andere Einheit angegeben. Zum Sichern auf die standardmäßige Bänderinheit **/dev/rmt0** eingeben.

Achtung: Durch Ausführen der Befehls **backup** werden alle auf dem Sicherungsträger vorhandenen Daten gelöscht.

■ Daten zurückschreiben

Es gibt verschiedene Möglichkeiten zum Zurückschreiben von gesicherten Daten. Die zu verwendende Methode hängt davon ab, mit welchem Befehl die Daten gesichert wurden.

Sicherungen oder Archivsicherungen können nur richtig zurückgeschrieben werden, wenn bekannt ist, mit welchem Befehl diese erstellt wurden. Jeder Sicherungsvorgang gibt darüber Aufschluß, wie diese Daten zurückgeschrieben werden können. Bei Verwendung des Befehls **backup** kann der Benutzer z. B. angeben, ob die Sicherung nach Dateiindexnummer oder nach Dateinamen erfolgen soll. Dementsprechend muß beim Zurückschreiben dieser Sicherung ebenfalls die Option nach Dateiindex oder nach Namen angegeben werden.

Es folgt eine Liste von Befehlen zum Zurückschreiben von gesicherten Daten:

restore	Schreibt Dateien zurück, die mit dem Befehl backup erstellt wurden.
rrestore	Schreibt Dateisysteme, die auf einer fernen Maschine gesichert wurden, auf die lokale Maschine zurück.
cpio	Kopiert Dateien in und aus Archivierungsspeichern.
tar	Verwaltet Archive.

Richtlinien für die Vorgehensweise beim Sichern

Es gibt keine Vorgehensweise beim Sichern, die den Anforderungen aller Benutzer gleichermaßen entspricht. Eine ideale Vorgehensweise für ein System mit einem Benutzer muß zum Beispiel nicht unbedingt für ein System mit fünf oder zehn verschiedenen Benutzern angebracht sein. Ebenso wäre eine optimale Vorgehensweise für ein System, auf dem die Daten täglich geändert werden, ungünstig für ein System, auf dem die Daten nicht regelmäßig geändert werden. Der Benutzer muß selbst entscheiden, welche Methode für sein System am geeignetsten ist. Dabei sollten jedoch folgende Richtlinien in Betracht gezogen werden:

- **Eine größere Menge verlorener Daten muß wiederhergestellt werden können.**

Kann bei einem Fehler an einer einzigen Festplatte am System noch weiter gearbeitet werden? Kann das System wiederhergestellt werden, wenn Fehler an allen Festplatten aufgetreten sind? Könnte das System nach Verlust der Sicherungsdisketten oder -bänder wiederhergestellt werden? Bei den Überlegungen zur besten Vorgehensweise beim Sichern zum Wiederherstellen des Systems sind alle Möglichkeiten eines Datenverlustes (auch Diebstahl oder Brand) in Betracht zu ziehen.

- **Sicherungen in regelmäßigen Abständen überprüfen.**

Auch die Sicherungsdatenträger und die entsprechende Hardware können fehlerhaft sein. Eine umfassende Bibliothek mit Sicherungsbändern oder -disketten ist nutzlos, wenn die gesicherten Daten nicht wieder auf eine Festplatte zurückgeschrieben werden können. Deshalb sollte überprüft werden, ob die Sicherungskopien fehlerfrei gelesen werden können. Dazu die gesicherten Dateien regelmäßig zurückschreiben (z. B. auf die Einheit `/dev/null`). Wird auf Disketten gesichert und stehen mehrere Laufwerke zur Verfügung, sollten die Disketten nicht unbedingt von dem Laufwerk aus gelesen werden, von dem aus die Sicherung erstellt wurde. Deshalb sollten alle Sicherungen der Stufe 0 zweifach auf Disketten vorgenommen werden. Wird eine Datenstrombandeinheit für Sicherungen verwendet, können mit dem Befehl `tapechk` grundlegende Konsistenzprüfungen für das Band ausgeführt werden.

- **Alte Sicherungskopien aufbewahren**

Die Sicherungsdatenträger in regelmäßigen Abständen wiederverwenden. Es sollten jedoch nicht alle Sicherungsdatenträger wiederverwendet werden. Es kann Monate dauern, bis ein Systembenutzer die Beschädigung oder den Verlust einer wichtigen Datei bemerkt. Aus diesem Grunde sollten alte Sicherungskopien aufgehoben werden. Nachfolgend ein Beispiel für die turnusmäßige Wiederverwendung von Sicherungsbändern oder -disketten:

- Einmal pro Woche alle Disketten mit den Tagessicherungen der vorhergehenden Woche, außer der vom Freitag, wiederverwenden.
- Einmal pro Monat die Disketten mit den Tagessicherungen von Freitagen, außer der vom letzten Freitag des Monats, wiederverwenden. Dadurch stehen jederzeit die Sicherungen der vergangenen vier Freitage zur Verfügung.
- Einmal im Quartal die Disketten des vorherigen Quartals, außer der des letzten Monats, wiederverwenden. Die Diskette mit der letzten Monatssicherung in einem Quartal grundsätzlich aufheben, wenn möglich in einem anderen Gebäude.

- **Vor dem Sichern die Dateisysteme überprüfen.**

Eine Sicherung, die von einem beschädigten Dateisystem erstellt wurde, kann wertlos sein. Daher vor dem Sichern die Integrität des Dateisystems mit dem Befehl `fsck` überprüfen.

- **Sicherstellen, daß während der Sicherung keine Dateien bearbeitet werden.**

Beim Erstellen der Sicherungskopien sollte das System nicht für andere Arbeiten aktiviert sein, da sonst Dateien während der Sicherung geändert werden. Dadurch wäre die Sicherung fehlerhaft.

- **Das System sichern, bevor größere Änderungen vorgenommen werden.**

Ferner ist es immer ratsam, eine Sicherungskopie des gesamten Systems anzulegen, bevor die Hardware überprüft wird, Reparaturen ausgeführt werden oder neue Einheiten, Programme oder andere Zusatzeinrichtungen des Systems installiert werden.

Empfohlene Dokumentation

Zugehörige Informationen

Mit dem Befehl **backup** werden Dateisysteme gesichert.

Mit dem Befehl **cpio** werden Dateien in und aus Archivierungsspeichern und Verzeichnissen kopiert.

Mit dem Befehl **dd** werden Dateien umgesetzt und kopiert.

Mit dem Befehl **rdump** werden Dateien auf einer fernen Einheit gesichert.

Mit dem Befehl **restore** werden Dateisysteme zurückgeschrieben, die mit dem Befehl **backup** gesichert wurden.

Mit dem Befehl **rrestore** werden Dateisysteme zurückgespeichert, die mit dem Befehl **backup** auf einer fernen Einheit gesichert wurden.

Mit dem Befehl **tar** werden Archive bearbeitet.

Mit dem Befehl **pack** werden Dateien komprimiert.

Mit dem Befehl **unpack** werden komprimierte Dateien entpackt.

Mit dem Befehl **fsck** wird ein Dateisystem überprüft.

In der Datei **filesystems** im Handbuch *Files Reference* werden Dateisystemmerkmale festgelegt.

Im Abschnitt "Einführung in Verzeichnisse" auf Seite 3-8 befinden sich Informationen über das Arbeiten mit Verzeichnissen, verschiedene Arten von Verzeichnissen und die Verwendung von Pfadnamen.

Der Abschnitt "Überblick über Dateisysteme" auf Seite 2-2 enthält Informationen über die Arten von Dateisystemen und die Dateisystemverwaltung.

Dateien sichern

Achtung: Beim Sichern eines angehängten Dateisystems wird eine Nachricht angezeigt. Der Befehl **backup** setzt die Sicherung fort, es können jedoch Inkonsistenzen im Dateisystem auftreten. Diese Warnung ist für das Stammdateisystem (/) nicht relevant.

Der Befehl **backup** erstellt Sicherungskopien von Dateien auf einem Sicherungsdatenträger, z. B. Magnetband oder Diskette. Die erstellten Kopien weisen eines der beiden folgenden Backup-Formate auf:

- Wenn bestimmte Dateien nach Namen gesichert werden sollen, die Option **-i** verwenden.
- Wenn das gesamte Dateisystem nach der Dateiindexnummer gesichert werden soll, die Parameter **-Ebene** und **Dateisystem** verwenden.

Anmerkungen:

1. Wird eine Datei geändert, während ein vollständiges Dateisystem gesichert wird, besteht die Gefahr, daß Daten beschädigt werden. Aus diesem Grund sollte die Systemaktivität während des Sicherungsvorgangs so gering wie möglich sein.
2. Wird die Sicherung auf einem 8-mm-Band erstellt, und die Einheitenblockgröße ist auf 0 (Null) gesetzt, ist es nicht möglich, die Daten direkt vom Band zurückzuschreiben. In diese Fall können die Daten mit dem im Abschnitt zum Befehl **restore** beschriebenen speziellen Verfahren zurückgeschrieben werden.

Beispiele

1. Sichern bestimmter Dateien nach Namen:

```
find $HOME -print | backup -i -v
```

Mit der Option **-i** wird das System aufgefordert, die Namen der zu sichernden Dateien von der Standardeingabeeinheit zu lesen. Mit dem Befehl **find** wird eine Liste der Dateien im Benutzerverzeichnis **\$HOME** erstellt. Diese Liste wird über eine Pipe als Standardeingabe an den Befehl **backup** übergeben. Die Option **-v** zeigt einen Fortschrittsbericht nach jeder kopierten Datei an. Die Dateien werden auf der Standardsicherungseinheit für das lokale System gesichert.

2. Sichern des Stammdateisystems:

Achtung: Es sollte sichergestellt werden, daß die angegebenen Optionen mit dem Sicherungsdatenträger übereinstimmen. Wenn versucht wird, ein angehängtes Dateisystem zu sichern, können Inkonsistenzen auftreten.

```
backup -0 -u /
```

Die Angabe der Sicherungsstufe 0 und des Stammdateisystems (/) veranlassen das System, das gesamte Stammdateisystem (/) zu sichern. Die Sicherung erfolgt in der Datei **/dev/rfd0**. Die Option **-u** teilt dem System mit, die aktuelle Sicherungsstufe in der Datei **/etc/dumpdates** zu aktualisieren. Die Sicherung umfaßt nur das Stammdateisystem (/), nicht aber angehängte Systeme.

3. Sichern aller Dateien im Stammdateisystem (/), die seit der letzten Sicherung mit Sicherungsstufe 0 geändert wurden:

```
backup -1 -u /
```


Achtung: Es sollte sichergestellt werden, daß die angegebenen Optionen mit dem Sicherungsdatenträger übereinstimmen. Wenn versucht wird, ein angehängtes Dateisystem zu sichern, können Inkonsistenzen auftreten.

Empfohlene Dokumentation

Befehl und Dateien:

rdump, **find**, **dd** und **restore** im Handbuch *Commands Reference*.

dumpdates, **filesystems** und **rmt** im Handbuch *Files Reference*.

Überblick über Drucker und Warteschlangen

Drucker werden zur Druckausgabe eingesetzt. Abhängig vom Druckertyp können Aussehen und Merkmale des Endausdrucks gesteuert werden. Diese Drucker können in demselben Bereich wie Systemeinheit und -konsole oder an einem anderen Ort aufgestellt sein.

Damit Druckjobs mit maximaler Effektivität verarbeitet werden, stellt das System sie so lange in eine Warteschlange, bis der Drucker verfügbar ist. Das System kann die Ausgabe einer oder mehrerer Dateien in der Warteschlange sichern. Während der Drucker die Ausgabe einer Datei generiert, verarbeitet das System bereits den nächsten Job aus der Warteschlange. Die Druckjobs werden nacheinander abgearbeitet, bis keine Jobs mehr in der Warteschlange stehen. Nähere Informationen zu Druckern und Warteschlangen können den folgenden Abschnitten entnommen werden:

- Druckjob starten
- Priorität an einen lokalen Druckjob vergeben
- Status eines Druckjobs anzeigen
- Druckjob abrechnen.

Im Abschnitt "Printer Overview for System Management" im Handbuch *System Management Guide* werden die erforderlichen Konzepte zum Installieren und Verwalten von Druckern erklärt. Im Abschnitt "Queuing System Overview for System Management" im Handbuch *System Management Guide* werden die Systemverwaltungsaufgaben für Warteschlangen beschrieben.

Druckjob starten

Ein Druckjob ist eine auf einem Drucker auszuführende Arbeitseinheit. Abhängig von der Druckjobanforderung kann ein Druckjob eine oder mehrere Dateien enthalten. Druckjobs können durch Anforderung in SMIT oder durch Eingeben der Druckbefehle ausgeführt werden.

Das System ordnet jedem Job, den es ausführt, eine eindeutige Jobnummer zu.

Über SMIT oder durch Eingabe des Befehls `qprt` zur Anforderung eines Druckjobs kann der Benutzer folgende Angaben machen:

- Name der zu druckenden Datei
- Name der Druckwarteschlange
- Anzahl zu druckender Exemplare
- Kopie der Datei am fernen Host anlegen
- Datei nach dem Drucken löschen
- Hinweis über Jobstatus an Benutzer senden
- Hinweis über Jobstatus über die Systempost an Benutzer senden
- Trennstatus
- Benutzername für die Ausgabekennzeichnung "Schicken an"
- Konsolenrückmeldung bei fernem Drucken
- Dateirückmeldung bei fernem Drucken
- Prioritätsebene

SMIT bietet jeweils Hilfen zu den einzelnen Angaben und zeigt gegebenenfalls vorgegebene Standardwerte sowie vorhandene Optionen an.

Weitere Informationen enthält der Abschnitt "Starten eines Druckjobs" auf Seite 3-48.

Priorität an einen lokalen Druckjob vergeben

Beim Senden eines Jobs an den Drucker kann eine Priorität an diesen Job vergeben werden. Dies kann über ein SMIT-Menü oder durch den Befehl **qpri** geschehen. Jobprioritäten können nur für lokale Warteschlangen vergeben werden.

Höhere Werte bedeuten eine höhere Priorität für den jeweiligen Druckjob. Standardwert ist 15. Maximaler Wert ist für die meisten Benutzer 20, Benutzer mit Root-Berechtigung und Mitglieder der Gruppe `printq` (Gruppe 0) können maximal den Prioritätswert 30 vergeben.

Die Priorität bereits vergebener Druckjobs kann mit dem Befehl **qpri** neu zugeordnet werden. Nur Benutzer mit Root-Berechtigung und Mitglieder der Gruppe `printq` können Prioritäten an Jobs vergeben, die bereits in der Druckwarteschlange stehen.

Hinweis: Es ist zubeachten, daß Jobprioritäten nur an lokalen Warteschlangen vergeben werden können. Bei fernem Drucken ist dies nicht möglich.

Weitere Informationen enthält der Abschnitt "Prioritäten an einen Druckjob vergeben" auf Seite 3-52.

Status eines Druckjobs anzeigen

Es besteht die Möglichkeit, die aktuellen Statusinformationen für angegebene Jobnummern, Warteschlangen, Drucker oder Benutzer anzuzeigen. Dies kann über ein SMIT-Menü oder durch den Befehl **qchk** erfolgen.

Hinweis: Im Warteschlangensystem werden Mehrbyte-Host-Namen nicht unterstützt.

Weitere Informationen enthält der Abschnitt "Status eines Druckjobs überprüfen" auf Seite 3-51.

Druckjob abbrechen

Ein Druckjob, der in der Warteschlange steht, kann jederzeit abgebrochen werden. Dies kann über ein SMIT-Menü oder durch den Befehl **qcan** geschehen.

Beim Abbrechen eines Druckjobs müssen der Name der Warteschlange, in der der Job steht, und die Jobnummer des zu löschenden Jobs angegeben werden.

Starten eines Druckjobs

Voraussetzungen

1. Zur Durchführung von lokalen Druckjobs muß der Drucker physisch an das System angeschlossen sein.
2. Zur Durchführung von fernen Druckjobs muß das System so konfiguriert sein, daß es Daten zum fernen Druckserver übertragen kann.

Erforderliche Schritte

1. Den SMIT-Direktaufbefehl eingeben:

smit qprt

2. Die erforderlichen Einträge zu folgenden Menüpunkten machen:

Zu druckende DATEI

Name(n) der auf dem Drucker oder Plotter auszugebenden Datei(en). Es können mehrere Dateinamen eingegeben werden. Alle Dateien zusammen ergeben einen Druckjob. Die Dateien werden in der Reihenfolge ausgedruckt, in der sie vom Benutzer in das vorgesehene Feld eingegeben werden.

DRUCHWARTESCHLANGE:[Warteschlangeneinheit]

Name der Druckwarteschlange. Wenn kein Name eingegeben wird, wird die Standardwarteschlange verwendet. Mit der Funktion **Liste** kann eine Warteschlange ausgewählt werden.

ANZAHL an Kopien

Eine Zahl für die zu druckenden Exemplare pro Datei eingeben.

KOPIE der Datei auf lokalem Host erstellen?

Nur für lokale Druckjobs. Zum Kopieren einer Datei **ja** eingeben. Standardeinstellung ist **nein** (zum Drucken der Datei wird keine Kopie erstellt). Die zu druckende Datei wird stets für fernes Drucken kopiert.

Datei nach dem Drucken LÖSCHEN?

Zum Löschen der Datei nach dem Drucken **ja** eingeben. Standardoption ist **nein** (die Datei wird nach dem Drucken nicht gelöscht).

Benachrichtigung über den Jobstatus senden?

Die Standardeinstellung ist **nein** (nach Beendigung des Druckjobs wird kein Hinweis gesendet). Mit der Funktion **Liste** die gewünschte Option auswählen. Zum Anzeigen eines Hinweises über die Beendigung eines Druckjobs bei lokalen Druckjobs an **Bildschirm** oder **per Post** auswählen. Bei fernen Druckjobs erfolgt der Hinweis immer **per Post**, selbst wenn die Option an **Bildschirm** gewählt wird.

BURST-STATUS

Mit der Funktion **Liste** kann eine der folgenden Optionen ausgewählt werden: **keine Trennseiten** (Standardeinstellung), **nur Kopfseite**, **nur Abschlußseite**, **Kopfseite und Abschlußseite**. Die Option **nur Kopfseite** druckt ein **Deckblatt** vor jeder Datei im Druckjob aus. Die Option **nur Abschlußseite** druckt eine **Abschlußseite** nach jeder Datei im Druckjob aus. Die Option

Kopfseite und Abschlußseite druckt ein Deckblatt und eine Abschlußseite für jede Datei eines Druckjobs aus.

BENUTZERNAME für die Ausgabekennzeichnung 'Schicken an'
Diese Option druckt den angegebenen Benutzernamen auf dem Deckblatt hinter "SCHICKEN AN".

KONSOLEN-BESTÄTIGUNGSNACHRICHT für fernen Drucker
Eine Nachricht eingeben, die auf der Konsole angezeigt werden soll, wenn der Druckjob an einen Drucker übergeben wurde und mit dem Drucken begonnen werden kann. Der Druckjob wird erst fortgesetzt, nachdem die Nachricht auf der Konsole bestätigt wurde.

DATEI-BESTÄTIGUNGSNACHRICHT für fernen Drucker
Den Namen der Datei eingeben, die die auf der Konsole anzuzeigende Bestätigungsnachricht enthält.

PRIORITÄT Die Prioritätsebene ist eine dem Druckjob zugeordnete Zahl, die angibt, in welcher Reihenfolge der Druckjob ausgeführt wird. Die Prioritätsstufe kann jede ganze Zahl von 0 bis 30 sein. Der Standardwert ist 15. Die maximale Prioritätsstufe für Standardbenutzer ist 20. Die maximale Prioritätsstufe für Roots ist 30. Hohe Zahlenwerte bedeuten eine höhere Priorität. Die Priorität kann nur bei lokalen Druckjobs verändert werden.

Dieses Verfahren kann mit Hilfe des folgenden Befehls durchgeführt werden:

```
qprt Dateiname -P Warteschlangenname
```

Für die Anpassung eines Druckjobformats stehen noch weitere Optionen zur Verfügung.

Empfohlene Dokumentation

Die Abschnitte zu den Befehlen **lpr**, **enq**, **lp**.

Einen Druckjob abbrechen

Das folgende Verfahren gilt sowohl für lokale als auch für ferne Druckjobs.

Voraussetzungen

1. Zur Durchführung von lokalen Druckjobs muß der Drucker physisch an das System angeschlossen sein.
2. Zur Durchführung von fernem Druckjobs muß das System so konfiguriert sein, daß es Daten zum fernen Druckerserver übertragen kann.

Erforderliche Schritte

1. Den SMIT-Direktaufrufbefehl eingeben:

```
smit qcar
```

2. Einträge zu folgenden Attributen machen:

DRUCKWARTESCHLANGE,

in der der Job steht

Den Namen der Druckwarteschlange eingeben oder mit der Funktion **Liste** eine Druckwarteschlange auswählen.

JOBNUMMER Die Jobnummer des Druckjobs eingeben oder mit der Funktion **Liste** die Nummer des aktuellen Druckjobs auswählen.

Dieses Verfahren kann mit auch Hilfe des folgenden Befehls durchgeführt werden:

```
qcan -P Warteschlangename-x Jobnummer
```

Zum Anzeigen der Jobnummer den Befehl **qpri-A** eingeben. Zum Abbrechen eines Druckjobs müssen unter Umständen weitere Optionen verwendet werden.

Den Status eines Druckjobs prüfen

Das folgende Verfahren gilt sowohl für lokale als auch für ferne Druckjobs.

Voraussetzungen

1. Zur Durchführung von lokalen Druckjobs muß der Drucker physisch an das System angeschlossen sein.
2. Zur Durchführung von fernen Druckjobs muß das System so konfiguriert sein, daß es Daten zum fernen Druckerserver übertragen kann.

Erforderliche Schritte

1. Den SMIT-Direktaufrufbefehl eingeben:

```
smit qchk
```

2. Zum Anzeigen aller Jobs in allen Warteschlangen ja beim folgenden Menüpunkt eingeben:

Status ALLER Druckjobs anzeigen?

Mit der Funktion **Liste** ja oder nein auswählen. Wenn ja gewählt wird, in den übrigen Feldern keine Einträge mehr machen.

Dieses Verfahren kann auch mit Hilfe des folgenden Befehls durchgeführt werden:

```
qchk -A
```

ODER

Zum Anzeigen bestimmter Informationen die gewünschten Einträge in beliebiger Kombination in den folgenden Menüpunkten eingeben:

Name der WARTESCHLANGE

Mit der Funktion **Liste** eine Warteschlange auswählen.

JOBNUMBER Eine Druckjobnummer eingeben.

JOBEIGNER Den Namen eines Benutzers eingeben.

Dieses Verfahren kann auch mit Hilfe des folgenden Befehls durchgeführt werden:

```
qchk -P Warteschlangenname -#Jobnummer-u Eigername
```

Ist die Jobnummer eines Druckjobs nicht bekannt, den Befehl **qchk -A** verwenden.

Empfohlene Dokumentation

Zugehörige Informationen

Der Abschnitt "Printer Overview for System Management" im Handbuch *System Management Guide*.

Der Abschnitt "Queuing System Overview for System Management" im Handbuch *System Management Guide*.

Priorität an einen Druckjob vergeben

Das folgende Verfahren gilt sowohl für lokale als auch für ferne Druckjobs.

Voraussetzungen

1. Zur Durchführung von lokalen Druckjobs muß der Drucker physisch an das System angeschlossen sein.
2. Zur Durchführung von fernem Druckjobs muß das System so konfiguriert sein, daß es Daten zum fernen Druckerserver übertragen kann.

Erforderliche Schritte

1. Den SMIT-Direktaufbefehl eingeben:

```
smit qpri
```

2. Bei folgenden Attributen die gewünschten Einträge machen:

JOBNUMMER Mit der Funktion **Liste** den gewünschten Job aus der Liste der aktuellen Druckjobs auswählen.

Prioritäts-EBENE

Eine Zahl zwischen 0 und 20 eingeben (die größere Zahl gibt die höhere Priorität an). Der Standardwert ist 15. Roots können die Prioritätsebene bis auf 30 setzen.

- Einem Druckjob eine neue Priorität zuweisen:

```
qpri -# Jobnummer -a Prioritätsebene
```

Zum Anzeigen der Jobnummer den Befehl **qpri-A** eingeben.

- Einem lokalen Druckjob bei der Übergabe eine Priorität zuweisen:

```
qprt -P Warteschlangenname -R Prioritätsebene Dateiname
```

Umsetzen und Drucken von ASCII-Dateien auf einem PostScript-Drucker

Voraussetzungen

1. Der Drucker muß physisch an das System angeschlossen sein.
2. Der Drucker muß konfiguriert und definiert sein.
3. Der Aufzeichnungsteil des Textformatierungssystems muß installiert sein.

Erforderliche Schritte

Das Textformatierungssystem verfügt über einen Filter, der ASCII-Druckdateien in PostScript-Dateien umsetzt, damit sie auf einem PostScript-Drucker ausgedruckt werden können. Dieser Filter wird aufgerufen, wenn mit dem Befehl **qprt-da** ein Druckjob in eine PostScript-Druckwarteschlange gestellt wird. Wenn beispielsweise `meinDat.ascii` eine ASCII-Datei ist, wird bei Eingabe von

```
qprt -da -Pps meinDat.ascii
```

die Datei `meinDat.ascii` in eine PostScript-Datei umgesetzt und auf einem PostScript-Drucker, der für die Druckwarteschlange `ps` konfiguriert ist, ausgedruckt.

Automatische Umsetzung von ASCII-Dateien in PostScript-Dateien

Zahlreiche Anwendungen, die PostScript-Druckdateien generieren, setzen die Zeichen `%!` an den Anfang der Datei, um sie als PostScript-Druckdateien zu kennzeichnen. Folgende Schritte durchführen, um das System so zu konfigurieren, daß in eine PostScript-Warteschlange gestellte ASCII-Druckdateien gesucht und automatisch in PostScript-Dateien umgesetzt werden, bevor sie an den PostScript-Drucker geschickt werden:

1. Den SMIT-Direktaufrufbefehl eingeben:

```
smit lsvirprt
```
2. Den virtuellen PostScript-Drucker aus der angezeigten Liste auswählen.
3. Folgendes in der Befehlszeile eingeben:

```
d=a
```
4. SMIT verlassen.

Durch Eingabe von einem der folgenden Befehle wird nun eine ASCII-Datei in eine PostScript-Datei umgesetzt und auf einem PostScript-Drucker ausgedruckt:

```
qprt -Pps meinDat.ps meinDat.ascii
```

```
lpr -Pps meinDat.ps meinDat.ascii
```

```
lp -dps meinDat.ps meinDat.acsii
```

wobei `ps` eine PostScript-Druckwarteschlange ist.

Überschreiben der automatischen Druckdateikennung

In folgenden beiden Fällen muß die automatisch erstellte Kennung für PostScript-Druckdateien überschrieben werden:

1. Zum Drucken einer PostScript-Druckdatei, die nicht mit den Zeichen `%!` beginnt, folgendes in der Befehlszeile eingeben:

```
qprt -ds -Pps meinDat.ps
```

2. Zum Drucken der Quellenliste einer PostScript-Druckdatei, die nicht mit den Zeichen "%!" beginnt, folgendes in der Befehlszeile eingeben:

```
qprt -da -Pps myfile.ps
```

Umsetzungsoptionen

Folgende Optionen können mit dem Befehl **qprt** kombiniert werden, um beim Stellen von ASCII-Dateien in eine PostScript-Druckwarteschlange die Ausgabe anzupassen:

- 1+** Fügt Kopfzeilen ein.
- 2+** Formatiert die Ausgabe zweispaltig.
- 3+** Stellt Kopfzeilen, Datum und Seitenzahlen einheitlich dar. Wird manchmal auch als "hervorgehobener Modus" bezeichnet.
- 4+** Druckt auch Dateien, die nichtdarstellbare Zeichen enthalten.
- 5+** Listet Zeichen auf, die in einer Schriftart nicht enthalten sind.
- h-Zeichenfolge** Gibt eine Zeichenfolge an, die für die Kopfzeile verwendet wird. Wenn diese Option nicht angegeben wird, enthält die Kopfzeile Dateinamen, Änderungsdatum und Seitenzahl.
- l-Wert** Gibt die maximale Zeilenzahl pro Seite an. Abhängig von der Schriftgröße kann der Ausdruck weniger Zeilen pro Seite enthalten.
- L!** Schneidet Zeilen ab, die über die Zeilenbreite hinausgehen.
- p** Gibt die Schriftgröße an. Wenn diese Option nicht angegeben wird, wird von einer Schriftgröße von 10 Punkt ausgegangen, es sei denn, der zweispaltige, um 90 Grad gedrehte Modus (**-2+ -z1**) wird angegeben, bei dem der Wert 7 verwendet wird.
- s** Gibt den Schriftarttyp an. Gültige Angaben sind:
 - Courier-Oblique
 - Helvetica
 - Helvetica-Oblique
 - Helvetica-Narrow
 - Helvetica-Narrow-Oblique
 - NewCenturySchlbk kursiv
 - Optima
 - Optima-Oblique
 - Palatino-Roman
 - Palatino kursiv
 - Times-Roman
 - Times kursiv.Wenn diese Option nicht angegeben wird, wird die Schriftart Courier verwendet.

Hinweis: Die angegebenen Schriftarten müssen auf dem PostScript-Drucker verfügbar sein.

-z1

Dreht die Ausgabe um 90 Grad (Querformat).

Allgemeine Aufgaben

Die folgenden allgemeinen Aufgaben können mit Betriebssystembefehlen durchgeführt werden. Für jede der folgenden Aufgaben werden ein Befehl und ein Beispiel angeführt:

- Disketten formatieren
- Den Anmeldenamen anzeigen
- Eine Liste der angemeldeten Benutzer anzeigen
- Die Systemidentifikation eines angegebenen Benutzers anzeigen
- Den Systemnamen anzeigen
- Die aktuelle Umgebung anzeigen
- Das Kennwort ändern
- Zur Root-Berechtigung wechseln
- Eine Nachricht an einen anderen angemeldeten Benutzer senden
- Einen Kalender anzeigen
- Die Bildschirmanzeige löschen

Disketten formatieren

Achtung: Beim Formatieren werden sämtliche Daten auf einer Diskette gelöscht.

Die Befehle **format** und **fdformat** formatieren Disketten, die sich in dem im Parameter *Einheit* (standardmäßig **/dev/rfd0**) angegebenen Diskettenlaufwerk befinden. Der Befehl **format** erkennt die folgenden Einheitentypen:

- 5,25-Zoll-Diskette (360 KB), niedrige Speicherdichte, mit 40x2 Spuren mit jeweils neun Sektoren
- 5,25-Zoll-Diskette (1,2 MB), hohe Kapazität, mit 80x2 Spuren mit jeweils 15 Sektoren
- 3,5-Zoll-Diskette (720 KB), niedrige Speicherdichte, mit 80x2 Spuren mit jeweils neun Sektoren
- 3,5-Zoll-Diskette (2,88 MB), hohe Kapazität, mit 80x2 Spuren mit jeweils 36 Sektoren.

Bei allen Diskettentypen liegt die Sektorengröße bei 512 Byte.

Mit dem Befehl **format** wird eine Diskette (falls der Parameter *Einheit* keine andere Speicherdichte angibt) für hohe Speicherdichte formatiert.

Mit dem Befehl **fdformat** wird eine Diskette für niedrige Speicherdichte formatiert, es sei denn, die Option **-h** ist angegeben. Der Parameter *Einheit* gibt die Einheit an, die die zu formatierende Diskette enthält (die Einheit **/dev/rfd0** für das Laufwerk 0).

Vor dem Formatieren einer Diskette fordern die Befehle **format** und **fdformat** den Benutzer auf, die Formatieranforderung zu bestätigen. Dies stellt sicher, daß die Operation fehlerfrei durchgeführt werden kann.

Beispiele

1. Formatieren einer Diskette in der Einheit **/dev/rfd0**:

```
format -d /dev/rfd0
```

2. Formatieren einer Diskette ohne Überprüfung auf fehlerhafte Spuren:

```
format -f
```

3. Formatieren einer 360-KB-Diskette in einem 5,25-Zoll-Diskettenlaufwerk, das für 1,2 MB ausgelegt ist, in der Einheit **/dev/rfd1**:

```
format -l -d /dev/rfd1
```

4. Die Formatierung einer Diskette für hohe Speicherdichte bei Verwendung des Befehls **fdformat** erzwingen:

```
fdformat -h
```

Den Anmeldenamen anzeigen

Mit dem Befehl **whoami** wird der Anmelde-name angezeigt.

Beispiel

Den Anmeldenamen anzeigen:

```
whoami
```

Damit wird der Anmelde-name angezeigt; wird jedoch der Befehl `whoami` von einer virtuellen Datenstation aus ausgeführt, ist die Ausgabe dieselbe wie beim Befehl **who am i**, außer daß der Name der Datenstation und der Anmelde-name nicht angezeigt werden. Bei Eingabe des Befehls `who am i` oder `who am I` werden Anmelde-name, Name der Datenstation sowie Datum und Uhrzeit der Anmeldung angezeigt.

Liste der angemeldeten Benutzer anzeigen

Der Befehl **who** zeigt Informationen über alle Benutzer an, die gerade am lokalen System angemeldet sind. Die folgenden Informationen werden angezeigt: Anmelde-name, Name der Datenstation, Datum und Uhrzeit der Anmeldung.

Hinweis: Mit diesem Befehl werden nur Informationen über Benutzer angezeigt, die am lokalen Knoten angemeldet sind.

Beispiel

Informationen über Benutzer anzeigen, die am lokalen Systemknoten angemeldet sind:

```
who
```

Folgende oder ähnliche Informationen sind zu sehen:

```
joe hft/0 Jun 8 08:34
```

Die Systemidentifikation eines angegebenen Benutzers anzeigen

Der Befehl **id** schreibt eine Nachricht mit der Systemidentifikation (ID) eines angegebenen Benutzers auf die Standardausgabeeinheit. Bei den System-IDs handelt es sich um Zahlen, mit denen Benutzer und Benutzergruppen eindeutig im System gekennzeichnet sind. Folgende Informationen werden, falls vorhanden, vom Befehl **id** ausgegeben:

- Benutzername und reale Benutzer-ID
- Name der Benutzergruppe und reale Gruppen-ID
- Name zusätzlicher Benutzergruppen und zusätzlicher Gruppen-IDs.

Daten zu zusätzlichen Gruppen werden nur an Systemen ausgegeben, die aus mehreren Benutzern bestehende Gruppen unterstützen, und nur für Benutzer, die Mitglied einer zusätzlichen Gruppe sind.

Beispiel

Alle Systemidentifikationen eines aktuellen Benutzers anzeigen:

```
id
```

Die Ausgabe für den Befehl `id` erfolgt in folgendem Format:

```
uid=1544(sah) gid=300(build) euid=0(root) egid=9(printq)
groups=0(system),10(audit)
```

Der Name des Benutzers lautet `sah`, und die ID-Nummer ist 1544; der Name seiner Primärgruppe lautet `build`, und die ID-Nummer ist 300; der aktuelle Benutzername lautet `root`, und die aktuelle Benutzer-ID-Nummer ist 0; der Name seiner aktuellen Gruppe lautet `printq`, und die aktuelle ID-Nummer ist 9; die Namen zweier zusätzlicher Gruppen lauten `system` und `audit`, und die zusätzlichen ID-Nummern sind 0 bzw. 10.

Den Systemnamen anzeigen

Der Befehl `uname` schreibt den Namen des verwendeten Betriebssystems auf die Standardausgabeeinheit.

Die Einheiten-ID besteht aus 12 Zeichen, die im folgenden Format angeordnet sind: `xyyyyyymmss`. Beim System haben die Stellen `xx` immer 00. Die Stellen `yyyyyy` enthalten die eindeutige ID-Nummer für die Zentraleinheit. `mm` steht für die Modellnummer und `ss` für die Untermodellnummer (immer 00). Die Hexadezimalcodes für folgende Modelle lauten:

Modellnummer	Modell-ID
7008/M20	43
7008/M20A	43
7011/220	41
7011/220	45
7012/320	31
7012/320H	35
7012/340	37
7012/350	38
7012/355	77
7012/360	76
7012/365	76
7012/370	75
7012/375	75
7013/520	30
7013/520H	34
7013/530	10
7013/530H	18
7013/540	14
7013/550	1C
7013/560	5C
7013/570	67

7016/730	10
7015/930	20
7015/950	2E
7015/970	63
7015/970B	63
7015/980	64
7015/980B	64

Der vom Befehl **uname** ausgegebene Wert für die Einheitenkennung kann abweichen, wenn neuere BOS-Softwarestände installiert sind. Durch diese Abweichung werden Anwendungen beeinträchtigt, die diesen Wert beim Zugriff auf Lizenzprogramme verwenden. Mit dem Befehl **uname -m** wird diese Kennung angezeigt.

Bei Beeinträchtigung von Anwendungen an die Systemunterstützung wenden.

Beispiel

Den Namen des Betriebssystems anzeigen:

```
uname
```

Die aktuelle Umgebung anzeigen

Der Befehl **env** zeigt die aktuelle Umgebung an.

Beispiel

Die aktuelle Umgebung anzeigen:

```
env
```

Die aktuelle Umgebung wird in einer Liste mit einer *Name=Wert*-Folge pro Zeile angezeigt.

Das Kennwort ändern

Mit dem Befehl **passwd** wird das Kennwort geändert.

Beispiel

Folgendes eingeben, um das Kennwort zu ändern:

```
passwd
```

Das alte Kennwort (falls vorhanden) wird angefordert. Für das neue Kennwort erfolgt zweimal die Eingabeaufforderung.

Zur Superuserberechtigung wechseln

Der Befehl **su** ändert die Benutzerberechtigung in die Root-Berechtigung um.

Beispiel

Zur Root-Berechtigung wechseln:

```
su
```

Dieser Befehl führt eine Sub-Shell mit der aktuellen Benutzer-ID und Root-Berechtigung durch. Der Benutzer wird nach dem Root-Kennwort gefragt. Mit der Tastenfolge für Dateiende (Strg-D) wird die Sub-Shell verlassen und wieder zum Ausgangspunkt in der ursprünglichen Shell zurückgekehrt.

Eine Nachricht an einen anderen angemeldeten Benutzer senden

Mit dem Befehl **write** kann eine Nachricht ohne Verzögerung über das System geschickt werden. Auf diese Weise kann ähnlich wie bei einem Dialog mit einem anderen angemeldeten Benutzer kommuniziert werden. Jeder Benutzer sendet und empfängt abwechselnd Kurznachrichten von und zur anderen Datenstation. Lange Nachrichten können geschickt werden, indem die gesamte Nachricht in eine Datei gestellt und anschließend die Datei als Eingabe an den Befehl **write** umgeleitet wird.

Unmittelbar nach Eingabe des Befehls **write** wird die folgende Nachricht zusammen mit einem Signalton (dem ASCII BEL-Zeichen) an den Empfänger oder das Ziel der Nachricht:

```
Nachricht von Sender-ID an SenderHostname (tty $n$ ) [Datum] ...
```

Bei bestehender Verbindung sendet der Befehl **write** ein ASCII BEL-Zeichen an beide Datenstationen. Der Signalton teilt dem Sender mit, daß mit der Übertragung der Nachricht begonnen wird, und dem Empfänger, daß eine Nachricht ankommt.

Mit dem Befehl **write** kann auch mit Benutzern auf anderen Hosts kommuniziert werden, wenn der Benutzer auf einem fernen Host mit der Option **-n Hostname** oder dem Parameter **Benutzer@Host** angegeben wird. Für das Schicken von Nachrichten an einen Benutzer auf einem fernen Host muß der Dämon **writesrv** sowohl auf dem aktuellen Host als auch auf dem fernen Host aktiv sein.

Beispiele

1. Eine Nachricht an einen anderen angemeldeten Benutzer senden:

```
write june
```

Eingabetaste drücken und Text eingeben

Ich muß dich sehen! Komm um 12:30 in den Computerraum.

Die Tastenfolge Strg-D verwenden, um den Befehlsmodus **write** zu beenden.

Wenn die Benutzer-ID *karen* ist, und der Benutzer auf der Datenstation *tty3* arbeitet, wird auf der Anzeige von *june's* Datenstation folgendes angezeigt:

```
Message from karen on trek tty3 Aug 17 11: 55:24 ...  
Ich muß dich sehen. Komm um 12:30 in den Computerraum.
```

```
<EOT>
```

2. Einen Dialog führen:

```
write june
```

Eingabetaste drücken und Text eingeben

Komm um 12:30 in den Computerraum.

o

Damit wird der Dialog begonnen. Das o am Anfang der folgenden Zeile bedeutet, daß die Nachricht beendet ist. Es teilt June mit, daß der Benutzer auf eine Antwort wartet. Wenn der Dialog fortgesetzt werden soll, darf die Tastenfolge Strg-D nicht gedrückt werden.

June antwortet durch Eingabe von:

```
write karen
```

June drückt die Eingabetaste und gibt Text ein

Ich habe um 12:30 eine Besprechung. Wie wäre es mit 3?

o

Die Antwort könnte lauten:

```
OK--Um 3 im Computerraum.  
oo
```

Das oo bedeutet "over and out" und teilt June mit, daß der Benutzer keine weiteren Nachrichten hat. Wenn June ebenfalls keine Nachricht mehr hat (oo), drücken beide Strg-D, um den Dialog zu beenden.

3. Eine vorbereitete Nachricht verschicken:

```
write june < message.text
```

Damit wird der Inhalt der Datei **message.text** an june's Datenstation geschickt.

4. Eine Nachricht an einen Benutzer an einer bestimmtem Datenstation senden:

```
write console
```

Eingabetaste drücken und Text eingeben

```
Der Drucker in Gebäude 998 ist defekt.  
Bitte schicken Sie Hilfe.
```

Anschließend die Tastenfolge Strg-D verwenden.

Damit wird die Nachricht an den Benutzer geschickt, der an der Datenstation /dev/console arbeitet.

5. Eine Nachricht an den Benutzer spuds am Host partya senden:

```
write -n partya spuds
```

Eingabetaste drücken und Text eingeben

```
Ihr neues Band ist soeben angekommen,  
Bitte hohlen Sie es bei mir ab.  
Danke!
```

Anschließend die Tastenfolge Strg-D verwenden.

oder

```
write spuds@party
```

Eingabetaste drücken und Text eingeben

```
Ihr neues Band ist soeben angekommen,  
Bitte hohlen Sie es bei mir ab.  
Danke!
```

Anschließend die Tastenfolge Strg-D verwenden.

Einen Kalender anzeigen

Der Befehl **cal** schreibt einen Kalender für das angegebene Jahr oder den angegebenen Monat auf die Standardausgabereinheit.

Der Parameter *Monat* bezeichnet den Monat, für den ein Kalender gewünscht wird. Zulässige Werte sind 1 bis 12 (für Januar bis Dezember). Wird der Parameter *Monat* nicht angegeben, geht der Befehl **cal** standardmäßig vom aktuellen Monat aus.

Der Parameter *Jahr* bezeichnet das Jahr, für das ein Kalender gewünscht wird. Da mit dem Befehl **cal** für jedes Jahr von 1 bis 9999 ein Kalender angezeigt werden kann, ist stets die vollständige Jahreszahl anzugeben. Wird der Parameter *Jahr* nicht angegeben, geht der Befehl **cal** standardmäßig vom aktuellen Jahr aus.

Beispiele

1. Einen Kalender für Februar 1984 an der Datenstation anzeigen:

```
cal 2 1984
```

2. Einen Kalender für 1984 ausdrucken:

```
cal 1984 | qprt
```

3. Einen Kalender für das Jahr 84 n. Chr. anzeigen:

```
cal 84
```

Den Bildschirm löschen

Der Befehl **clear** löscht die Anzeige, sofern möglich. Der Befehl **clear** überprüft zuerst die Umgebungsvariable **TERM** für den Datenstationstyp. Als nächstes überprüft er das Verzeichnis **/usr/share/lib/terminfo**, das die Datenstationsdefinitionsdateien enthält, um festzustellen, wie die Anzeige gelöscht werden kann. Wenn die Umgebungsvariable **TERM** nicht gesetzt ist, wird der Befehl **clear** beendet, ohne daß ein Verarbeitungsvorgang durchgeführt wird.

Beispiel

Bildschirm löschen:

```
clear
```

Kapitel 4. AIXwindows Window Manager

Dieser Abschnitt bietet einen Überblick über den Zusammenhang zwischen dem X-Window-System, dem AIXwindows Window Manager und der Arbeitsoberfläche (Desktop). Darüber hinaus enthält der Abschnitt Informationen und Prozeduren zum AIXwindows Window Manager.

Der AIXwindows Window Manager bietet Werkzeuge für die Steuerung von Fenstern, den definierten Arbeitsbereichen auf der Anzeige, in denen Programme Informationen darstellen und Eingaben akzeptieren. Viele für den Computer verfügbare Anwendungen basieren auf dem Window Manager, so daß die bei der Benutzung des Window Manager erworbenen Kenntnisse bei der Arbeit mit verschiedenen Anwendungen eingesetzt werden können.

Übersicht über den AIXwindows Window Manager

In diesem Abschnitt werden folgende Themen behandelt:

- AIXwindows Window Manager und AIXwindows Desktop
- Einführung in AIXwindows
- Benutzung des AIXwindows Window Manager
- Interaktion mit AIXwindows-Anwendungen

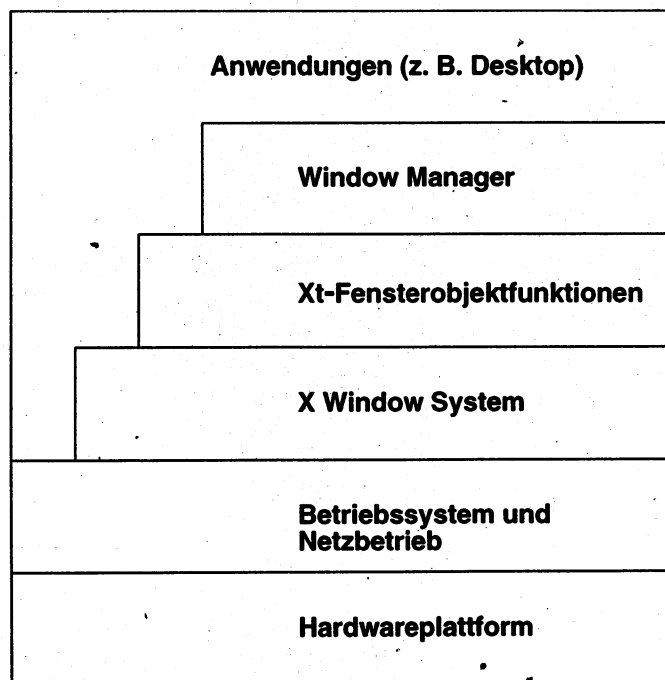
AIXwindows Window Manager and AIXwindows Desktop

AIXwindows Window Manager und AIXwindows Desktop stellen zwei Schnittstellen zwischen Benutzer und Computer dar. Grafische Arbeitsplätze haben die Interaktion zwischen Benutzern und Datenverarbeitungssystemen verändert. Datenstationen mit integrierten Grafikfunktionen haben grafische Benutzerschnittstellen wie z. B. AIXwindows Desktop ermöglicht. Mit AIXwindows Desktop können Befehle zum Ausführen von Programmen, Editieren, Kopieren und Löschen von Dateien über die Maus ausgeführt werden und müssen nicht über die Tastatur eingegeben werden. AIXwindows Desktop teilt außerdem die physische Bildschirmfläche in Bereiche, sogenannte *Fenster*, auf. In diesen wird die Ausgabe verschiedener Anwendungen angezeigt. Grafische Schnittstellen bieten eine einfache Möglichkeit zum Arbeiten mit Datenstationen.

Die grafische Benutzerschnittstelle AIXwindows Desktop arbeitet nicht eigenständig. Sie ist abhängig vom AIXwindows Window Manager, welcher wiederum vom X Window System abhängig ist.

Der Window Manager stellt Hilfsprogramme für die Fenster, die definierten Arbeitsbereiche auf der Anzeige, zur Verfügung, in denen Programme Daten ausgeben und Eingaben vorgenommen werden können. Viele für den Computer verfügbare Programme basieren auf dem Window Manager, so daß die Kenntnisse, die beim Arbeiten mit dem Window Manager erworben werden, auch beim Arbeiten mit diversen anderen Anwendungen verwendet werden können.

Folgende Abbildung zeigt die Beziehung zwischen verschiedenen Komponenten des Systems.



Entwicklungsmodell der Benutzerschnittstelle

Der folgende Abschnitt enthält einen Überblick über die besondere Beziehung zwischen dem X Window System, dem AIXwindows Window Manager und dem Desktop.

X Window System

Das X Window System bietet die Möglichkeit, hierarchische Fensterstrukturen zu erstellen und in Fenstern zu zeichnen. Das System erkennt, ob eine Taste oder eine Maustaste gedrückt wurde. Xwindows ist keine grafische Benutzerschnittstelle. Es ist ein vernetztes Fenstersystem. Die tatsächliche Interaktion mit dem Benutzer erfolgt über die grafische Benutzerschnittstelle AIXwindows Desktop, die auf dem X Window System basiert und vom AIXwindows Window Manager verwaltet wird.

Vorteile von Windows

- Das Ausführen mehrerer Prozesse ist möglich.
- Jedem Prozeß ist ein eigenes Fenster zugeordnet.
- Alle Fenster können gleichzeitig angezeigt werden.
- Größe und Position der Fenster kann nach deren Bedeutung festgelegt werden.
- Ein Fenster kann auf einem fernen System ausgeführt werden.

Klienten-Server-Modell

Das X Window System arbeitet auf der Basis eines Klienten-Server-Modells. Das heißt, der Server steuert die zur Ein-/Ausgabe verwendete Hardware und die Klienten senden Anforderungen an den Server. Dadurch können Klientenprogramme hardwareunabhängig arbeiten.

Anzeigen-Server

In der Terminologie von Xwindows wird der Anzeigen-Server als Server oder als Anzeige bezeichnet. Die physische Hardware wird als Bildschirm bezeichnet.

Klient

Klienten sind Anwendungsprogramme, die Anforderungen an den Anzeigen-Server richten. AIXwindows Window Manager und AIXwindows Desktop sind Beispiele für Klienten.

Grafische Benutzerschnittstelle

Die Benutzerschnittstelle einer Anwendung bestimmt deren Erscheinungsbild und ihre Funktionalität. Verwendet eine Benutzerschnittstelle Grafikobjekte, wie z. B. Fenster und Menüs, wird sie als grafische Benutzerschnittstelle bezeichnet.

Das grafische *Fenstersystem*, das Teil der grafischen Benutzerschnittstelle ist, ordnet die grafische Ausgabe auf der Anzeige und führt grundlegende Text- und Grafikzeichenfunktionen aus.

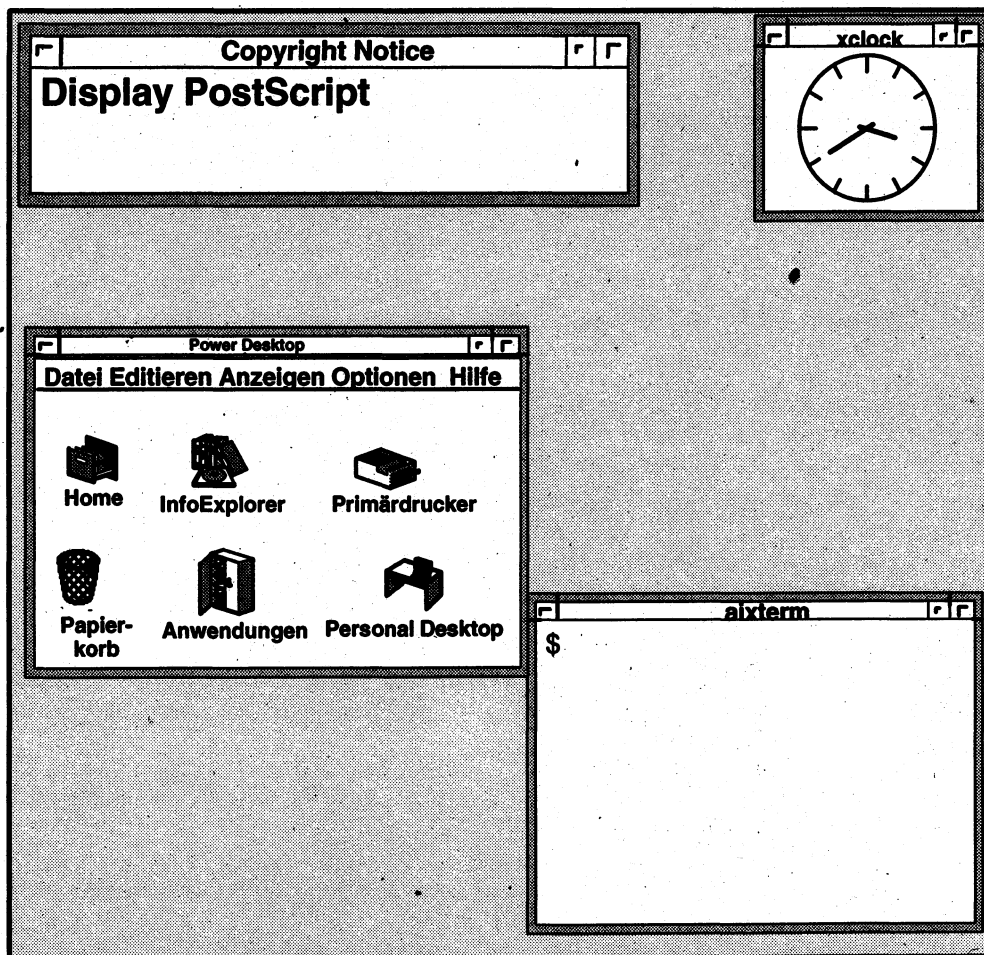
Der *Window Manager*, ein weiterer Teil der grafischen Benutzerschnittstelle, ermöglicht dem Benutzer das Positionieren und das Verändern der Größe von Fenstern. Außerdem versieht er die Fenster mit einem Rahmen. Der AIXwindows Window Manager steuert auch den *Eingabefokus*, den Mechanismus, durch den ein Benutzer eines von mehreren Fenstern auf der Anzeige auswählen und zum aktiven Fenster machen kann.

AIXwindows Window Manager und AIXwindows Desktop

Der AIXwindows Window Manager ist ein besonderer Xwindows-Klient, der die Interaktion zwischen Fenstern verschiedener Klienten auf der Anzeige steuert. Der Window Manager steuert die Position und die Größe aller Fenster auf der Anzeige. Ohne den Window Manager kann die Größe oder die Position eines Fensters nicht geändert werden.

Das AIXwindows Desktop ist ein weiterer Xwindows-Klient, der auch vom AIXwindows Window Manager abhängig ist. Die Arbeitsoberfläche stellt eine einfache, effektive Möglichkeit zum Arbeiten mit der Datenstation dar. Das visuelle System von AIXwindows hilft sowohl Benutzern ohne Vorkenntnisse als auch erfahrenen Benutzern beim Verwalten ihrer Arbeit und Ausführen verschiedener Aufgaben mit dem Betriebssystem. AIXwindows Window Manager ist die zugrundeliegende Software, die die Grafikausgabe auf der Anzeige steuert.

Folgende Abbildung zeigt die Arbeitsoberfläche und den Window Manager.



Die Arbeitsoberfläche und der Window Manager

Einführung in AIXwindows

AIXwindows ist ein Softwaresystem, das vielfältige Steuerungsmöglichkeiten für das Erscheinungsbild der visuellen Anzeige des Computers bietet.

Dieser Artikel enthält Informationen zu folgenden Themen:

- X Window System und AIXwindows auf Seite 4-5
- Einführung in das Fensterkonzept auf Seite 4-5
- Funktionsweise von AIXwindows auf Seite 4-7
- X und AIXwindows Window Manager starten und verlassen auf Seite 4-9
- Standardeinstellungen wiederherstellen auf Seite 4-9

X Window System und AIXwindows

Durch größere Bildschirme, insbesondere bei Datenstationen im Konstruktionsbereich, können heute im Vergleich zu früher mehr Informationen auf dem Bildschirm angezeigt werden. Zum Verwalten dieser Informationen wird die Anzeige in mehrere rechteckige Bereiche, sogenannte *Fenster*, unterteilt. Das *X Window System* (allgemein mit *X* bezeichnet) wurde im Massachusetts Institute of Technology entwickelt, um als Industriestandard für die Anzeige von Fenstern mit Text und Grafik zu dienen. Das *X Window System* ist so konzipiert, daß es unabhängig von der verwendeten Computerhardware ist.

Das *X Window System* besteht aus einem *X Server*, durch den die visuelle Anzeige verwaltet wird, und *Klientenanwendungsprogrammen*. Klientenanwendungsprogramme können eine Reihe von Tasks ausführen, wie z. B. die Verarbeitung elektronischer Post, die Verwaltung einer Datenbank oder einfach die Anzeige der aktuellen Uhrzeit. Jede Anwendung erscheint in einem eigenen Fenster oder in einer Familie zugeordneter Fenster. Der Server übermittelt die Benutzereingaben, wie z. B. das Klicken mit der Maus oder einen Tastenanschlag, an die entsprechende Klientenanwendung. Klientenanwendungen übertragen ihre Anforderungen von Anzeigeaktionen auf den Server. Die *X Server*- und Klientenanwendungen können sich auf demselben Computer oder auf unterschiedlichen Computern in einem Netz befinden.

Das *X Window System* stellt dem Programmierer ein allgemeines System zur Verfügung, das eine beliebige Anzahl verschiedener Benutzerschnittstellen unterstützt, ohne genau anzugeben, wie Anwendungen erscheinen oder sich verhalten sollten. Ohne zusätzliche Programmierungserweiterungen sind Fenster in *X* einfache Rechtecke mit Rahmen. AIXwindows implementiert ein flexibles Softwaresystem, das auf *X* aufsetzt, um bestimmte Anzeigekomponenten wie Schiebeleisten und Menüs zu erstellen. Programmierer können durch Kombination dieser Komponenten Benutzerschnittstellen erstellen, die die Interaktion mit Anwendungsprogrammen erleichtern.

Einführung in das Fensterkonzept

Die Verwaltung von Fenstern auf der Anzeige wird häufig mit der Verwaltung von Blättern auf einem Schreibtisch verglichen. Blätter können auf einem Schreibtisch beliebig angeordnet werden, sie können z. B. übereinandergestapelt werden, so daß ein Blatt die anderen Blätter ganz oder teilweise verdeckt. Die Blätter können auch abgelegt oder entfernt werden, oder es können neue Blätter zu der Blattsammlung hinzugefügt werden.

Anders als bei Blättern auf einem Schreibtisch können der Inhalt, die Größe und andere Attribute von Fenstern geändert werden. Fenster können über die Anzeige hinausgehen, so

daß Teile des Fensters nicht sichtbar sind. Die Anzahl und der Inhalt von Fenstern auf der Anzeige ist abhängig von den jeweils ausgeführten Anwendungen.

Fensterhierarchie

Fenster werden in einer Hierarchie verwaltet. Die gesamte Anzeige wird (analog zu einem Schreibtisch) von einem *Basisfenster* ausgefüllt, das nicht entfernt werden kann. Die anderen Fenster erscheinen innerhalb des Basisfensters und sind "Kinder" dieses Fensters. Jedes Fenster außer dem Basisfenster besitzt ein Elternfenster. Fenster, die dasselbe Elternfenster besitzen, werden als Geschwisterfenster betrachtet.

In AIXwindows-Anwendungen gibt es drei grundlegende Arten von Fenstern:

- Primärfenster
- Sekundärfenster
- Menüfenster

Primärfenster verwalten den Inhalt der Anwendung. Primärfenster dienen als Elternfenster für andere Fenster in einer Anwendung. Sie sind normalerweise Kinder des Basisfensters.

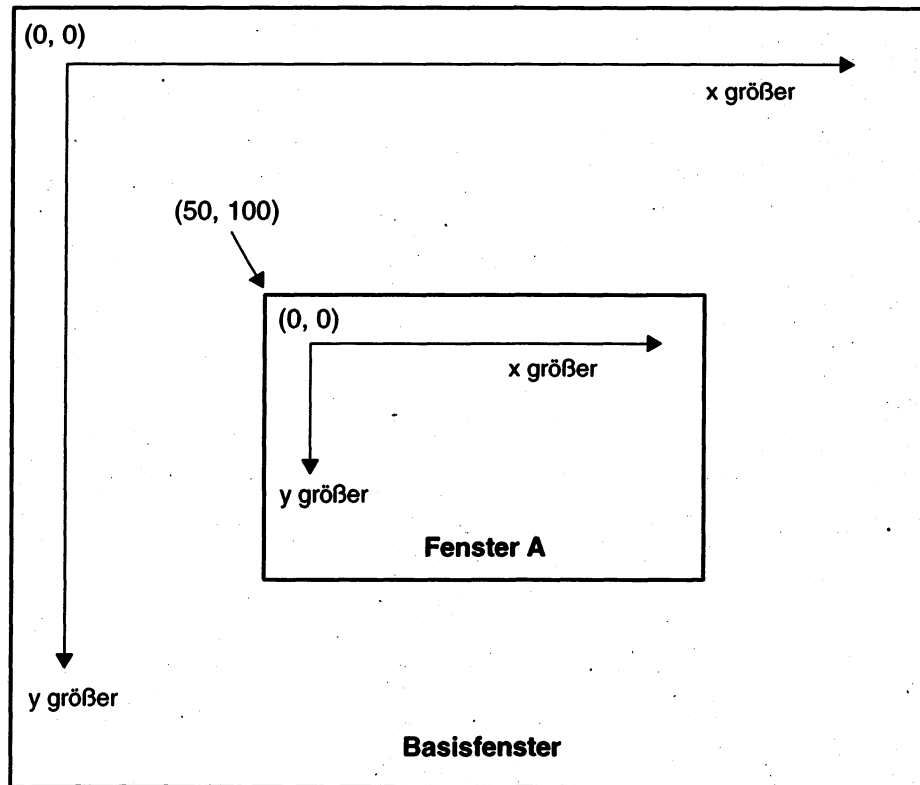
AIXwindows-Anwendungen verwenden *Sekundärfenster* für die Übermittlung von Informationen an den Benutzer oder die Anforderung von Informationen von einem Benutzer. Ein Sekundärfenster besitzt immer ein Elternfenster; das Elternfenster kann ein Primärfenster oder ein weiteres Sekundärfenster sein.

Menüfenster (oder einfach Menüs) zeigen temporäre Listen von Auswahlmöglichkeiten an. Ein Menüfenster ist immer einem Elternfenster zugeordnet (ein Primärfenster, ein Sekundärfenster oder ein anderes Menü).

Fenstergeometrie

Die Position und Größe eines Fensters auf einer Anzeige wird als *Fenstergeometrie* bezeichnet. Höhe und Breite eines Fensters werden normalerweise in *Pixeln* (Bildpunkte) angegeben. Eine Anwendung kann jedoch Einheiten verwenden, die in dieser Anwendung sinnvoll sind, wie z. B. die Anzahl Zeichen oder Zeilen.

Wie die Punkte eines Diagramms besitzt jedes Fenster seine eigenen Koordinaten. Die horizontale Position des Fensters wird durch x , die vertikale Position durch y angegeben. Die Koordinatenposition x, y der linken oberen Ecke jedes Fensters (Ursprung) ist $(0, 0)$. Die x -Koordinate wird größer, wenn sich der Benutzer innerhalb des Fensters nach rechts bewegt, die y -Koordinate wird größer, wenn sich der Benutzer innerhalb des Fensters nach unten bewegt. Die folgende Abbildung zeigt das System der Fensterkoordinaten.



Fensterkoordinaten

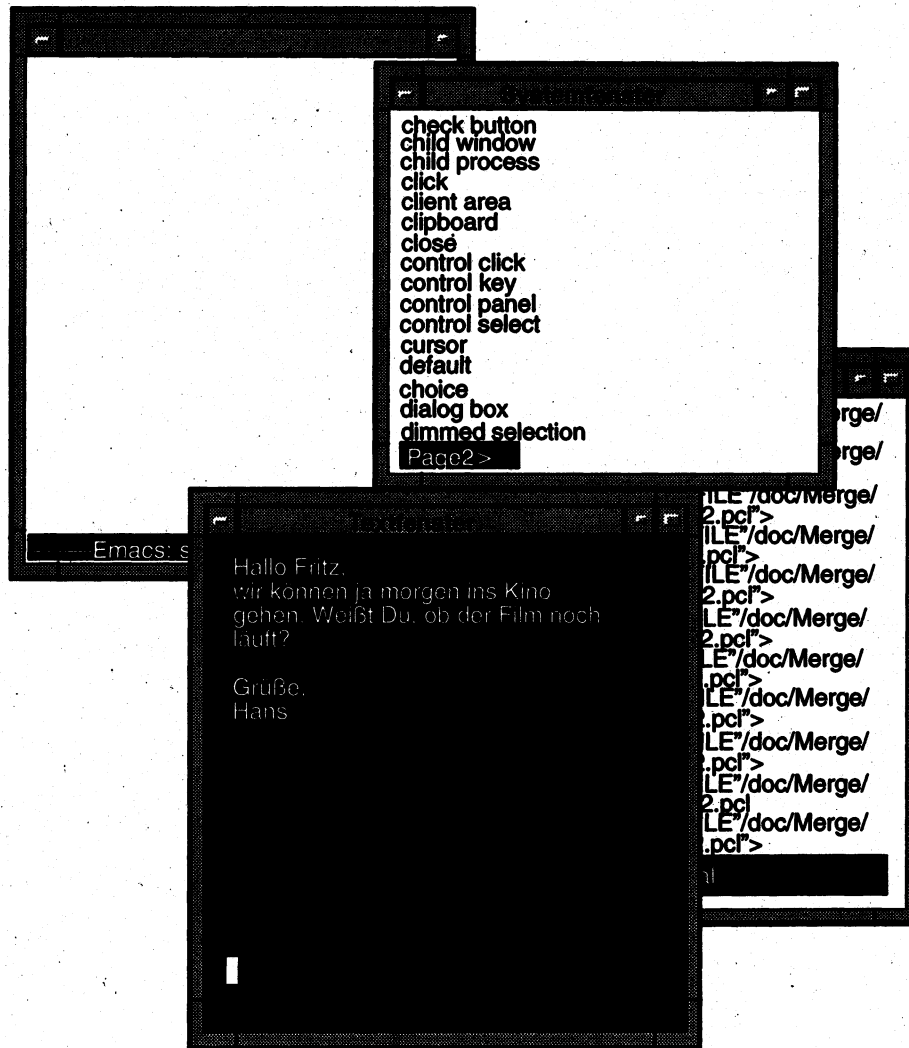
Relativ zum Basisfenster befindet sich der Ursprung von Fenster A (linke obere Ecke) bei der Koordinate (50, 100). Relativ zu Fenster A ist dies jedoch die Koordinate (0, 0). Unter *Customizing AIXwindows Window Manager Overview in AIXwindows and AIXwindows Desktop User's Guide* wird beschrieben, wie die Koordinaten für Fenster in AIXwindows-Anwendungen angegeben werden.

Interaktion mit Fenstern

Mit Hilfe eines Klientenprogramms mit dem Namen *Window Manager* hat der Benutzer die Möglichkeit, Fenster zu verschieben, von der Anzeige zu entfernen, zu vergrößern oder zu verkleinern oder die Stapelordnung zu verändern. AIXwindows besitzt einen eigenen Window Manager, den AIXwindows Window Manager (MWM). Die Interaktion mit Klientenanwendungsprogrammen (einschließlich Window Manager) kann über eine Zeigereinheit (z. B. eine Maus) oder die Tastatur erfolgen. Es gibt Tastenanschläge, die Maustastenaktionen entsprechen. Unter Benutzung des AIXwindows Window Manager auf Seite 4-11 wird MWM ausführlich beschrieben.

Funktionsweise von AIXwindows

Das deutlichste Unterscheidungsmerkmal von AIXwindows ist die dreidimensionale Darstellung. Die dreidimensionale Darstellung wird durch Schatteneffekte erreicht, durch die die Ränder der visuellen Komponenten von AIXwindows abgeschrägt erscheinen. Die folgende Abbildung zeigt eine typische Anzeige von AIXwindows.



Neben der dreidimensionalen Darstellung zeichnet sich AIXwindows durch eine Vielzahl von Benutzerschnittstellenkomponenten, sogenannte Fensterobjekte, aus. AIXwindows stellt Funktionalität für folgende Anzeigekomponenten bereit:

- Knöpfe
- Menüs
- Dialogfenster
- Teilfenster und Blätterfenster
- Listen
- Text

Eine Beschreibung dieser Komponenten und der Interaktion mit ihnen befindet sich unter Interaktion mit AIXwindows-Anwendungen auf Seite 4-21.

AIXwindows bietet die Möglichkeit, das Erscheinungsbild und die Verhaltensweise von AIXwindows-Anwendungen zu ändern. Der Benutzer hat zum Beispiel die Möglichkeit, die Farben und Schriftarten der eigenen Klientenanwendungen zu ändern, Menüs zum AIXwindows Window Manager hinzuzufügen und das Anfangsfensterlayout beim Starten auszuwählen.

Diese Dokumentation erläutert AIXwindows-Anwendungen (einschließlich MWM) in bezug auf ihre Verträglichkeit mit den Richtlinien im AIXwindows Style Guide Overview in *User Interface Programming Concepts Volume 1*. Dieser Ansatz ermöglicht dem Benutzer einen problemlosen Einstieg in das Arbeiten mit beliebigen Anwendungen, die mit dem Style Guide in Einklang stehen.

X und AIXwindows Window Manager starten und verlassen

Die Vorgehensweise zum Starten von X und AIXwindows ist vom jeweiligen Datenverarbeitungssystem abhängig. Aus diesem Grund sollte beim Systemadministrator nachgefragt werden, wie zum Starten vorzugehen ist. Normalerweise werden X und MWM über eine Shell-Prozedur gestartet, die nach dem Anmelden automatisch ausgeführt wird. Es besteht aber auch die Möglichkeit, X oder MWM oder beides zu starten.

Wird nach dem Anmelden festgestellt, daß die Anzeige als einzelne nichtprogrammierbare Datenstation ohne Anzeige von Fenstern arbeitet, kann X durch Absetzen des folgenden Befehls gestartet werden:

% **xinit**

Wird X durch diesen Befehl nicht gestartet, zusammen mit dem Systemadministrator überprüfen, ob das Verzeichnis X11, das die ausführbaren Programme enthält, im Suchpfad enthalten ist. Der richtige Pfad kann von System zu System unterschiedlich sein.

Werden nach dem Anmelden ein oder mehrere Fenster ohne Rahmen angezeigt, kann MWM mit folgendem Befehl gestartet werden:

% **mwm &**

Vor Eingabe dieses Befehls sicherstellen, daß der Zeiger sich in einem Fenster mit einer Eingabeaufforderung befindet.

Zum Verlassen von AIXwindows die Tastenfolge Alt-Strg-Rückschrittaste verwenden.

Hinweis: Bevor AIXwindows verlassen wird, sollten alle Anwendungsprogramme verlassen und alle Befehle, die in Datenstationsfenstern ausgeführt werden, gestoppt werden. Dadurch wird ein möglicher Verlust von Daten vermieden, der durch nicht korrektes Stoppen von Programmen verursacht werden kann.

Nach dem Verlassen eines Programms wird die Befehlszeileneingabeaufforderung wieder im Datenstationsfenster angezeigt. Wurde das Programm jedoch automatisch oder über ein Menü gestartet, wird bei Verlassen des Programms das Datenstationsfenster ebenfalls entfernt. Durch Auswahl der Option **Schließen** in einem Fenstermenü wird sofort jedes Programm gestoppt, das in dem Fenster ausgeführt wird. Das Unterbrechen von Programmen auf diese Weise kann zu Datenverlust führen. Die Uhr, "Histogramm laden" sowie "leere" Datenstationsfenster (Datenstationsfenster, in denen nur die Befehlszeileneingabeaufforderung angezeigt wird) können jedoch geschlossen werden, ohne daß Daten verlorengehen.

Standardeinstellungen wiederherstellen

Da AIXwindows eine Vielzahl von Anpassungsmöglichkeiten sowohl für Programmierer, die AIXwindows-Anwendungen schreiben, als auch für Benutzer bietet, kann es sein, daß die Maustasten und andere Funktionen anders funktionieren als in dieser Dokumentation angegeben. Die AIXwindows-Umgebung kann daher durch gleichzeitiges Drücken der folgenden Tasten auf die Standardeinstellungen zurückgesetzt werden:

Alt-Strg-Umschalttaste-!

Durch erneutes Verwenden dieser Tastenkombination kann die angepaßte Einstellung erneut aktiviert werden. Erlaubt das System nicht diese Kombination von Tastenanschlägen

können die Standardeinstellungen auch über das Standardstammenü wiederhergestellt werden.

Benutzung des AIXwindows Window Manager

Mit Hilfe des AIXwindows Window Manager (MWM) kann der Benutzer Fenster verschieben, ihre Größe ändern, Fenster in Symbole umwandeln und neue Fenster erstellen. Durch Editieren von Konfigurationsdateien können das Erscheinungsbild und die Verhaltensweise von MWM wie gewünscht angepaßt werden. Unter Customizing the AIXwindows Window Manager im *AIXwindows and AIXwindows Desktop User's Guide* befinden sich weitere Informationen zur Anpassung von MWM.

Dieser Artikel beschreibt die standardmäßige Verhaltensweise von MWM und bietet Informationen zu folgenden Themen:

- Interaktion mit MWM auf Seite 4-11
- Zugriff auf Fensterverwaltungsfunktionen auf Seite 4-12
- Fenster für die Eingabe auswählen und Fenster in den Vordergrund stellen auf Seite 4-13
- Benutzung des MWM-Fenstermenüs auf Seite 4-15
- Fenster verschieben auf Seite 4-16
- Größe von Fenstern ändern auf Seite 4-17
- Fenster auf maximale Größe vergrößern auf Seite 4-17
- Benutzung von Symbolen auf Seite 4-18
- Fenster schließen auf Seite 4-19
- Benutzung des Standardstammenüs auf Seite 4-19

Interaktion mit MWM

Die Interaktion mit MWM kann über eine Zeigereinheit (normalerweise eine Maus) und/oder über verschiedene Tastenkombinationen auf der Tastatur erfolgen. MWM verwendet ein Objektaktionsmodell der Interaktion. Der Benutzer wählt ein Fenster, ein Menü, ein Symbol oder ein anderes Grafikobjekt aus und gibt eine Aktion für dieses Objekt an.

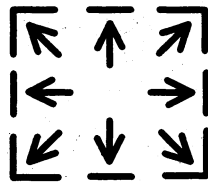
Ein Grafikabbild auf der Anzeige (*Zeiger*) verfolgt die Bewegung der Maus auf dem Schreibtisch. Beim Starten von X hat der Zeiger die Form eines x. (Diese Form ist die Standardeinstellung; die Zeigerform kann angepaßt werden.) Beim Bewegen der Maus kann sich das Abbild des Zeigers verändern. Die Zeigerform ist so konzipiert, daß sie visuell die gerade ausgeführte Aktion darstellt. Der Zeiger kann zum Beispiel in einem Datenstationsfenster oder Texteditor wie ein I-Balken aussehen. Die folgende Abbildung zeigt die Zeigerformen.



Zeigen



Ausführen einer
Aktion



Ändern der
Größe



Verschieben

**Zeigerformen zeigen
Aktivitäten visuell an**

Benutzung der Maus

Mit einer Maus können folgende Aktionen ausgeführt werden:

- Zeigen** Bewegen der Maus auf der Arbeitsoberfläche, um den Zeiger auf ein Objekt auf der Anzeige zu plazieren.
- Drücken** Niederhalten einer Maustaste.
- Klicken** Drücken und Freigeben einer Maustaste, ohne den Zeiger zu bewegen.
- Doppelklicken** Eine Maustaste zweimal in kurzer Abfolge drücken.
- Ziehen** Drücken und Niederhalten einer Maustaste, während die Maus bewegt wird.

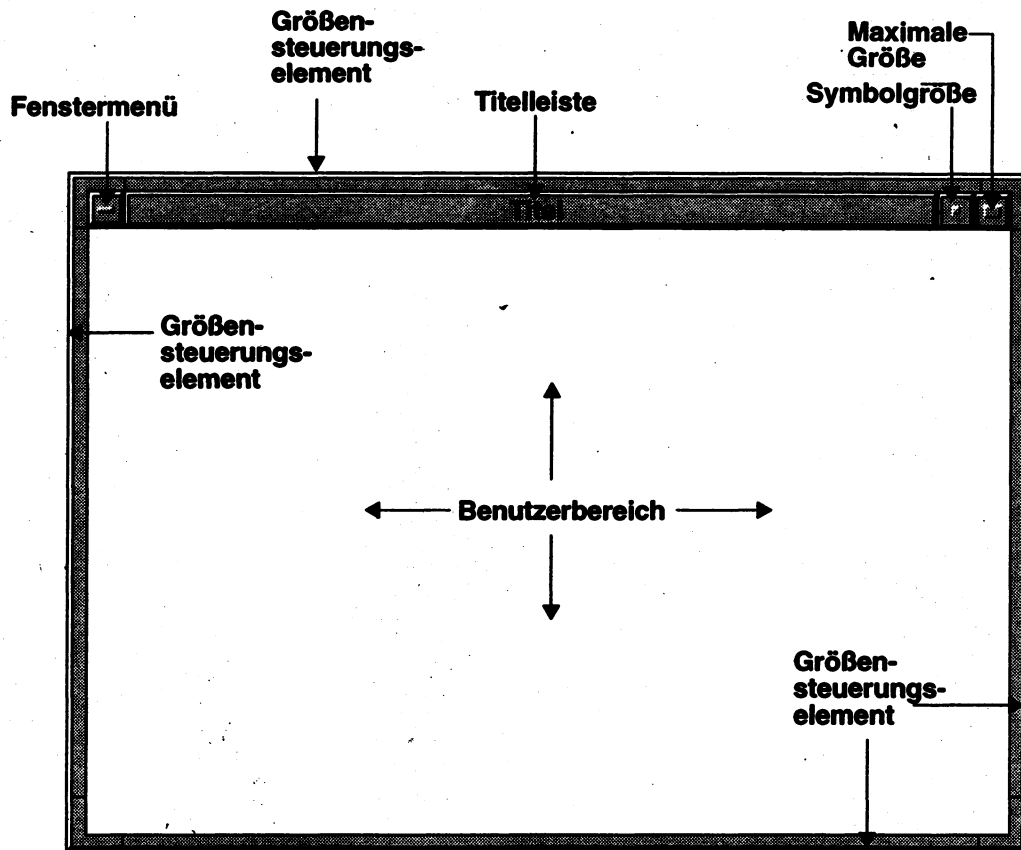
Benutzung der Tastatur

Viele Fensterverwaltungsfunktionen können ohne eine Maus ausgeführt werden. Zur Ausführung von Fensterverwaltungsfunktionen über die Tastatur sind häufig Kombinationstasten erforderlich. Durch Drücken dieser Tasten werden die Aktionen, die normalerweise anderen Tasten zugeordnet sind, verändert. Da AIXwindows auf vielen verschiedenen Datenstationstypen ausgeführt wird, sind die Tastenbezeichnungen von System zu System unterschiedlich.

Alle MWM-Tastaturaktionen entsprechen der Verhaltensweise, die im AIXwindows Style Guide in *User Interface Programming Concepts Volume 1* angegeben ist. Zum Beispiel wird bei gleichzeitigem Drücken von Umschalttaste und Abbruchtaste das Fenstermenü für das aktive Fenster angezeigt. Mit Hilfe der Pfeiltasten kann in den Menüpunkten geblättert werden. Durch Drücken der Rückföhrtaste wird ein Menüpunkt ausgewählt. Bei Drücken der Abbruchtaste wird die aktuelle Fensterverwaltungsoperation abgebrochen.

Zugriff auf Fensterverwaltungsfunktionen

Der Zugriff auf die meisten Fensterverwaltungsfunktionen wird über Elemente des MWM-Fensterrahmens ermöglicht. Die folgende Abbildung zeigt ein typisches Fensterlayout.



Typisches Fensterlayout

In der folgenden Liste sind die Elemente des Fensterrahmens und ihre Funktionen aufgeführt:

Element des Rahmens	Funktion
Titelbereich	Fenster verschieben.
Fenstermenüknopf	Fenstermenü anzeigen.
Knopf für Symbolgröße	Fenster auf Symbolgröße verkleinern.
Knopf für maximale Größe	Fenster auf maximale Größe vergrößern.
Größensteuerungselemente	Fenster horizontal, vertikal oder diagonal (in zwei Richtungen) vergrößern oder verkleinern.

Fenster für Eingabe auswählen

Bevor ein Fenster Eingaben empfangen kann, muß es als *Eingabebereich* ausgewählt werden. Es kann nur jeweils ein Fenster ausgewählt sein. Dieses Fenster wird als *aktives Fenster* bezeichnet. Wenn ein Fenster aktiv ist, ändert sich die Farbe oder die Schattierung des Fensterrahmens. Farbe und Schattierung für aktive und inaktive Fenster können vom Benutzer ausgewählt werden. Unter Customizing the AIXwindows Window Manager im *AIXwindows and AIXwindows Desktop User's Guide* befinden sich Informationen über die Auswahl von Farben und Schattierungen für aktive und inaktive Fenster.

Das *Eingabebereichsverfahren* legt fest, wie ein Fenster ausgewählt wird. Der Window Manager unterstützt sowohl das *explizite Eingabebereichsverfahren* als auch das *implizite Eingabebereichsverfahren*. Das implizite Eingabebereichsverfahren wird auch als *Zeigereingabebereichsverfahren* bezeichnet. Standardmäßig wird das explizite

Eingabebereichsverfahren unterstützt; das bedeutet, daß der Benutzer mit der linken Maustaste auf den Fensterrahmen klicken oder die Tastenfolge Alt-Tabulatortaste verwenden muß, um den Eingabebereich von einem Fenster auf ein anderes zu übertragen.

Dieses Zeigereingabebereichsverfahren veranlaßt MWM, den Eingabebereich entsprechend der Zeigerposition zu ändern. Es sind keine weiteren Aktionen erforderlich. Das Fenster, in dem sich der Zeiger befindet, besitzt daher immer den Eingabebereich. Dieses Eingabebereichsverfahren kann angegeben werden, indem die Standardwerte in der Konfigurationsdatei von X geändert werden. Unter Customizing the AIXwindows Window Manager im *AIXwindows and AIXwindows Desktop User's Guide* befinden sich nähere Informationen zum Festlegen des Eingabebereichsverfahrens.

Am einfachsten kann ein Fenster für die Eingabe ausgewählt werden, indem die Maus so bewegt wird, daß der Zeiger innerhalb des Rahmens des zu verwendenden Fensters steht. Ist ein explizites Eingabebereichsverfahren wirksam, muß ebenfalls mit der linken Maustaste geklickt werden. Beim expliziten Eingabebereichsverfahren empfängt nur das für den *Tastatureingabebereich* ausgewählte Fenster Tastatureingaben, unabhängig davon, wo sich der Zeiger befindet. Bei Verwendung des expliziten Eingabebereichsverfahrens kann der Eingabebereich auch mit Hilfe der Tastatur verschoben werden. Durch die Tastenfolge Alt-Tabulatortaste wird der Eingabebereich zum nächsten Fenster im Fensterstapel bewegt. Durch die Tastenfolge Umsch.-Alt-Tabulatortaste wird der Eingabebereich auf das vorherige Fenster übertragen.

Fenster in den Vordergrund stellen

Soll ein teilweise verdecktes Fenster an den Anfang des Fensterstapels geholt werden, den Zeiger auf den Rahmen des Fensters stellen und mit der linken Maustaste klicken. Darauf achten, daß sich der Zeiger nicht auf einem der Befehlsknöpfe (Symbolgröße, maximale Größe oder Fenstermenü) des Fensterrahmens befindet.

Der Window Manager stellt auch Standardmethoden zur Verfügung, mit deren Hilfe ein Fenster über die Tastatur in den Vordergrund gestellt werden kann. Diese sind nachfolgend aufgeführt:

Tastenfolge	Funktion
Alt-Tabulatortaste	Stellt das nächste Fenster im Stapel in den Vordergrund und überträgt den Eingabebereich auf dieses Fenster (nur expliziter Eingabebereich).
Alt-Umsch.-Tabulatortaste	Stellt das vorherige Fenster im Stapel in den Vordergrund und überträgt den Eingabebereich auf dieses Fenster (nur expliziter Eingabebereich).
Alt-Abbruchtaste	Stellt das nächste Fenster im Stapel in den Vordergrund, ohne den Eingabebereich zu ändern.
Alt-Umsch.-Abbruchtaste	Stellt das vorherige Fenster im Stapel in den Vordergrund, ohne den Eingabebereich zu ändern.

Wird bei Verwendung eines expliziten Eingabebereichsverfahrens ein Fenster in den Vordergrund gestellt, wird der Eingabebereich ebenfalls auf dieses Fenster übertragen. Bei Verwendung eines Zeigereingabebereichsverfahrens setzt der Window Manager den Eingabebereich jedoch nicht automatisch auf das im Vordergrund befindliche Fenster. Das heißt, ein Fenster kann weiterhin Eingaben empfangen, auch wenn es sich nicht am Anfang des Stapels befindet. Eine Ressource in der Konfigurationsdatei von X kann jedoch auch so festgelegt werden, daß ein Fenster automatisch an den Anfang des Stapels gestellt wird, wenn der Eingabebereich auf dieses Fenster gesetzt wird. Unter Customizing the AIXwindows Window Manager im *AIXwindows and AIXwindows Desktop User's Guide*

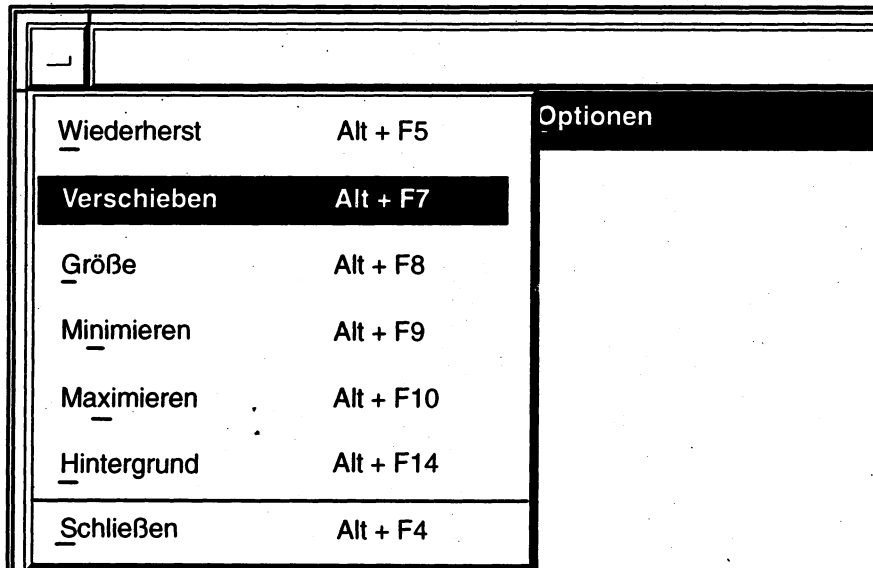
befinden sich Informationen darüber, wie eine Ressource in der Konfigurationsdatei von X festgelegt wird, damit eine Datei automatisch an den Anfang des Stapels verschoben wird, wenn der Eingabebereich auf dieses Fenster gesetzt wird.

Benutzung des MWM-Fenstermenüs

Das *Fenstermenü* stellt eine alternative Möglichkeit für den Zugriff auf die Funktionalität des Window Manager dar. Das Fenstermenü kann auf zwei Arten verwendet werden:

- Den Zeiger auf den Fenstermenüknopf in der linken oberen Ecke des Fensterrahmens stellen und die linke Maustaste drücken. Soll das Menü geöffnet bleiben, mit der linken Maustaste klicken. Das Fenstermenü kann auch geöffnet werden, indem der Zeiger auf eine beliebige Stelle des Fensterrahmens gestellt und die rechte Maustaste gedrückt wird. Wenn das Menü angezeigt wird, kann durch Ziehen der Maus in den Menüpunkten geblättert werden. Wenn der Zeiger eine verfügbare Auswahl berührt, wird die Auswahl hervorgehoben. Zur Auswahl eines Menüpunkts die Maustaste freigeben, wenn die gewünschte Auswahl hervorgehoben wird.
- Den Tastatureingabebereich auf ein Fenster setzen und die Tastenfolge Umschalttaste-Abbruchtaste oder Alt-Leertaste verwenden. Dadurch wird das Fenstermenü angezeigt oder *geöffnet*. Mit Hilfe der Aufwärts- und Abwärtspfeiltasten kann in den Menüpunkten geblättert werden. Zur Auswahl des hervorgehobenen Menüpunkts die Rückföhrtaste, die Auswahlstaste oder die Leertaste drücken. Durch Drücken der Abbruchtaste kann das geöffnete Menü entfernt werden.

Die folgende Abbildung zeigt ein geöffnetes Fenstermenü.



Fenstermenüknopf mit Aktionsfenstermenü

In der folgenden Liste sind die standardmäßigen Auswahlmöglichkeiten des Fenstermenüs und die zugehörigen Funktionen aufgeführt:

Auswahlmöglichkeit	Funktion
Wiederherst	Ein Fenster in Symbolgröße oder maximaler Größe wird in der Originalgröße wiederhergestellt.
Verschieben	Die Position eines Fensters wird geändert.
Größe	Die Größe eines Fensters wird geändert.

Minimieren	Ein Fenster wird auf Symbolgröße verkleinert.
Maximieren	Ein Fenster wird auf die maximale Größe vergrößert.
Hintergrund	Ein Fenster wird an das Ende eines Fensterstapels gestellt.
Schließen	Die Ausführung eines Programms in einem Fenster wird gestoppt und das Fenster geschlossen.

Menüpunkte für Funktionen, die momentan nicht ausführbar sind, werden abgeblendet angezeigt; das Auswählen dieser Funktionen hat keine Auswirkung.

Fenstermenüpunkte können auch über die Tastatur ausgewählt werden. *Direktaufrufe* sind Tastenbelegungen, die einen schnellen Zugriff auf Menüfunktionen ohne Öffnen des Menüs ermöglichen. Zum Beispiel wird mit der Tastenfolge Alt-F3 das Fenster in den Hintergrund gestellt. Direktaufrufe werden im Menü rechts neben dem entsprechenden Menüpunkt angezeigt.

Durch *mnemonische Zeichen* kann ein Menüpunkt nach dem Öffnen des Fenstermenüs ausgewählt werden. Mnemonische Zeichen sind in einem Menüpunkt durch ein unterstrichenes Zeichen dargestellt. Zum Beispiel hat das Drücken der Taste V bei geöffnetem Fenstermenü die gleiche Wirkung wie die Auswahl von **Verschieben** mit der Maus.

Fenster verschieben

Ein Fenster kann verschoben werden, indem der Zeiger in den Titelbereich des Fensters gestellt, die linke Maustaste gedrückt und festgehalten und die Maus gezogen wird. Der Zeiger ändert normalerweise während des Ziehvorgangs seine Form. Beim Bewegen der Maus zeigt standardmäßig ein Umriss die aktuelle Position des Fensters an. Eine Ressource kann so festgelegt werden, daß bei der Verschiebeoperation das Fenster selbst und nicht der Umriss angezeigt wird. Unter Customizing the AIXwindows Window Manager im *AIXwindows and AIXwindows Desktop User's Guide* befinden sich Informationen darüber, wie eine Ressource festgelegt wird, damit bei der Verschiebeoperation statt einem Umriss das Fenster angezeigt wird. Außerdem wird ein Rückmeldungsfenster angezeigt, das die aktuelle Position des Fensters in Bildpunkten relativ zum Ursprung des Basisfensters anzeigt.

Befindet sich der Umriss an der gewünschten Position, die Maustaste freigeben. Das Fenster wird an diese Position verschoben. Das Verschieben kann durch Drücken der Abbruchtaste abgebrochen werden.

Ein Fenster kann auch mit Hilfe der Tastatur verschoben werden. Hierzu zunächst die Menüoption **Verschieben** aus dem Fenstermenü auswählen oder den Direktaufruf Alt-F7 verwenden. Mit Hilfe der Pfeiltasten wird das Fenster in Richtung des Pfeils verschoben. Wird die Steuerungstaste beim Drücken der Pfeiltasten gedrückt gehalten, wird das Fenster in größeren Sprüngen verschoben (siehe die folgende Liste). Das Fenster kann auch mit Hilfe der Maus verschoben werden, nachdem die Verschiebeoperation über die Tastatur eingeleitet wurde. Zum Beenden der Verschiebeoperation die Rückföhrtaste drücken.

Taste	Funktion
< ↑ >	Ein Fenster wird um einen Bildpunkt nach oben verschoben.
< ↓ >	Ein Fenster wird um einen Bildpunkt nach unten verschoben.
< ← >	Ein Fenster wird um einen Bildpunkt nach links verschoben.
< → >	Ein Fenster wird um einen Bildpunkt nach rechts verschoben.

Strg-<Pfeiltaste>
Rückföhrtaste

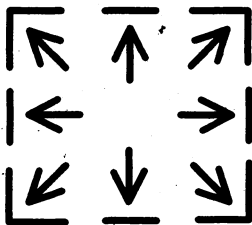
Ein Fenster wird um mehrere Bildpunkte verschoben.
Die interaktive Positionierung wird beendet.

Fenstergröße ändern

Die Größe eines Fensters kann geändert werden, indem der Zeiger auf folgende Größensteuerungselemente des Fensterrahmens gestellt wird:

- Oberer Rand
- Unterer Rand
- Linke Seite
- Rechte Seite
- Untere rechte Ecke
- Untere linke Ecke
- Obere rechte Ecke
- Obere linke Ecke

Die folgende Abbildung zeigt die Cursorform jedes Größensteuerungselements.



Cursor für
Größenänderung

Richtung und Ergebnis des Größenänderungsvorgangs hängen vom ausgewählten Größensteuerungselement ab. Zum Auswählen des Fensters die linke Maustaste drücken, wenn sich der Mauszeiger auf einem der Größensteuerungselemente befindet. Durch Bewegen der Maus das Fenster vergrößern oder verkleinern. Ein Umriß zeigt die veränderten Abmessungen. Außerdem zeigt ein kleines Rückmeldungsfenster die aktuellen Abmessungen des Rahmens an. Die linke Maustaste freigeben, um die Größenänderungsoperation zu beenden. Die Operation kann jederzeit durch Drücken der Abbruchtaste abgebrochen werden.

Die Größe eines Fensters kann auch über die Tastatur geändert werden. Hierzu zunächst die Menüoption **Größe** aus dem Fenstermenü auswählen oder den Direktaufruf Alt-F8 verwenden. Zum Ändern der Größe des Fensters die Pfeiltaste drücken, die in Richtung der zu ändernden Abmessung zeigt. Anschließend durch Drücken derselben bzw. der entgegengesetzten Pfeiltaste das Fenster vergrößern bzw. verkleinern. Die Rückföhrtaste drücken, wenn das Fenster die gewünschte Größe besitzt.

Fenster auf maximale Größe vergrößern

Beim *Maximieren* eines Fensters wird das Fenster auf die größtmögliche Größe vergrößert. Die größtmögliche Größe eines Fensters kann durch das Anwendungsprogramm oder durch konfigurierte Ressourcen begrenzt werden.

Soll ein Fenster auf maximale Größe vergrößert werden, den Zeiger auf den Knopf für maximale Größe in der rechten Ecke des Fensterrahmens stellen und mit der linken Maustaste klicken. Das Klientfenster wird auf die durch das Anwendungsprogramm oder

die konfigurierten Ressourcen festgelegte maximale Größe vergrößert. Standardmäßig füllt das Fenster in maximaler Größe die Anzeige aus.

Wenn das Fenster auf maximale Größe vergrößert ist, erscheint das Quadrat auf dem Knopf für maximale Größe, als wäre es in die Anzeige eingelassen. Durch Klicken mit der linken Maustaste auf den Knopf für maximale Größe wird das Fenster wieder in der vorherigen Größe und an der vorherigen Position angezeigt.

Ein Fenster kann auch auf maximale Größe vergrößert werden; indem die Option **Maximieren** im Fenstermenü ausgewählt oder die Tastenfolge Alt-F10 verwendet wird. Ein Fenster in maximaler Größe kann wiederhergestellt werden, indem **Wiederherst** im Fenstermenü ausgewählt oder der Direktaufruf Alt-F5 verwendet wird.

Benutzung von Symbolen

Symbole sind kleine grafische Darstellungen von Fenstern, die dazu dienen, die Übersichtlichkeit der Anzeige zu verbessern. Ein Klientfenster kann jederzeit in ein Symbol oder ein Symbol wieder in ein Fenster umgewandelt werden.

Ein typisches Symbol besteht aus zwei Komponenten: einem Abbild und einer Marke. Die folgende Abbildung zeigt ein typisches Symbol.



Typisches Symbol

Standardmäßig ordnet der Window Manager Symbole am unteren Rand der Anzeige von links nach rechts an. Symbole können auch in ein *Symbolfeld* gestellt werden. Das Symbolfeld ist ein MWM-Fenster, das Symbole enthält. Das Symbolfeld kann ebenfalls auf Symbolgröße verkleinert (d. h. in ein Symbol umgewandelt) werden. Die Erstellung eines Symbolfelds kann angegeben werden, indem ein Wert in der Konfigurationsdatei von X angegeben wird. Unter *Advanced AIXwindows Window Manager Customization* im *AIXwindows and AIXwindows Desktop User's Guide* befinden sich Informationen darüber, wie ein Symbol erstellt wird, indem ein Wert in der Konfigurationsdatei von X gesetzt wird.

Ein Symbol besitzt ebenfalls ein Menü, das dieselben Auswahlmöglichkeiten wie das Fenstermenü des zugehörigen Klientfensters enthält. Zum Anzeigen des Menüs für das Symbol mit Hilfe der Maus den Zeiger auf das Symbol stellen und die rechte Maustaste drücken. Zum Anzeigen des Menüs für das Symbol über die Tastatur die Tastenfolgen Umschalttaste-F10 oder Alt-Leertaste verwenden. Innerhalb des Menüs kann durch Ziehen der Maus oder durch Drücken der Pfeiltasten geblättert werden. Zum Auswählen eines Menüpunktes bei hervorgehobenem Menüpunkt die Maustaste freigeben oder die Rückföhrtaste drücken. Nicht verfügbare Menüpunkte sind abgeblendet dargestellt. Wenn das Menü geöffnet ist, kann es abgebrochen werden, indem die Abbruchtaste gedrückt oder mit der rechten Maustaste außerhalb des Menüs geklickt wird.

Fenster in ein Symbol umwandeln

Zum Umwandeln eines Fensters in ein Symbol den Zeiger auf den Knopf für Symbolgröße am Fensterrahmen stellen und mit der linken Maustaste klicken. Der Knopf für Symbolgröße befindet sich links neben dem Knopf für maximale Größe. Ein Fenster kann auch auf

Symbolgröße verkleinert werden, indem **Minimieren** im Fenstermenü ausgewählt oder der Direktaufruf Alt-F9 bei ausgewähltem Fenster verwendet wird.

Symbole verschieben

Zum Verschieben eines Symbols den Zeiger auf das Symbol stellen, die linke Maustaste drücken und gedrückt halten und die Maus bewegen. Das Symbol erscheint als Umriß. Befindet sich der Umriß an der gewünschten Position, die Maustaste freigeben.

Der Verschiebevorgang kann auch eingeleitet werden, indem **Verschieben** aus dem Menü für das Symbol ausgewählt oder der Direktaufruf Alt-F7 verwendet wird. Die Maus ziehen, bis sich der Umriß an der Stelle befindet, an die das Symbol verschoben werden soll. Zum Beenden der Verschiebeoperation mit der linken Maustaste klicken oder die Rückföhrtaste drücken. Im expliziten Eingabebereichsmodus ist der Eingabebereich nach dem Verschieben auf das Symbol gesetzt.

Auf Symbolgröße verkleinertes Fenster wiederherstellen

Zum Wiederherstellen eines auf Symbolgröße verkleinerten Fensters den Mauszeiger auf das Symbol stellen und mit der linken Maustaste doppelt klicken. Daraufhin wird das Fenster mit der vorherigen Größe und Position wiederhergestellt. Ein auf Symbolgröße verkleinertes Fenster kann auch wiederhergestellt werden, indem **Wiederherst** im Fenstermenü für das Symbol ausgewählt oder der Direktaufruf Alt-F5 verwendet wird.

Fenster schließen

Durch Schließen eines Fensters wird dieses von der Anzeige entfernt. Zum Schließen eines Fensters die Option **Schließen** aus dem Fenstermenü auswählen oder den Direktaufruf Alt-F4 bei ausgewähltem Fenster verwenden. Ist das Fenster das Hauptfenster einer laufenden Anwendung, wird die Anwendung normalerweise abgebrochen. Einige Anwendungen können nicht direkt durch Schließen des Hauptfensters abgebrochen werden. Bei solchen Anwendungen müssen zum Abbrechen bei Schließen des Hauptfensters weitere Schritte ausgeführt werden. Verwendet eine Anwendung mehrere unabhängige Primärfenster, wird durch Auswahl von **Schließen** entweder nur dieses Fenster geschlossen oder die Anwendung verlassen.

Benutzung des Standardstammenüs

Das Stammenü ermöglicht den Zugriff auf MWM-Funktionen, die sich auf die gesamte Anzeige beziehen. Zum Beispiel kann das Stammenü Auswahlmöglichkeiten zum Erstellen eines neuen Fensters oder zum Neustarten von MWM enthalten. Einträge für das Stammenü können in der Konfigurationsdatei von MWM angegeben werden. Unter *Advanced AIXwindows Window Manager Customization* im *AIXwindows and AIXwindows Desktop User's Guide* wird beschrieben, wie Einträge für das Stammenü in der Konfigurationsdatei von MWM angegeben werden.

Zum Anzeigen des Stammenüs den Zeiger auf eine beliebige Stelle des Stammenüs stellen und die rechte Maustaste drücken. Die folgende Abbildung zeigt das Standardstammenü.

Stammenü
Neues Fenster
Vordergrund
Hintergrund
Aktualisieren
Symbole neu anordnen
Neu starten . . .

Standardstammenü

In der folgenden Liste sind die Auswahlmöglichkeiten des Standardstammenüs und die zugehörigen Funktionen aufgeführt:

Auswahlmöglichkeit	Funktion
Neues Fenster	Ein neues Datenstationsemulationsfenster wird auf der Anzeige geöffnet.
Vordergrund	Das Fenster am Ende des Fensterstapels wird in den Vordergrund gestellt.
Hintergrund	Das Fenster am Anfang des Fensterstapels wird in den Hintergrund gestellt.
Aktualisieren	Der Inhalt der gesamten Anzeige wird aktualisiert.
Symbole neu anordnen	Symbole werden nebeneinander ausgerichtet.
Neu starten	MWM wird gestoppt und neu gestartet.

Neu starten wird normalerweise ausgewählt, wenn MWM-Ressourcen geändert wurden und die Änderungen aktiviert werden sollen. Unter Using Customization Features im *AIXwindows and AIXwindows Desktop User's Guide* befinden sich Informationen zur Auswahl von **Neu starten** (Restart), wenn MWM-Ressourcen geändert wurden und die Änderungen aktiviert werden sollen. Bei Auswahl von **Neu starten** wird der Benutzer gefragt, ob MWM tatsächlich neu gestartet werden soll. Den Zeiger auf OK setzen und mit der linken Maustaste klicken, wenn MWM neu gestartet werden soll; andernfalls den Zeiger auf Abbrechen setzen und mit der linken Maustaste klicken, wenn der Vorgang abgebrochen werden soll.

Interaktion mit AIXwindows-Anwendungen

AIXwindows enthalten verschiedene visuelle Komponenten, die die Interaktion mit dem Programm ermöglichen. Trotz der Unterschiede zwischen AIXwindows-Anwendungen können bestimmte Verhaltensweisen in allen AIXwindows-Anwendungen erwartet werden. Diese Dokumentation beschreibt einige der Komponenten, die wahrscheinlich in Anwendungen zu finden sind, die mit den im AIXwindowsStyle Guide in *User Interface Programming Concepts Volume 1* angegebenen Richtlinien übereinstimmen.

Dieser Artikel enthält Informationen zu folgenden Themen:

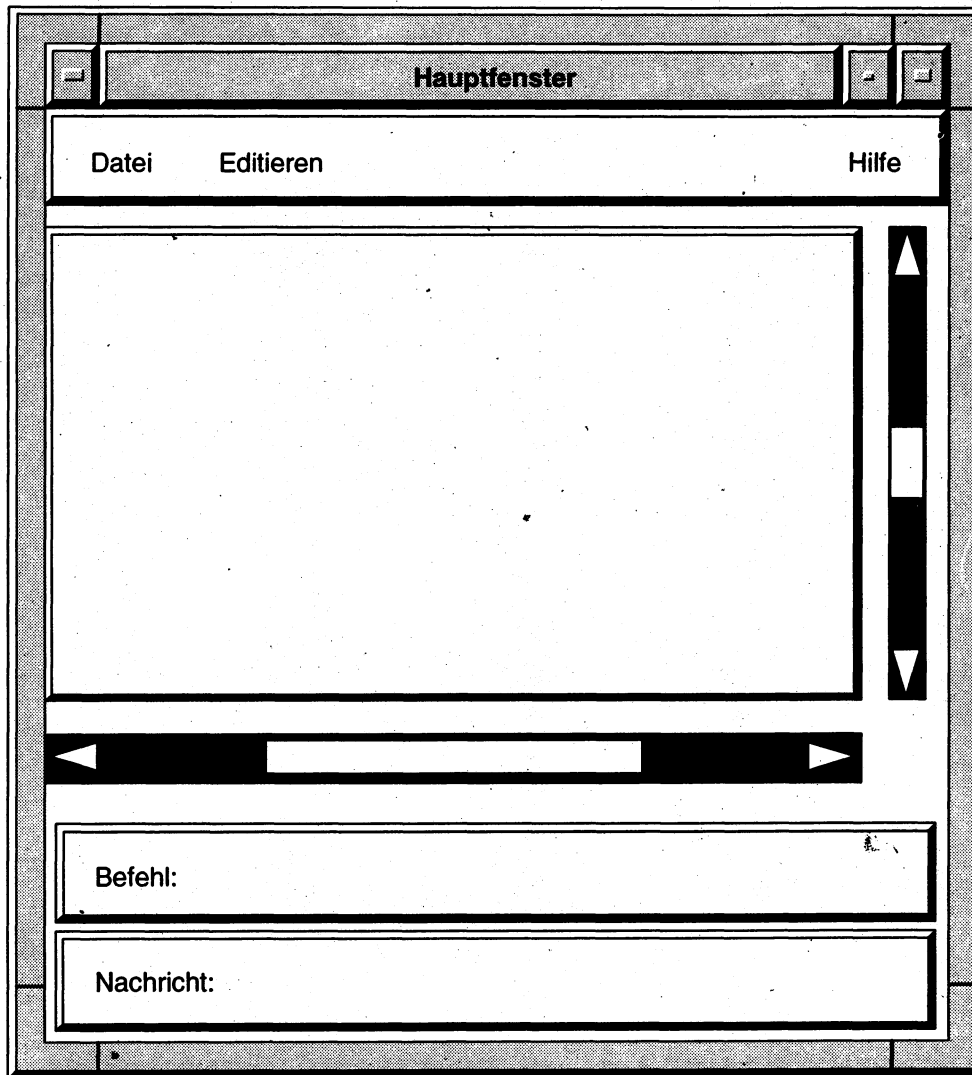
- Arbeiten mit dem Hauptfenster auf Seite 4-21
- Benutzung von Knöpfen auf Seite 4-22
- Benutzung von Menüs auf Seite 4-24
- Benutzung von Dialogfenstern auf Seite 4-29
- Benutzung von einstellbaren Komponenten auf Seite 4-32
- Benutzung von Listen auf Seite 4-35
- Text eingeben und editieren auf Seite 4-36
- Benutzung der Tastatur auf Seite 4-38
- Benutzung der Funktion "ziehen und übergeben" auf Seite 4-39

Arbeiten mit dem Hauptfenster

Jede AIXwindows-Anwendung enthält ein *Hauptfenster*, in dem der Inhalt der Anwendung verwaltet wird. Im allgemeinen erfolgt der Zugriff auf die Primärfunktionen einer Anwendung über Komponenten im Hauptfenster. Die Komponenten in einem Hauptfenster sind von Anwendung zu Anwendung unterschiedlich. Das Hauptfenster in einer typischen Anwendung kann eine Menüleiste, Schiebeleisten, Bereiche für die Eingabe von Befehlen und zur Anzeige von Nachrichten sowie anwendungsspezifische Komponenten enthalten.

Die *Menüleiste* ist eine horizontale Leiste, die sich unmittelbar unter der Titelleiste des Fensters befindet. In der Menüleiste werden die wichtigsten Funktionen einer Anwendung verwaltet. Der Bereich unter der Menüleiste enthält die Komponenten für die Primärfunktionen der Anwendung. Zum Beispiel würde in einem Tabellenkalkulationsprogramm das Tabellenkalkulationsraster in diesem Bereich angezeigt. In einem elektronischen Postprogramm würde dieser Bereich Komponenten für das Lesen und Senden von Nachrichten enthalten. Mit Hilfe von vertikalen und horizontalen Schiebeleisten können Daten angezeigt werden, für die mehr Platz erforderlich ist als im Hauptfenster zur Verfügung steht. In Anwendungen, in denen neben der Grafikschnittstelle eine Befehlszeilenschnittstelle verwendet wird, steht ein Befehlsbereich für die Eingabe von Befehlen zur Verfügung. Der Befehlszeilenbereich erstreckt sich von Begrenzung zu Begrenzung am unteren Rand des Hauptfensters. Eine Anwendung kann auch einen Bereich für die Anzeige kurzer Nachrichten enthalten.

Die folgende Abbildung zeigt die Basiskomponenten eines Hauptfensters für AIXwindows-Anwendungen. Die meisten Anwendungen verfügen über zusätzliche Komponenten in den Hauptfenstern.



Typisches Hauptfenster und Benutzerbereiche

Benutzung von Knöpfen

Knöpfe sind Basiskomponenten, mit denen Funktionen in AIXwindows-Anwendungen aktiviert werden. Knöpfe sind eigenständige Schnittstellenkomponenten, die auch beim Erstellen von Menüs und Dialogfenstern verwendet werden.

Benutzung von Druckknöpfen

Ein *Druckknopf* enthält eine Marke, die die Funktion des Druckknopfs angibt. Bei der Marke kann es sich um Text oder ein Abbild handeln. Der AIXwindows-Druckknopf besitzt eine dreidimensionale Darstellung. Ein Druckknopf erscheint von der Anzeige abgesetzt. Nach Auswählen des Knopfes erscheint der Knopf in die Anzeige eingelassen. Die folgende Abbildung zeigt einen typischen Druckknopf.



Zum Aktivieren der Funktionalität eines Druckknopfs mit der Maus wie folgt vorgehen:

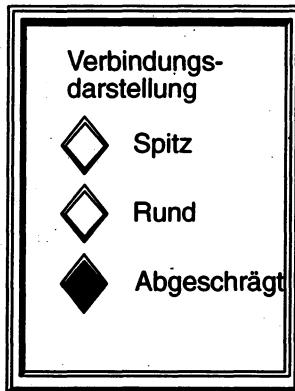
1. Den Zeiger auf den Druckknopf setzen.
2. Mit der linken Maustaste klicken.

Verfügt der Druckknopf über den Eingabebereich, kann er über die Tastatur aktiviert werden, indem die Rückföhrtaste, die Auswahl taste oder die Leertaste gedrückt wird.

Benutzung von Umschaltknöpfen

Ein *Umschaltknopf* ist ein Knopf, der zwei Status besitzen kann: ein oder aus. Umschaltknöpfe erlauben dem Benutzer normalerweise die Auswahl einer oder mehrerer Optionen aus einer Auswahlliste. Jeder Umschaltknopf verfügt über eine Marke und eine Grafik, die den aktuellen Status angibt.

Radioknöpfe sind eine Gruppe von Umschaltknöpfen, bei denen jeweils nur einer der Umschaltknöpfe eingeschaltet sein kann. Die zugehörige grafische Anzeige für den Status von Radioknöpfen ist eine Raute. Wenn die Raute ausgefüllt ist, ist der Knopf eingeschaltet; wenn die Raute leer ist, ist der Knopf ausgeschaltet. Die folgende Abbildung zeigt einen typischen Radioknopf.



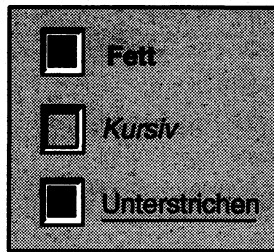
Zum Auswählen eines Radioknopfs mit der Maus wie folgt vorgehen:

1. Den Zeiger auf die Marke des Radioknopfs stellen.
2. Mit der linken Maustaste klicken.

Wird auf einen Radioknopf geklickt, wenn dieser sich im ausgeschalteten Status befindet, wird der Radioknopf dadurch eingeschaltet. Jeder andere Radioknopf in der Gruppe, der sich im eingeschalteten Status befindet, wird auf 'aus' gesetzt. Wird auf einen Radioknopf geklickt, der bereits eingeschaltet ist, bleibt der Knopf eingeschaltet.

Besitzt ein Radioknopf den Eingabebereich, kann er eingeschaltet werden, indem die Auswahl taste oder die Leertaste gedrückt wird. Jeder Radioknopf in der Gruppe, der eingeschaltet war, wird ausgeschaltet. Die Verwendung der Pfeiltasten ermöglicht das Navigieren in einer Gruppe von Radioknöpfen.

Markierungsknöpfe sind eine Gruppe von Umschaltknöpfen, bei der eine beliebige Anzahl der Umschaltknöpfe gleichzeitig eingeschaltet sein kann. Zusammen mit dem Markierungsknopf wird normalerweise ein Quadrat als Grafik angezeigt, das bei eingeschaltetem Knopf ausgefüllt oder anders eingefärbt ist. Die folgende Abbildung zeigt einen Umschaltknopf.



Zum Auswählen eines Markierungsknopfs mit der Maus wie folgt vorgehen:

1. Den Zeiger auf die Marke des Markierungsknopfs setzen.
2. Mit der linken Maustaste klicken.

Wird auf einen Markierungsknopf geklickt, wenn dieser sich im ausgeschalteten Status befindet, wird der Markierungsknopf dadurch eingeschaltet. Der Knopf behält diesen Status bei, wenn auf andere Markierungsknöpfe geklickt wird. Wird auf einen Markierungsknopf geklickt, wenn dieser sich im eingeschalteten Status befindet, wird der Markierungsknopf dadurch ausgeschaltet.

Besitzt ein Markierungsknopf den Eingabebereich, kann zwischen dem Status 'ein' und 'aus' hin- und hergeschaltet werden, indem die Auswahl Taste oder die Leertaste gedrückt wird. Die Verwendung der Pfeiltasten ermöglicht das Navigieren in einer Gruppe von Markierungsknöpfen.

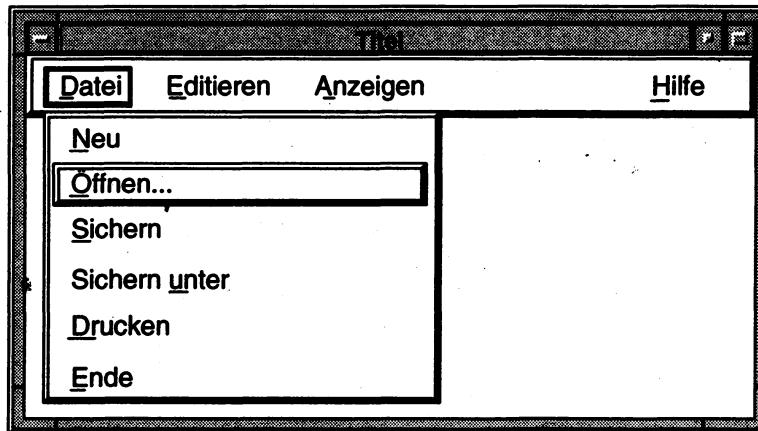
Benutzung von Menüs

Menüs enthalten Steuerelemente für häufig verwendete Funktionen einer Anwendung. Nach Auswählen eines Menüs kann durch Auswahl eines Eintrags aus diesem Menü eine zugehörige Aktion eingeleitet werden. Während ein Menü geöffnet ist, ist keine Interaktion mit anderen Komponenten der Anwendung möglich. AIXwindows unterstützt folgende Arten von Menüs:

- Aktionsfenstermenüs
- Dialogfenstermenüs
- Auswahlmenüs
- Abreißmenüs

Benutzung von Aktionsfenstermenüs

Der Zugriff auf *Aktionsfenstermenüs* erfolgt über die Marken, durch die sie dargestellt werden. Nach Auswahl der Marke eines Aktionsfenstermenüs wird das Menüteilfenster angezeigt. Die in einer Anwendung am häufigsten verwendeten Aktionsfenstermenüs werden in die Menüleiste einer Anwendung gestellt. Die folgende Abbildung zeigt das Menü "Datei" und die Menüleiste.



Das Menü "Datei" und die zugehörigen Auswahlmöglichkeiten

Aktionsfenstermenüs enthalten Einträge, mit denen häufig benötigte Anwendungsaktionen ausgeführt werden. Aktionsfenstermenüs können außerdem verschachtelte Untermenüs enthalten, die als *weiterführende Menüs* bezeichnet werden.

AIXwindows ermöglicht zwei verschiedene Arten von mausgestützter Interaktion mit Aktionsfenstermenüs: Interaktion *mit Festhalten der Maustaste* und Interaktion *ohne Festhalten der Maustaste*.

Zum Auswählen eines Menüpunkts durch Interaktion mit Festhalten der Maustaste wie folgt vorgehen:

1. Den Zeiger auf den Namen des Menüs setzen.
2. Die linke Maustaste drücken und gedrückt halten, um das Menü anzuzeigen.
3. Den Zeiger im Menüteifenster nach unten ziehen und den auszuwählenden Menüpunkt hervorheben.
4. Die linke Maustaste freigeben, um den Menüpunkt auszuwählen.

Bei einer Interaktion mit Festhalten der Maustaste kann der Inhalt jedes Menüs in der Menüleiste angezeigt werden, indem die linke Maustaste gedrückt wird und der Zeiger über die Titel in der Menüleiste gezogen wird. Beim Ziehen durch die verschiedenen Titel wird das zugehörige Menü angezeigt. Eine Interaktion mit Festhalten der Maustaste kann abgebrochen werden, indem der Zeiger an eine beliebige Stelle außerhalb des Menüsystems gestellt und die linke Maustaste freigegeben wird.

Zum Auswählen eines Menüpunkts durch Interaktion ohne Festhalten der Maustaste wie folgt vorgehen:

1. Den Zeiger auf den Namen eines Menüs setzen.
2. Mit der linken Maustaste klicken, um das Menü zu öffnen.
3. Mit der linken Maustaste auf den auszuwählenden Menüpunkt klicken.

Bei einer Interaktion ohne Festhalten der Maustaste kann das Menü abgebrochen werden, indem der Zeiger auf den Menünamen oder an eine beliebige Stelle außerhalb des Menüsystems gesetzt und die linke Maustaste gedrückt wird.

Zum Auswählen eines Menüpunkts über die Tastatur wie folgt vorgehen:

1. Die Taste F10 drücken, um die Menüleiste auszuwählen.
2. Mit Hilfe der Linkspfeil- und Rechtspfeiltaste ein Menü auswählen.

3. Zum Öffnen des Menüs die Rückföhrtaste, die Auswahlaste oder die Leertaste drücken.
4. Mit Hilfe der Pfeiltasten den auszuwählenden Menüpunkt hervorheben.
5. Die Rückföhrtaste, die Auswahlaste oder die Leertaste drücken, um den Menüpunkt zu aktivieren.

Bei Verwendung der Tastatur kann ein Menü durch Drücken der Taste F10 abgebrochen werden. Der Eingabebereich kehrt zu der Komponente zurück, bei der er sich vor der Menüinteraktion befand. Die Abbruchtaste drücken, um die Menühierarchie zu verlassen. Befindet sich der Eingabebereich in der Menüleiste oder in einem geöffneten Menü, hat die Abbruchtaste die gleiche Funktion wie die Taste F10. Andernfalls bewirkt das Drücken der Abbruchtaste das Schließen des letzten weiterführenden Untermenüs.

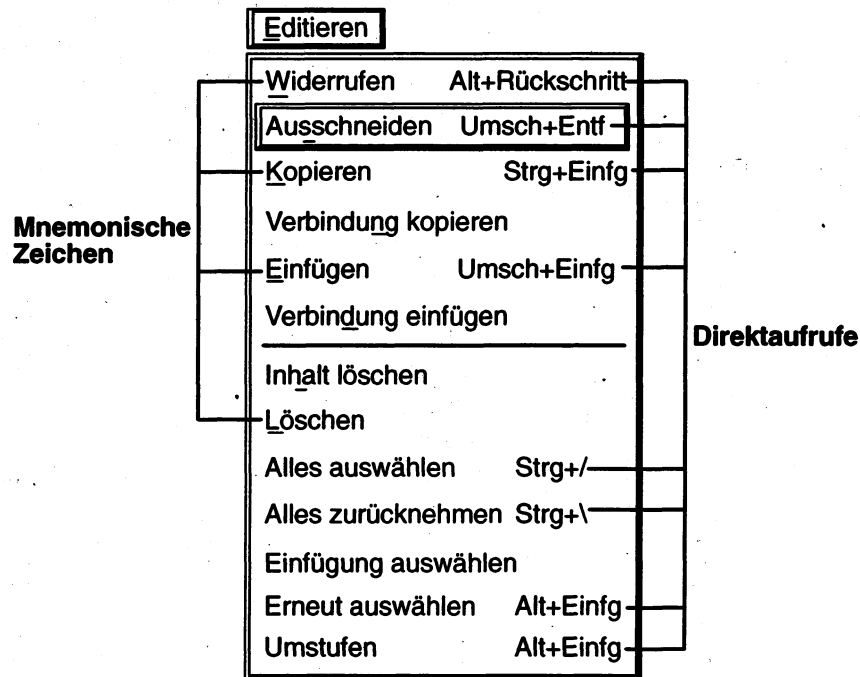
In Anwendungen, die mit dem AIXwindows Style Guide in Einklang stehen, befindet sich die Option Ende immer im ersten Aktionsfenstermenü in der Menüleiste, bei dem es sich in der Regel um das Menü "Datei" handelt. Besitzt eine Anwendung ein Hilfemenü, ist dieses immer das am weitesten rechts befindliche Menü in der Menüleiste. Anwendungen, die mit dem AIXwindows Style Guide in Einklang stehen, können auch andere im Style Guide definierte Standardmenüs verwenden, wie z. B. die Menüs "Editieren", "Anzeigen" und "Optionen".

Benutzung von Menüdirektaufrufen

Neben dem Namen des Menüpunkts kann eine Menüauswahl weitere visuelle Hinweise zu ihrer Funktionalität geben. Zum Beispiel wird das Vorhandensein eines weiterführenden Untermenüs durch ein Dreieck auf der rechten Seite eines Menüpunkts angezeigt. Ein abgeblendeter Name zeigt an, daß die Auswahl zur Zeit nicht gültig ist. Die ... (Auslassungspunkte) hinter einem Menüpunkt weisen darauf hin, daß weitere Angaben gemacht werden müssen, bevor die entsprechende Aktion ausgeführt werden kann.

Durch Angabe mnemonischer Auswahlmöglichkeiten kann dem Benutzer einer Anwendung ein schneller Zugriff auf Menüs und Menüpunkte ermöglicht werden. Ist einem Menü oder einem Menüpunkt eine mnemonische Auswahlmöglichkeit zugeordnet, ist das mnemonische Zeichen in der Marke des Menüs oder des Menüpunkts unterstrichen. Durch Drücken der Taste Alt und der mnemonischen Taste für ein Menü kann das zugehörige Menü geöffnet werden. Sobald ein Menü geöffnet ist, ist das Drücken der mnemonischen Taste für einen Menüpunkt äquivalent zum Auswählen des zugehörigen Menüpunkts. Mnemonische Zeichen für Menüpunkte sind nur aktiv, wenn das Menü mit den Einträgen geöffnet ist.

Bei einigen Anwendungen wird eine Kombination von Tastenanschlägen, sogenannte *Direktaufrufe*, verwendet, um einen schnelleren Zugriff auf Menüpunkte zu ermöglichen. Wenn einem Menüpunkt ein Direktaufwurf zugeordnet ist, werden die erforderlichen Tastenanschläge rechts neben dem Namen des Menüpunkts angezeigt. Bei Eingabe einer Direktaufwurfrolge wird der Menüpunkt unabhängig davon aktiviert, ob das Menü geöffnet ist oder nicht. Die folgende Abbildung zeigt ein Aktionsfenstermenü mit mnemonischen Zeichen und Direktaufrufen.



Benutzung von Dialogfenstermenüs

Verschiedenen Bereichen einer Anwendung können *Dialogfenstermenüs* zugeordnet sein. Auf ein Dialogfenstermenü kann nur zugegriffen werden, wenn sich der Zeiger über einem Bereich befindet, dem ein Dialogfenstermenü zugeordnet ist. Dialogfenstermenüs bieten keinerlei visuelle Hinweise auf ihre Existenz. Aus diesem Grund stehen für Dialogfenstermenüs nur Direktaufrufe für Funktionen zur Verfügung, die von jedem Punkt des Programms aufgerufen werden können. Der Inhalt eines Dialogfenstermenüs kann *kontextabhängig* sein; das bedeutet, daß die Einträge in einem Menü sich abhängig von einer Reihe von Faktoren unterscheiden können, wie z. B. der Position des Zeigers, wenn das Menü geöffnet ist, oder dem Auswahlstatus einer Komponente.

Zum Aktivieren eines Dialogfenstermenüs mit der Maus wie folgt vorgehen:

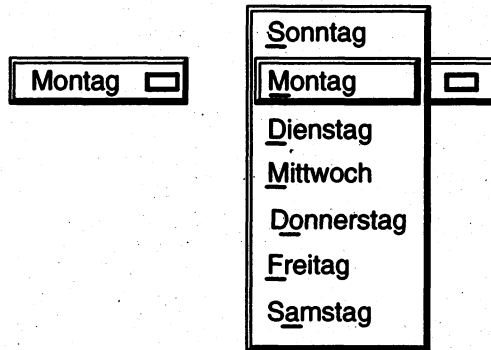
1. Den Zeiger auf eine Komponente stellen, die ein Dialogfenstermenü enthält. Die Dokumentation zur Anwendung gibt Auskunft darüber, wo sich die Dialogfenstermenüs befinden.
2. Die rechte Maustaste drücken.

Dialogfenstermenüs ermöglichen sowohl eine Interaktion mit Festhalten der Maustaste als auch eine Interaktion ohne Festhalten der Maustaste. Die Auswahl von Menüpunkten aus einem Dialogfenstermenü erfolgt auf die gleiche Weise wie bei einem Aktionsfenstermenü. Menüpunkte in einem geöffneten Dialogfenstermenü können mit Hilfe der linken oder der rechten Maustaste ausgewählt werden. Zum Abbrechen eines geöffneten Dialogfenstermenüs mit der linken oder der rechten Maustaste außerhalb des Menüteiffensters klicken.

Befindet sich der Eingabebereich in einem Bereich mit einem inaktiven Dialogfenstermenü, kann das Menü durch Drücken der Menütaste oder mit Hilfe der Tastenfolge Umsch-F10 geöffnet werden. Ein Menüpunkt in einem Dialogfenstermenü kann mit Hilfe der Tastatur ausgewählt werden, indem die Pfeiltasten verwendet und die Rückföhrtaste gedrückt wird. Zum Abbrechen eines Dialogfenstermenüs über die Tastatur die Abbruchtaste drücken.

Benutzung von Auswahlmenüs

Ein *Auswahlmenü* stellt eine Liste von Auswahlmöglichkeiten innerhalb eines relativ kleinen Bereichs zur Verfügung. Eine Balkengrafik auf der rechten Seite eines Auswahlmenüs zeigt das Vorhandensein eines Auswahlmenüs an. Nach Auswahl eines Auswahlknopfs wird das zugehörige Auswahlmenü angezeigt, und es kann eine Auswahl getroffen werden. Wenn das Auswahlmenü nicht geöffnet ist, wird durch den Auswahlknopf die aktuelle Auswahl angezeigt. Die folgende Abbildung zeigt einen Auswahlknopf mit der Balkengrafik und dem zugehörigen Auswahlmenü.



Zum Öffnen eines Auswahlmenüs mit der Maus wie folgt vorgehen:

1. Den Zeiger auf den Auswahlknopf setzen.
2. Die linke Maustaste drücken.

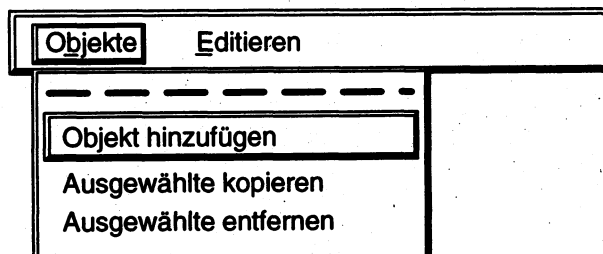
Auswahlmenüs ermöglichen sowohl eine Interaktion mit Festhalten der Maustaste als auch eine Interaktion ohne Festhalten der Maustaste. Die Auswahl von Menüpunkten aus einem Auswahlmenü erfolgt auf die gleiche Weise wie bei einem Aktionsfenstermenü. Zum Abbrechen eines geöffneten Auswahlmenüs mit der linken Maustaste auf die Balkengrafik oder auf eine beliebige Stelle außerhalb des Menüteilfensters klicken.

Besitzt ein Auswahlknopf den Eingabebereich, kann das Menü über die Tastatur durch Drücken der Auswahl Taste oder der Leertaste geöffnet werden. Ein Menüpunkt im Auswahlmenü kann ausgewählt werden, indem die Pfeiltasten verwendet werden und die Auswahl Taste oder die Leertaste gedrückt wird. Zum Abbrechen eines Auswahlmenüs über die Tastatur die Abbruchtaste drücken.

Benutzung von Abreißmenüs

Abreißmenüs erlauben es dem Benutzer, häufig benutzte Menüs auf dem Bildschirm zu behalten. Normalerweise verschwindet ein Menü von der Anzeige, wenn eine Auswahl getroffen wurde. Ein Abreißmenü bleibt dagegen angezeigt, bis es abgebrochen wird.

Abreißmenüs können Aktionsfenstermenüs, Dialogfenstermenüs und Auswahlmenüs sein. Steht die Abreißfunktion für ein bestimmtes Menü zur Verfügung, wird am oberen Rand des Menüs ein Abreißknopf mit einer gestrichelten Linie angezeigt. Die folgende Abbildung zeigt ein Abreißmenü.



Zum Abreißen eines Menüs wie folgt vorgehen:

1. Das Menü mit Hilfe der Tastatur oder der Maus öffnen.
2. Den Abreißknopf wie einen normalen Menüpunkt auswählen. Der Cursor kann mit Hilfe der Pfeiltasten oder der Maus zu dem Abreißknopf bewegt werden.

Das Menü wird abgerissen und an die Position gestellt, an der es sich ursprünglich befand. Es kann wie ein normales Fenster an eine neue Position bewegt werden.

Um ein Menü abzureißen und es an eine neue Position zu bewegen, wie folgt vorgehen:

1. Das Menü mit Hilfe der Tastatur oder der Maus öffnen.
2. Die mittlere Maustaste über dem Abreißknopf drücken und gedrückt halten.
3. Das Menü an die gewünschte Position bewegen und die mittlere Maustaste freigeben.

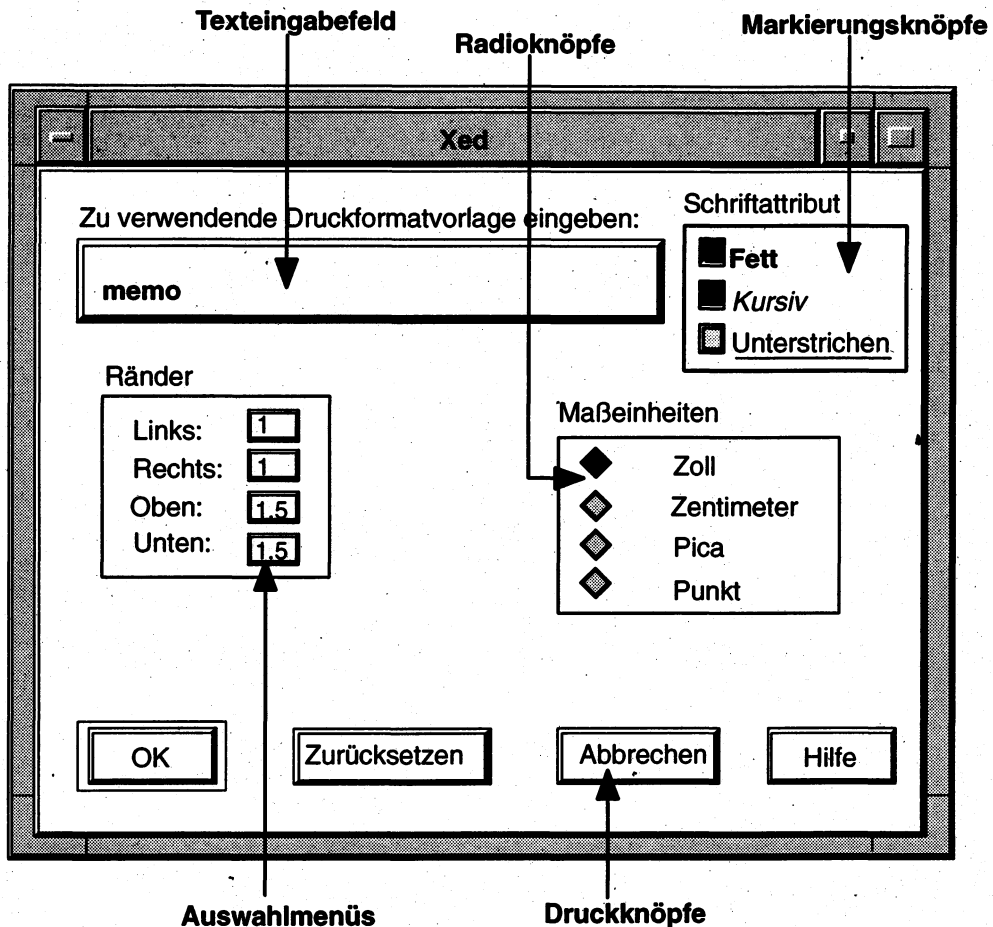
Nach dem Abreißen eines Menüs wird das ursprüngliche Menü geschlossen. Ein abgerissenes Menü ist wie ein normales Sekundärfenster aufgebaut. Es besitzt eine Titelleiste, einen Fensterrahmen und Knöpfe für die Fensterverwaltung. Das abgerissene Menü kann über die Titelleiste neu positioniert werden. Es steht nur jeweils ein Exemplar jedes Abreißmenüs zur Verfügung. Wird ein Menü abgerissen, das bereits abgerissen wurde, wird das erste Exemplar durch das neue Exemplar ersetzt. Das Menü kann jederzeit an seiner normalen Position geöffnet werden, ohne die abgerissene Version zu beeinflussen.

Zum Entfernen eines Abreißmenüs die Option **Schließen** aus dem Fenstermenü auswählen oder die Abbruchtaste drücken, wenn sich der Eingabebereich im Abreißmenü befindet.

Benutzung von Dialogfenstern

AIXwindows-Anwendungen verwenden *Dialogfenster* für weniger wichtige Aufgaben in einer Anwendung. In einem Dialogfenster können Informationen (wie z. B. eine Fehlermeldung) angezeigt werden oder es kann die Eingabe von Informationen (wie z. B. der Name einer zu öffnenden Datei) angefordert werden. Dialogfenster können die Interaktion mit anderen Fenstern in einer Anwendung einschränken. Bei einem *Modaldialogfenster* muß auf die Anzeige reagiert werden, bevor eine Interaktion mit anderen Komponenten in der Anwendung möglich ist. Ein *nichtmodales Dialogfenster* schränkt die Interaktion mit der übrigen Anwendung nicht ein, so daß es für die Dauer einer Anwendung verfügbar bleiben und zur Ausführung mehrerer Aktionen verwendet werden kann.

In Anwendungen können Dialogfenster mit einer beliebigen Anzahl und Kombination von Komponenten verwendet werden. Die folgende Abbildung zeigt ein Dialogfenster mit den verschiedenen Komponenten zur Anpassung der Einstellungen für einen Texteditor.

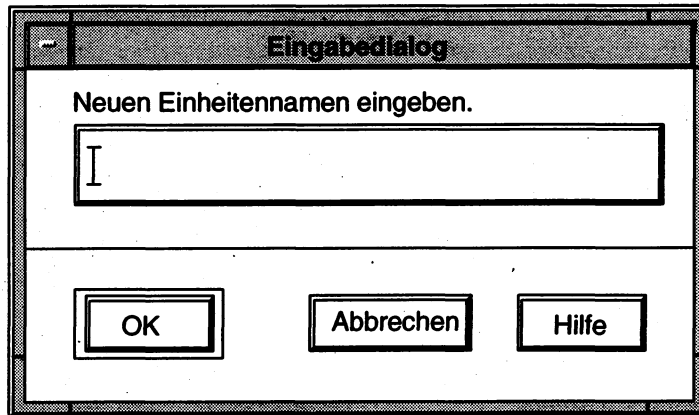


Ein Dialogfenster enthält zumindest eine Marke und einen Druckknopf für die Eingabe einer Antwort. Das AIXwindows Toolkit stellt folgende Arten von Dialogfenstern für allgemeine Funktionen bereit, die in vielen Anwendungen verwendet werden:

- Eingabedialoge für Benutzereingaben.
- Befehlsdialoge für die Eingabe von Tastaturbefehlen.
- Auswahldialoge für die Auswahl aus einer Liste von Auswahlmöglichkeiten.
- Dateiauswahldialoge für die Eingabe des Namens einer zu verarbeitenden Datei.
- Nachrichtendialoge für die Anzeige von Informationen.

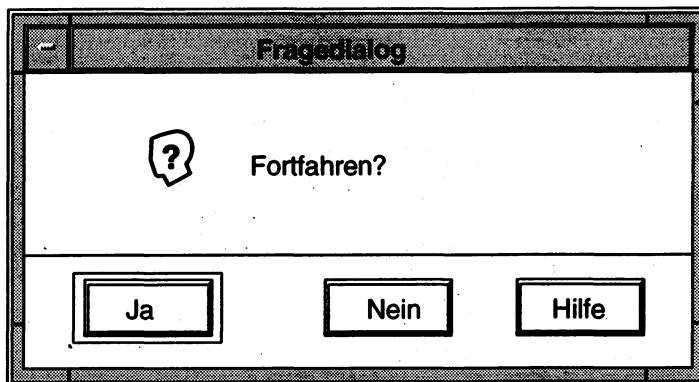
Jedes dieser Dialogfenster verfügt über eine eigene Definition der Benutzerinteraktion.

Eine Anwendung verwendet normalerweise einen Eingabedialog dazu, Informationen abzurufen, die für die Fortsetzung der Verarbeitung benötigt werden. Ein Eingabedialog unterbricht normalerweise die Interaktion des Benutzers mit dem übrigen Programm. Der Eingabedialog enthält eine Nachricht, einen Texteingabebereich, in den der Benutzer die angeforderten Informationen eingibt, und Druckknöpfe für das Reagieren auf das Dialogfenster. Die folgende Abbildung zeigt einen Eingabedialog.



Eingabedialog

AIXwindows bietet außerdem eine Reihe von Nachrichtendialogen, darunter ein Fehlerdialog, ein Informationsdialog, ein Fragedialog, ein Warnungsdialog und ein Arbeitsdialog. Jede Art von Nachrichtendialog enthält ein grafisches Symbol, das den Zweck des Symbols veranschaulicht, sowie eine Textnachricht und die für das Dialogfenster erforderlichen Druckknopfbefehle. Die folgende Abbildung zeigt einen Fragedialog. Ein Fragedialog enthält ein Fragezeichensymbol, eine an den Benutzer gerichtete Frage und Druckknöpfe, mit deren Hilfe der Benutzer auf die Frage antworten kann.



Fragedialog

Jedes Dialogfenster enthält eine Zeile mit Druckknöpfen am unteren Rand des Fensters, mit deren Hilfe der Benutzer auf das Dialogfenster reagieren kann. Im AIXwindows Style Guide sind eine Reihe von Standardaktionen für Dialogfenster definiert. Zweck der Standardaktionen ist es, eine konsistente Verhaltensweise in verschiedenen AIXwindows zu erreichen. In Anwendungen, die mit dem AIXwindows Style Guide in Einklang stehen, haben die Standarddialogfensteraktionen folgende Bedeutung und Verhaltensweise:

- Ja und Nein sind Antworten auf eindeutig gestellte Fragen. Nach Auswahl eines dieser Knöpfe wird das Dialogfenster geschlossen.
- OK bewirkt das Anwenden aller im Dialogfenster vorgenommenen Änderungen und das Schließen des Dialogfensters.
- Schließen bewirkt das Schließen des Dialogfensters, ohne daß eine Aktion ausgeführt wird.
- Anwenden bewirkt das Anwenden aller im Dialogfenster vorgenommenen Änderungen, ohne daß das Dialogfenster geschlossen wird.

- Wiederholen bewirkt das erneute Ausführen der laufenden Task. Diese Aktion ist normalerweise in Fehlerdialogen zu sehen.
- Stopp bewirkt das Beenden der aktuellen Task am frühestmöglichen Unterbrechungspunkt. Diese Aktion ist im allgemeinen in Arbeitsdialogen zu finden.
- Anhalten bewirkt das Anhalten der momentan ausgeführten Aktion. Diese Aktion ist normalerweise in Arbeitsdialogen zu finden.
- Wiederaufnehmen wird in Verbindung mit Anhalten verwendet. Diese Funktion bewirkt das Wiederaufnehmen einer angehaltenen Task.
- Zurücksetzen bewirkt das Abbrechen aller Änderungen, die noch nicht angewendet wurden.
- Abbrechen bewirkt das Schließen des Dialogfensters, ohne daß Änderungen, die noch nicht angewendet wurden, ausgeführt werden.
- Hilfe bewirkt das Anzeigen von Informationen über den Dialog.

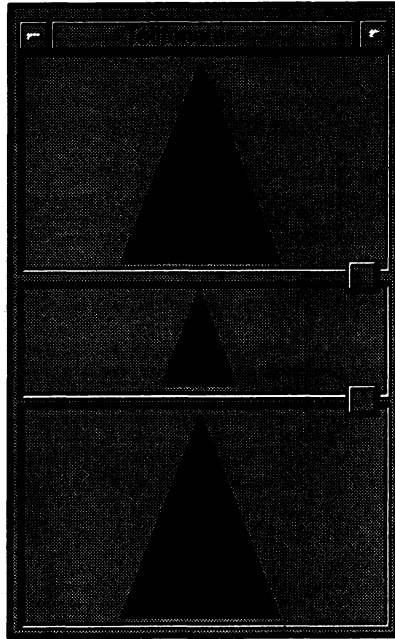
Benutzung von einstellbaren Komponenten

AIXwindows enthält folgende einstellbare Komponenten:

- Teilfenster für die Einstellung der relativen Größe von Komponenten.
- Blätterfenster für die Anzeige des gesamten Inhalts einer Komponente, wenn im Anzeigebereich nicht der gesamte Inhalt auf einmal angezeigt werden kann.
- Skalen für die Auswahl eines Werts innerhalb eines Bereichs.

Benutzung von Teilfenstern

Ein Teilfenster unterteilt Komponenten in separate Teile. Teilfenster sind Abschnitte innerhalb eines Fensters. Benachbarte Teilfenster sind voneinander durch eine horizontale Trennlinie und ein *Größensteuerfeld* getrennt. Die folgende Abbildung zeigt ein Teilfenster mit einem Größensteuerfeld. Das Größensteuerfeld ermöglicht die Einstellung der relativen Größe von benachbarten Teilfenstern. Durch Verschieben des Größensteuerfelds wird die Position der Trennlinie zwischen zwei Teilfenstern und damit die Größe der Teilfenster verändert. Beim Verschieben des Größensteuerfelds wird das Teilfenster, das sich in der Verschiebungsrichtung befindet, kleiner, während das andere Teilfenster proportional größer wird.



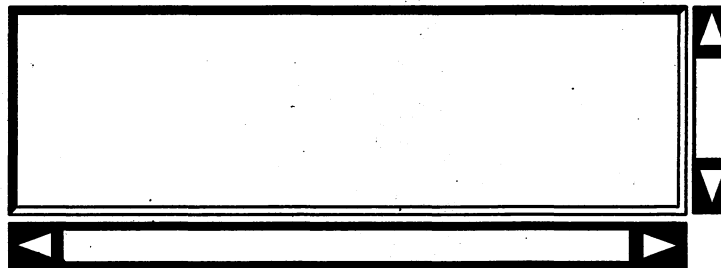
Zum Ändern der Größe eines Teilfensters wie folgt vorgehen:

1. Den Zeiger auf das Größensteuerfeld setzen.
2. Wenn sich die Zeigerform zu einem Pluszeichen (+) verändert, die linke Maustaste drücken und gedrückt halten.
3. Die Maus ziehen, bis die Teilfenster die gewünschte Größe haben.
4. Die linke Maustaste freigeben.

Besitzt ein Größensteuerfeld den Eingabebereich, kann ein Teilfenster über die Tastatur eingestellt werden. Mit Hilfe der Aufwärts- und Abwärtspfeiltasten kann die Trennlinie in kleinen Schritten verschoben werden. Soll die Trennlinie in größeren Schritten verschoben werden, ist beim Drücken der Pfeiltasten die Steuerungstaste gedrückt zu halten.

Benutzung von Blätterfenstern

Ein Blätterfenster umgibt andere Komponenten und stellt ggf. *Schiebeleisten* zur Verfügung, mit deren Hilfe der Inhalt des Fensters angezeigt werden kann. Mit Schiebeleisten wird der sichtbare Inhalt eines Fensters verändert, wenn der gesamte Inhalt nicht in das Fenster paßt. Eine Schiebeleiste besteht aus einer verschiebbaren Komponente, einem sogenannten *Schiebefeld*, das innerhalb einer Leiste verschoben wird, die die Länge des Blätterbereichs besitzt. Position und Größe des Schiebefelds in der Leiste geben die relative Position und die Größe des sichtbaren Bereichs der Blätterkomponente an. Eine Schiebeleiste verfügt außerdem an jedem Ende der Leiste über Pfeilköpfe, mit deren Hilfe die Schiebeleiste verschoben werden kann. Die folgende Abbildung zeigt ein Blätterfenster.



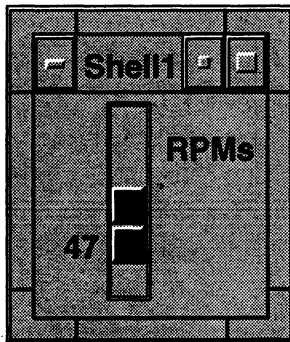
Ein Bereich kann mit Hilfe der Maus wie folgt verschoben werden:

- Soll das Schiebefeld an eine beliebige Position bewegt werden, in der Leiste die mittlere Maustaste drücken. Daraufhin springt das Schiebefeld an diese Position. Das Schiebefeld kann dann vor- und zurückgezogen werden, um den Bereich zu verschieben.
- Soll das Schiebefeld an eine bestimmte Position verschoben werden, den Zeiger auf das Schiebefeld setzen. Die linke Maustaste drücken und gedrückt halten. Das Schiebefeld kann dann zu jedem beliebigen Bereich vor- und zurückgezogen werden.
- Soll um eine Seite nach oben oder unten geblättert werden (mit Hilfe der vertikalen Schiebeleisten), mit der linken Maustaste in der Leiste oberhalb oder unterhalb des Schiebefelds klicken. Horizontale Schiebeleisten verfügen über den gleichen Mechanismus zum seitenweisen Verschieben nach links oder rechts.
- Soll der Bereich um eine Einheit verschoben werden, mit der linken Maustaste auf den Pfeilknopf für die Richtung klicken, in die der Bereich verschoben werden soll.

Detaillierte Informationen zum Navigieren in einem Blätterfenster mit Hilfe der Tastatur können den folgenden Abschnitten zu Listen und Textkomponenten entnommen werden.

Benutzung von Skalen

Mit Hilfe einer Skala kann ein Wert aus einem fortlaufenden Bereich ausgewählt werden. Wie eine Schiebeleiste besteht eine Skala aus einem Schiebefeld, das sich in einer Leiste befindet, und Pfeilknopfen. Eine Skala kann vertikal oder horizontal ausgerichtet sein. Die Position des Schiebefelds gibt den aktuellen Wert relativ zum gesamten Bereich an. Der aktuelle Wert einer Skala kann neben dem Schiebefeld angezeigt werden. Eine Skala kann außerdem Marken enthalten, die den Wert der Skala angeben, wenn sich das Schiebefeld an bestimmten Positionen befindet. Die folgende Abbildung zeigt eine Skala.



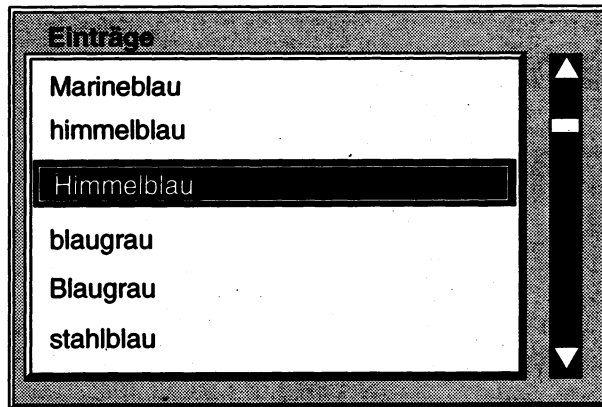
Eine Skala kann mit Hilfe der Maus wie folgt manipuliert werden:

- Soll das Schiebefeld an eine beliebige Position bewegt werden, in der Leiste die mittlere Maustaste drücken. Daraufhin springt das Schiebefeld an diese Position. Das Schiebefeld kann dann vor- und zurückgezogen werden, um den Wert der Skala zu ändern.
- Soll das Schiebefeld an eine bestimmte Position verschoben werden, den Zeiger auf das Schiebefeld setzen. Die linke Maustaste drücken und gedrückt halten. Das Schiebefeld kann dann zu jedem beliebigen Bereich vor- und zurückgezogen werden.
- Soll die Skala um eine Einheit verändert werden, mit der linken Maustaste auf den Pfeilknopf für die Richtung klicken, in die der Wert geändert werden soll.

Wenn eine Skala den Eingabebereich besitzt, kann der Wert der Skala über die Tastatur verändert werden. Zum Verschieben des Schiebefelds in Einzelschritten die entsprechende Pfeiltaste drücken. Zum Verschieben des Schiebefelds in größeren Schritten die Steuerungstaste drücken und gedrückt halten und die entsprechenden Pfeiltasten drücken.

Benutzung von Listen

Eine *Listenkomponente* stellt eine Liste von Einträgen bereit, unter denen eine Auswahl zu treffen ist. AIXwindows verfügt über Listenkomponenten, mit denen einzelne Einträge, mehrere Einträge oder nicht zusammenhängende Bereiche von Einträgen ausgewählt werden können. In einer Anwendung können Listen für verschiedene Auswahlarten verwendet werden: der Zweck der Liste innerhalb der Anwendung bestimmt die Anzahl der Einträge, die ausgewählt werden können. Eine Liste kann darüber hinaus vertikale und horizontale Schiebeleisten besitzen, wenn der Inhalt der Liste größer als das Fenster ist. Die ausgewählten Einträge in einer Liste sind hervorgehoben. Die folgende Abbildung zeigt eine Liste.



Zum Auswählen eines einzelnen Eintrags aus einer Liste wie folgt vorgehen:

1. Den Zeiger auf den auszuwählenden Eintrag setzen.
2. Mit der linken Maustaste klicken.

Eine Einzelauswahlliste kann auch Blätterauswahl unterstützen. Bei der Blätterauswahl kann ein Eintrag ausgewählt werden, indem die linke Maustaste auf einem Eintrag gedrückt, der Zeiger durch die Liste gezogen und die linke Maustaste auf dem auszuwählenden Eintrag freigegeben wird.

Zum Auswählen einer fortlaufenden Gruppe von Einträgen aus einer Liste wie folgt vorgehen:

1. Den Zeiger auf den ersten Eintrag in der Gruppe setzen.
2. Die linke Maustaste drücken und den Zeiger auf den letzten Eintrag in der Gruppe ziehen.
3. Die linke Maustaste freigegeben.

Zum Auswählen einer Gruppe von Einträgen kann der Benutzer auch einen einzelnen Eintrag auswählen, den Zeiger auf den letzten Eintrag in der Gruppe setzen und die Umschalttaste drücken, während er die linke Maustaste drückt. Daraufhin sind alle Einträge zwischen dem ersten und dem letzten Eintrag ausgewählt.

Zum Hinzufügen einer nicht zusammenhängenden Gruppe von Einträgen zu einer Auswahl wie folgt vorgehen:

1. Den Zeiger auf den ersten Eintrag in der nicht zusammenhängenden Gruppe setzen.
2. Die Steuerungstaste drücken und gedrückt halten und linke Maustaste drücken.
3. Den Zeiger zum letzten Eintrag in der Gruppe ziehen.
4. Die linke Maustaste freigegeben.

Besitzt eine Liste den Eingabebereich, kann der Benutzer durch die Einträge in der Liste navigieren und über die Tastatur seine Auswahl treffen. Listenkomponenten verwenden einen *Positionscursor*, um anzuzeigen, welcher Eintrag in der Liste den Eingabebereich besitzt. Der Positionscursor wird durch einen Rahmen um den Listeneintrag dargestellt. Mit Hilfe der Aufwärts- und Abwärtspfeiltasten kann der Positionscursor durch die Einträge in der Liste bewegt werden. Abhängig von der verwendeten Listenart kann die Auswahl mit dem Positionscursor verschoben werden. Wird die Auswahl nicht mit dem Positionscursor verschoben, kann ein Eintrag durch Drücken der Auswahlstaste oder der Leertaste ausgewählt werden.

Die Listenkomponente erlaubt es dem Benutzer, die verschiedenen Arten der Auswahl über die Tastatur zu treffen. Weitere Informationen zur Auswahl über die Tastatur befinden sich im AIXwindows Style Guide in *User Interface Programming Concepts Volume 1*.

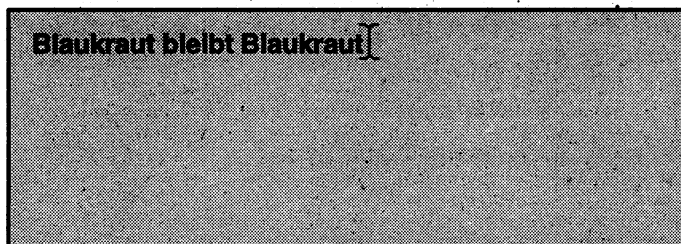
In einer Liste, in der mehrere Einträge ausgewählt werden können, können alle Einträge in der Liste mit der Tastenfolge Strg-/ ausgewählt werden. Mit der Tastenfolge Strg-\ kann die Auswahl zurückgenommen werden.

Eine laufende Ziehauswahloperation kann jederzeit durch Drücken der Abbruchtaste abgebrochen werden.

Text eingeben und editieren

Viele Anwendungen besitzen Bereiche zum Eingeben und Editieren von Text. Der *Texteingabebereich* kann eine einzelne Zeile zur Eingabe der Antwort auf eine Frage in einem Dialogfenster sein. Der Textbereich kann eine beliebige Länge haben, abhängig von den Anforderungen der Anwendung. In dieser Dokumentation wird davon ausgegangen, daß Anwendungen in einer englischsprachigen Umgebung verwendet werden. Werden Anwendungen verwendet, die auf anderen Umgebungen basieren, können Informationen zum Eingeben und Editieren von Text der Anwendungsdokumentation entnommen werden.

Textkomponenten können einen *Einfügekursor* verwenden, um anzuzeigen, an welcher Stelle die eingegebenen Informationen eingefügt werden. Wenn eine Textkomponente den Eingabebereich besitzt, ist der Einfügekursor durch einen blinkenden I-Balken-Cursor dargestellt. Die folgende Abbildung zeigt eine Textkomponente.



Zum Verschieben des Texteingügekursors wie folgt vorgehen:

1. Den Zeiger an die Position setzen, an der mit dem Eingeben begonnen werden soll.
2. Mit der linken Maustaste klicken. Der I-Balken-Einfügekursor wird an die entsprechende Position bewegt und zeigt dem Benutzer damit an, daß mit dem Eingeben begonnen werden kann.

Besitzt eine Textkomponente den Eingabebereich, ist die Navigation durch den Text über die Tastatur möglich. Mit der Rechtspfeil- und der Linkspfeiltaste wird der Einfügekursor um ein Zeichen verschoben. Mit der Aufwärts- und der Abwärtspfeiltaste wird der Cursor um eine Zeile verschoben. Werden die Linkspfeil- und Rechtspfeiltasten zusammen mit der Steuerungstaste gedrückt, wird der Cursor wortweise verschoben. Werden die Aufwärts- und Abwärtspfeiltasten zusammen mit der Steuerungstaste verwendet, wird der Cursor

absatzweise verschoben. Mit der Taste zum Zurückblättern, Vorwärtsblättern, nach links blättern oder nach rechts blättern und der Tastenfolge Strg-BildAuf und Strg-BildAb wird der Einfügecursor seitenweise verschoben.

Text kann wie folgt ausgewählt werden:

- Die linke Maustaste drücken und die Maus über den auszuwählenden Bereich ziehen.
- Mit der linken Maustaste zweimal auf ein Wort klicken, um das Wort auszuwählen.
- Zum Auswählen größerer Bereiche muß je nach Anwendung mehrmals mit der linken Maustaste geklickt werden.

Soll eine Auswahl mit der Maus zurückgenommen werden, den Zeiger auf eine beliebige Stelle außerhalb des ausgewählten Bereichs stellen und mit der linken Maustaste klicken. Soll der Einfügecursor verschoben werden, ohne die Auswahl zu ändern, den Zeiger an die Position setzen, an der mit dem Eingeben begonnen werden soll, die Steuerungstaste drücken und gedrückt halten und mit der linken Maustaste klicken.

Für das Kopieren oder Verschieben von Text innerhalb derselben Komponente oder von einer Komponente zu einer anderen stehen vier Möglichkeiten zur Verfügung: Übertragung über die Zwischenablage, Übertragung durch Ziehen, Primärübertragung und Schnellübertragung.

Zur Übertragung über die Zwischenablage wie folgt vorgehen:

1. Den zu kopierenden oder zu verschiebenden Text auswählen.
2. Die Tastenfolge Strg-Einfügen verwenden, um den Text in die Zwischenablage zu kopieren bzw. die Tastenfolge Umsch-Löschen verwenden, um den Text in die Zwischenablage auszuschneiden.
3. Den Einfügecursor an die Stelle setzen, an der der Text eingefügt werden soll.
4. Die Tastenfolge Umsch-Einfügen verwenden, um den Text an der neuen Position einzufügen.

Zum Verschieben oder Kopieren von Text mit Hilfe der Übertragung durch Ziehen wie folgt vorgehen:

1. Den zu kopierenden oder zu verschiebenden Text auswählen.
2. Den Zeiger auf den ausgewählten Text setzen und die mittlere Maustaste drücken.
3. Den Zeiger an die Position ziehen, an der der Text eingefügt werden soll.
4. Die mittlere Maustaste freigeben. Der Text wird nun normalerweise an die neue Position verschoben. Die Umschalttaste gedrückt halten, während die mittlere Maustaste freigegeben wird, um sicherzustellen, daß eine Verschiebeoperation ausgeführt wird. Die Steuerungstaste gedrückt halten, während die mittlere Maustaste freigegeben wird, um sicherzustellen, daß eine Kopieroperation ausgeführt wird.

Zur Verwendung einer Primärübertragung wie folgt vorgehen:

1. Den zu kopierenden oder zu verschiebenen Text auswählen.
2. Den Zeiger an die Position setzen, an der der ausgewählte Text eingefügt werden soll.
3. Zum Kopieren des Texts die mittlere Maustaste drücken. Zum Verschieben des Texts die Umschalttaste gedrückt halten und mit der mittleren Maustaste klicken.

Zum Verschieben oder Kopieren von Text mit Hilfe einer Schnellübertragung wie folgt vorgehen:

1. Den Zeiger an die Position setzen, an der der Text eingefügt werden soll.

2. Mit der linken Maustaste klicken.
3. Den Zeiger an die Position des Texts setzen, die verschoben oder kopiert werden soll.
4. Die Taste Alt gedrückt halten und die mittlere Maustaste drücken und gedrückt halten und den Zeiger über den zu übertragenden Text ziehen. Der Text in dem ausgewählten Bereich erscheint unterstrichen.
5. Die mittlere Maustaste freigeben, um den Text zu kopieren. Zum Verschieben des Texts die Taste Alt und die Umschalttaste drücken und gedrückt halten, während die Auswahl getroffen wird, und die mittlere Maustaste freigeben.

Eine laufende Schnellübertragungsoperation kann abgebrochen werden, indem vor dem Freigeben der mittleren Maustaste die Abbruchtaste gedrückt wird oder indem die mittlere Maustaste außerhalb der Textkomponente freigegeben wird.

Text kann anstelle der Maus auch über die Tastatur ausgewählt, kopiert und verschoben werden. Um eine Auswahl zu treffen, die Umschalttaste drücken und gedrückt halten und die verschiedenen Textnavigationstasten verwenden, um den auszuwählenden Text hervorzuheben. Die Umschalttaste freigeben, um die Auswahl zu beenden.

Nachdem die Auswahl getroffen wurde, wird durch weitere Verwendung der Navigationstasten die Auswahl für den Bereich zurückgenommen. Zum Kopieren oder Verschieben der Auswahl kann entweder wie zuvor beschrieben die Maus verwendet werden oder es kann die Tastenfolge Umsch-F8 verwendet werden, um den *Hinzufügemodus* zu aktivieren. Im Hinzufügemodus kann der Einfügekursor bewegt werden, ohne daß die Auswahl zurückgenommen wird. Nach dem Bewegen des Einfügekursors die Tastenfolge Alt-Strg-Einfügen verwenden, um die Auswahl zu kopieren. Zum Verschieben der Auswahl die Tastenfolge Alt-Umsch-Löschen verwenden.

Zum Löschen von Text innerhalb einer Textkomponente die Rückschrittaste und Löschtaste verwenden. Bei ausgewähltem Text wird die Auswahl durch beide Tasten gelöscht. Ist kein Text ausgewählt, wird durch die Rückschrittaste der Text vor dem Cursor und durch die Löschtaste das Zeichen nach dem Cursor gelöscht.

Benutzung der Tastatur

AIXwindows-Anwendungen, die mit dem AIXwindows Style Guide in Einklang stehen, müssen Interaktionsmethoden unterstützen, die mausunabhängig sind. Zum Zwecke der Navigation über die Tastatur werden Komponenten der Anwendung in Gruppen, sogenannten *Feldern*, verwaltet. Durch Drücken der Tabulatortaste kann der Eingabebereich von einem Feld zum nächsten verschoben werden. Mit der Tastenfolge Umsch-Tabulator kann der Eingabebereich in die umgekehrte Richtung verschoben werden. Mit den Pfeiltasten kann der Eingabebereich zwischen den Komponenten in einem Feld verschoben werden.

Für die Navigation in Menüs werden in AIXwindows-Anwendungen spezielle Tastenbelegungen verwendet, die nachfolgend zusammengefaßt sind:

- Die Taste F10 dient zum Umschalten von und zur Menüleiste.
- Mit der Menütaste oder der Tastenfolge Umsch-F10 werden Dialogfenstermenüs in Bereichen mit Dialogfenstermenüs geöffnet und abgebrochen.
- Mit der Leertaste und der Auswahl Taste werden Auswahlmenüs aktiviert.
- Die Pfeiltasten dienen zum Wechseln von Menüteilfenstern.
- Mit der Abbruchtaste kann die Menühierarchie verlassen werden.

In Anwendungen können auch Seitennavigationstasten verwendet werden, um einen Bereich um jeweils eine Seite zu verschieben. Unter AIXwindows Keyboard and Mouse

Operations in *AIXwindows and AIXwindows Desktop User's Guide* sind die Standardtastenbelegungen von AIXwindows aufgeführt.

Ziehen und übergeben

Die Funktion *ziehen und übergeben* ermöglicht es dem Benutzer, ein Objekt von einem Teil der Anzeige aufzunehmen und an eine andere Position zu verschieben. Das aufgenommene Objekt wird als *Ziehquelle* bezeichnet. Die Position, an der das Objekt übergeben wird, wird als *Übergabepunkt* bezeichnet.

Mit der Funktion "ziehen und übergeben" können folgende Tasks ausgeführt werden:

- Text oder andere Informationen zwischen Fenstern verschieben (ausschneiden und einfügen).
- Aktionen starten.
- Informationen zu Übergabepunkten erhalten.

Zum Beispiel kann der Inhalt einer Datei gedruckt werden, indem das Symbol für die Datei auf ein Symbol für einen Drucker gezogen und übergeben wird.

Quelle und Ziel des Übergebens können sich in derselben Anwendung oder in unterschiedlichen Anwendungen befinden. Textkomponenten, Marken und Listen sind Ziehquellen in allen AIXwindows-Anwendungen. Textkomponenten sind außerdem automatisch Übergabepunkte. Weitere Ziehquellen und Übergabepunkte werden durch die jeweiligen Anwendungen definiert.

Ziehvorgang beginnen

Zum Beginnen eines Ziehvorgangs wie folgt vorgehen:

1. Das zu ziehende Objekt wie folgt auswählen:
 - a. Bei Textkomponenten den zu ziehenden Text auswählen.
 - b. Bei Listen die zu ziehenden Einträge auswählen. Ein einzelner Eintrag kann ohne vorheriges Auswählen gezogen werden.
 - c. Andere Objekte können normalerweise ohne vorheriges Auswählen gezogen werden. Je nach Anwendung kann es jedoch erforderlich sein, einzelne Objekte vor dem Ziehen auszuwählen.
2. Den Zeiger auf das zu ziehende Objekt setzen. Dieses Objekt wird als Ziehquelle bezeichnet.
3. Die mittlere Maustaste drücken und gedrückt halten.
4. Durch Drücken von Kombinationstasten (siehe die folgende Tabelle) die Operation angeben, die nach der Übergabe erfolgen soll.

Taste	Aktion
Umsch	Verschieben; die Informationen werden vom Initiator zum Empfänger verschoben.
Strg	Kopieren; die Informationen werden vom Initiator zum Empfänger kopiert.
Umsch-Strg	Verbinden; die Informationen werden vom Initiator zum Empfänger verbunden.

Wird keine Kombinationstaste gedrückt, wird im allgemeinen eine Verschiebeoperation angenommen; die tatsächlich ausgewählte Operation hängt jedoch von der Quelle und dem Ziel ab. Die Anwendungen, die Quelle und Ziel steuern, geben möglicherweise an, daß nur einige der möglichen Operationen zulässig sind. Bei einer Verschiebeoperation werden die

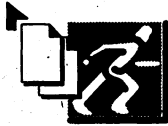
Informationen nach dem Kopieren an die neue Position an der ursprünglichen Position gelöscht.

Wählt das System eine vom Benutzer nicht gewünschte Operation aus, kann die Operation während des Ziehvorgangs durch Drücken der entsprechenden Kombinationstasten jederzeit durch eine neue Operation ersetzt werden.

Ein *Ziehsymbol*, das die Ziehquelle darstellt, wird anstelle des normalen Zeigers angezeigt. Das verwendete Ziehsymbol ist von der Ziehquelle abhängig. Das Ziehsymbol besteht aus den drei folgenden Komponenten:

- **Quellensymbol:** Das Quellensymbol ist eine grafische Darstellung des gezogenen Objekts.
- **Statussymbol:** Das Statussymbol stellt die Beziehung zwischen dem Ziehsymbol und dem aktuellen Hintergrund dar. Das Statussymbol ist unterschiedlich, je nachdem, ob sich das Ziehsymbol über einem gültigen Übergabepunkt, einem nicht unterstützten Übergabepunkt oder über keinem Übergabepunkt befindet.
- **Operationssymbol:** Das Operationssymbol stellt die Operation dar, die nach der Übergabe ausgeführt wird (Kopieren, Verschieben oder Verbinden). Wird eine Operation angefordert, die der Initiator oder der Empfänger nicht zulässt, ist das Symbol leer.

Die folgende Abbildung zeigt ein Ziehsymbol. Das Quellensymbol ist die laufende Figur, das Statussymbol ist der Pfeil in der linken oberen Ecke und das Operationssymbol ist das Feld unter dem Pfeil.

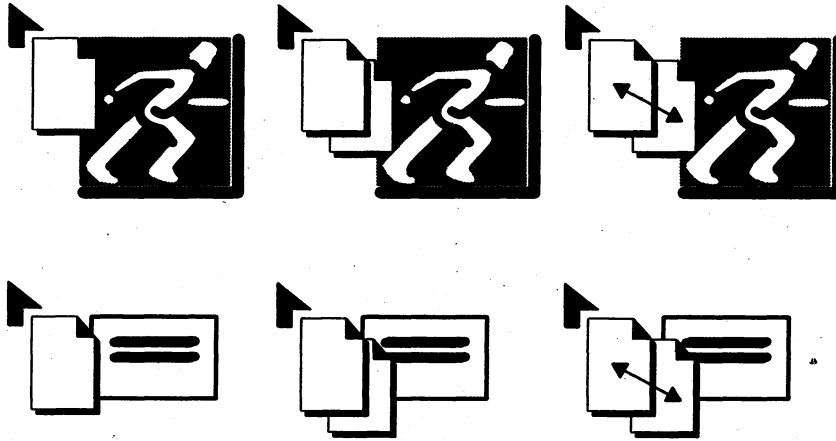


Ziehsymbol

AIXwindows definiert ein Standardquellensymbol für Textinformationen, das verwendet wird, wenn die Ziehquelle eine Listen-, Marken- oder Textkomponente ist. Ist die Quelle eine Grafikkomponente, versucht AIXwindows, das Grafikabbild als Quellensymbol zu verwenden. Ist die Quelle anderen Typs und ist von der Anwendung kein Quellensymbol für diesen Datentyp vorgesehen, wird die laufende Figur als Quellensymbol verwendet.

Status- und Operationssymbole sind in einem Ziehsymbol wahlfrei. AIXwindows stellt Standardoperationssymbole für Verschiebe-, Kopier- und Verbindungsoperationen zur Verfügung. Das Standardstatussymbol für alle drei Status ist ein Pfeil in der linken oberen Ecke des Ziehsymbols.

Die folgende Abbildung zeigt die Standardziehsymbole bei Grafik- und Textinformationen für Verschiebe-, Kopier- und Verbindungsoperationen.



Ziehsymbole

Eine Anwendung kann verschiedene Symbole für jede Operation und jeden Status angeben und die Symbole auf andere Weise als in der Abbildung dargestellt anordnen. Durch Festlegen von Ressourcen können ebenfalls andere Symbole angegeben werden. Unter *AIXwindows AIXwindows Keyboard and Mouse Operations in AIXwindows and AIXwindows Desktop User's Guide* befinden sich Informationen zum Angeben anderer Symbole durch Festlegen von Ressourcen.

Während des Ziehvorgangs

Die Maus verschieben, ohne dabei die Maustaste freizugeben. Das Ziehsymbol, das die Quelle darstellt, wird anstelle des normalen Zeigers über die Anzeige bewegt.

Jedes Ziehsymbol verfügt über einen *Detailpunkt*. Das gesamte Symbol gilt als innerhalb oder außerhalb des Übergabepunkts, wenn sich der Detailpunkt innerhalb oder außerhalb des Übergabepunkts befindet. Zeigt ein Ziehsymbol ein Statussymbol an, befindet sich der Detailpunkt darin. Andernfalls befindet sich der Detailpunkt im Quellsymbol. Der Detailpunkt im Standardstatussymbol befindet sich am Punkt des Pfeils.

Während das Ziehsymbol über die Anzeige bewegt wird, kann sich seine Farbe und Form ändern, um anzuzeigen, ob sich das Ziehsymbol über einem gültigen Übergabepunkt, einem nicht unterstützten Übergabepunkt oder über keinem Übergabepunkt befindet. Diese Änderungen werden als *Zieheffekte für Ziehsymbol* bezeichnet. Standardmäßig tritt keine erkennbare Änderung während des Verschiebens ein. Eine Anwendung kann jedoch andere Zieheffekte für Ziehsymbol bieten oder es können eigene Effekte mit Hilfe von Ressourcen angegeben werden.

Das Operationssymbol gibt an, welche Operation nach der Übergabe eintritt. Kombinationstasten können geändert werden, um die Operation während eines Ziehvorgangs jederzeit umzuschalten. Das Operationssymbol zeigt die neue Operation durch die Änderung an. Wurde keine Operation mit Hilfe der Kombinationstasten angegeben, ist die Standardoperation vom Übergabepunkt abhängig, über dem sich das Ziehsymbol befindet.

Die Darstellung eines Übergabepunkts kann sich außerdem ändern, wenn das Ziehsymbol hinein- und hinausbewegt wird. Diese Änderungen werden als *Zieheffekte für Übergabepunkt* bezeichnet. Standardmäßig wird als Zieheffekt für Übergabepunkt bei einem gültigen Übergabepunkt ein Rahmen um den Übergabepunkt angezeigt. Andere Zieheffekte für Übergabepunkt sind von der Anwendung abhängig, die den Übergabepunkt steuert.

Objekt an einem Ziel übergeben

Zum Übergeben eines gezogenen Objekts an einem Ziel wie folgt vorgehen:

1. Den Zeiger über einen Übergabepunkt stellen.
2. Die mittlere Maustaste freigeben. Dabei Kombinationstasten, die zum Festlegen der Operation verwendet werden, weiterhin gedrückt halten.

Ist der Übergabepunkt für diese Ziehquelle gültig, wird die Übergabe dadurch angezeigt, daß das Ziehsymbol mit der Zielposition verschmilzt. Ist der Übergabepunkt für diese Ziehquelle nicht gültig, springt das Ziehsymbol an den Punkt zurück, von dem aus der Ziehvorgang eingeleitet wurde.

Ist die Ziehquelle eine Marke, eine Liste oder eine Textkomponente, sind die gezogenen Daten Textinformationen, die auf jedem Textziel übergeben werden können. Andere Kombinationen von Quellentyp und Zieltyp werden durch die Anwendung definiert.

Enthält das Ziel Daten oder Informationen, werden die gezogenen Informationen im allgemeinen an der Zeigerposition eingefügt. Stellt das Ziel eine Anwendung oder eine Einheit dar, wird durch die Übergabe im allgemeinen eine Aktion gestartet. Die Art der Aktion ist von der Anwendung abhängig. Wird beispielsweise ein Symbol für eine Datei auf ein Symbol für einen Drucker gezogen, wird der Inhalt der Datei auf dem durch das Symbol dargestellten Drucker gedruckt.

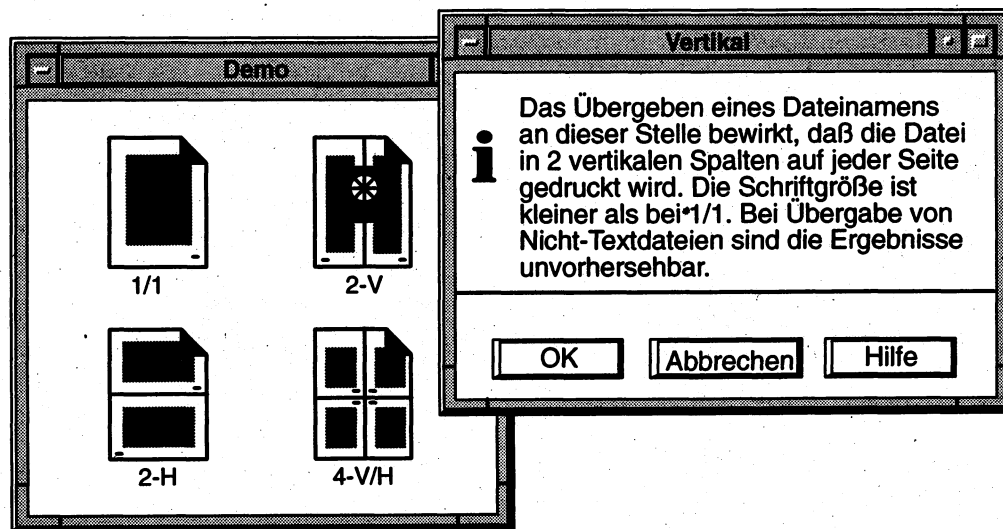
Hilfe anfordern

Wenn die Anwendung Hilfetexte bereitstellt, kann Hilfe zu einem bestimmten Übergabepunkt angefordert werden. Der Hilfetext erläutert normalerweise die von einem Übergabepunkt erwarteten Quellen und die Aktionen, die bei der Übergabe eintreten.

Zum Anfordern von Hilfe wie folgt vorgehen:

1. Den Zeiger über einen Übergabepunkt stellen.
2. Die Hilfetaste oder die Taste F1 drücken.

Die folgende Abbildung zeigt ein Beispiel für einen Hilfetext.



Hilfetext

Ziehvorgang abbrechen

Ein Ziehvorgang kann jederzeit abgebrochen werden. Zum Abbrechen eines laufenden Ziehvorgangs die Abbruchtaste drücken. Das Ziehsymbol wird durch die normale Zeigerform ersetzt und alle speziellen Zieheffekte für Ziehsymbol und Übergabepunkt werden beendet.

Empfohlene Dokumentation

Grundlegende Informationen

Kapitel 1. Lerntext: Erste Schritte

Kapitel 2. Grundlagen des Betriebssystems

Kapitel 3. Dateien, Verzeichnisse und allgemeine Aufgaben

Zugehörige Informationen

Kapitel 5. AIXwindows Desktop: Grundlagen

Kapitel 6. Netze und TCP/IP

Kapitel 7. Post

Kapitel 8. Online-Informationen und Hilfetext

Kapitel 9. Editoren und Textformatierung

Kapitel 10. Problemlösung und Hardwareinformation

Anhang A. Korn-Shell

Anhang B. Bourne-Shell

Anhang C. C-Shell

Kapitel 5. AIXwindows Desktop

AIXwindows Desktop ist eine objektorientierte Benutzerschnittstelle, die mit Fenstern arbeitet. Dieses Programm stellt eine einfache und effiziente Möglichkeit zum Arbeiten mit der Datenstation dar. Für Benutzer mit unterschiedlichen Kenntnisstufen ist das visuelle System von AIXwindows eine gute Möglichkeit, die Arbeit zu verwalten und mit dem Betriebssystem diverse Aufgaben auszuführen.

Das AIXwindows Desktop zeigt die Arbeitsumgebung des Benutzers an. Beim Starten von AIXwindows Desktop, wird das *Standard-Arbeitsoberflächenfenster* des Benutzers angezeigt. Dieses Arbeitsoberflächenfenster enthält Hilfsprogramme und einen allgemeinen Bereich, in dem mit Objekten aus unterschiedlichen Systembereichen gearbeitet werden kann. Benötigte Dateien und Hilfsprogramme können aus beliebigen Positionen im Dateisystem verwendet werden, vorausgesetzt, der Dateieigner hat hierzu die erforderliche Berechtigung vergeben.

Es können mehrere Arbeitsoberflächenfenster mit zusätzlichen Arbeitsbereichen geöffnet werden, welche dem Benutzer das Arbeiten mit Objekten aus unterschiedlichen Systembereichen ermöglichen.

Verzeichnisfenster sind AIXwindows-Fenster mit Symbolen, die auf dem System vorhandene Verzeichnisse darstellen.

Der *Mauszeiger* oder Cursor stellt die Maus auf der Anzeige dar. Er verändert sich, wenn er in andere Bereiche der Anzeige bewegt wird. Mit dem Mauszeiger können Symbole ausgewählt, Fenster verwaltet und Menüpunkte ausgewählt werden.

AIXwindows Desktop-*Menüs* und -*Symbole* erleichtern den Zugriff auf viele Befehle, die beim Arbeiten mit AIXwindows Desktop benötigt werden.

Symbole stellen die Objekte (Dateien, Programme, Anwendungen, Verzeichnisse oder Einheiten) dar, mit denen der Benutzer auf der Arbeitsoberfläche arbeitet. Zum Verwalten des Dateisystems können Symbole mit einer Maus ausgewählt und bewegt werden. Menüoptionen können auch mit einer Maus ausgewählt werden.

In *Menüs* werden Befehle aufgelistet, mit denen Funktionen an Objekten in Arbeitsoberflächen- und Verzeichnisfenstern ausgeführt werden. In der *Menüleiste* werden Menütitel, die in Arbeitsoberflächen- und Verzeichnisfenstern verwendet werden, aufgelistet.

Folgende Abschnitte enthalten ausführliche Informationen zu den Grundfunktionen und wichtigsten Komponenten von AIXwindows Desktop.

- AIXwindows Desktop-Merkmale auf Seite 5-2.
- AIXwindows Desktop starten auf Seite 5-5.
- AIXwindows Desktop--Anzeigekomponenten auf Seite 5-7.
- Mit der Maus bei AIXwindows Desktop arbeiten auf Seite 5-9.
- AIXwindows Desktop-Symbole auf Seite 5-12.
- AIXwindows Desktop-Menüs auf Seite 5-24.
- Tastenäquivalente für Menüoptionen bei AIXwindows Desktop auf Seite 5-34.
- Arbeitsoberflächentypen bei AIXwindows Desktop auf Seite 5-38.

AIXwindows Desktop-Merkmale

Für unerfahrene Benutzer wird das Arbeiten mit dem Betriebssystem durch AIXwindows Desktop erleichtert. Außerdem bietet es einige Merkmale, die das Bearbeiten von Text und Programmen wesentlich verbessern. Diese Merkmale werden in den folgenden Abschnitten beschrieben.

Anzeigekomponenten

AIXwindows Desktop-Anzeigekomponenten stellen das Datenverarbeitungssystem dar. Sie bieten dem Benutzer die Möglichkeit, ihre Objekte in geordneter und logisch strukturierter Weise zu bearbeiten. Mit der **Maus** kann der Benutzer Programme ausführen und seine Arbeitsumgebung organisieren, ohne hierzu Betriebssystembefehle oder Dateinamen eingeben zu müssen.

Symbole

Die Verzeichnisfenster von AIXwindows Desktop stellen eine optisch ansprechende Möglichkeit zum Anzeigen der Dateien zur Verfügung. In Ihnen werden alle Verzeichnisse und Dateien bildhaft, als sogenannte **Symbole** dargestellt. Symbole informieren auf einen Blick über die Objekte, die sie darstellen. AIXwindows Desktop enthält einen Satz großer und kleiner Symbole, die in Farbe oder monochrom angezeigt werden können. Die Standardsymbole sind kleine, farbige Symbole. So können in einem Fenster mehrere Dateien angezeigt werden.

Symbolmenüs

Zu AIXwindows Desktop-Symbolen gehören auch **Symbolmenüs**, die nützliche Optionen zur Verfügung stellen. Mit Hilfe dieser Menüs kann der Benutzer auf der Arbeitsoberfläche in einem Schritt ein Symbol auswählen und eine gewünschte Aktion anfordern.

Andere Arbeitsoberflächentypen

Von AIXwindows Desktop werden drei unterschiedliche Standard-Arbeitsoberflächen als benutzerfreundliche Arbeitsumgebungen für Benutzer mit unterschiedlichen Betriebssystemkenntnissen zur Verfügung gestellt. Das **General Desktop** ist an die Anforderungen von Benutzern mit wenig oder keinen Kenntnissen des Betriebssystems angepaßt. Das **Power Desktop** stellt für Benutzer mit guten Kenntnissen des Betriebssystems einfache Zugriffsmöglichkeiten auf Betriebssystembefehle und das Dateisystem zur Verfügung. Das **Admin Desktop** ermöglicht Systemadministratoren das Installieren, Konfigurieren und Verwalten des Systems.

Leichte Verwaltung von Arbeitsoberflächen und Dateien

Bei AIXwindows Desktop kann die Maus zum Ausführen von Programmen, zum Editieren, Kopieren und Löschen von Dateien verwendet werden. Alle Arbeitsoberflächen- und Verzeichnisfenster enthalten **Arbeitsoberflächenmenüs** und **Verzeichnismenüs**, mit denen direkter Zugriff auf die beim Arbeiten mit der Arbeitsoberfläche nützlichsten Befehle besteht.

Mehrere Arbeitsoberflächen

AIXwindows Desktop ermöglicht dem Benutzer das Erstellen mehrere Arbeitsoberflächenumgebungen, die der Benutzer optimal an seine Anforderungen anpassen kann. Er kann z. B. alle Hilfsprogramme für die Programmierung auf eine Arbeitsoberfläche und alle Hilfsprogramme zur Textbearbeitung auf eine andere Arbeitsoberfläche stellen und leicht zwischen diesen Arbeitsoberflächen hin- und herschalten.

Mehrere Arbeitsoberflächen sind besonders nützlich bei der Unterstützung spezialisierter Anwendungen oder besonderer Anforderungen von Benutzern.

Hilfsprogramme zum interaktiven Anpassen von AIXwindows Desktop

Für besondere Anforderungen individueller Benutzer können viele Merkmale des Erscheinungsbildes und der Funktionalität von AIXwindows Desktop mit dem **Profileditor** geändert werden. Dieses Hilfsprogramm ermöglicht den Zugriff auf diverse interaktive Menüs, mit denen verschiedene Merkmale der Arbeitsoberfläche geändert werden können. Benutzer mit Programmiererfahrung können die Regeln zum Steuern von Symbolen ändern oder neue Symbole zum Ausführen neuer Funktionen hinzufügen.

Es können auch neue Symbole entworfen werden. Hierzu den **Symboleditor** verwenden. Mit diesem Hilfsprogramm können bestehende Pixel-Abbild-Dateien editiert und neue Symbole erstellt werden.

Hilfsprogramme für das Hinzufügen von Programmen und das interaktive Editieren von Objekten

Benutzerfreundliche interaktive Hilfsprogramme zum Hinzufügen von Programmen und Editieren von Objekten werden ebenso über die Arbeitsoberfläche zur Verfügung gestellt. Diese Programme werden durch die Symbole **Programm hinzufügen** und **Objekt editieren** dargestellt. Mit diesen Hilfsprogrammen können Programme und Anwendungen in AIXwindows Desktop integriert werden.

Diese Hilfsprogramme ermöglichen durch eine vollständige Integration in die Arbeitsoberfläche die Optimierung der Flexibilität und der Einsetzbarkeit der Software. Sie ermöglichen die Steuerung und Verwaltung der Benutzerschnittstenumgebung und der Rechneraktivitäten ohne die Eingabe komplexer Befehle.

Arbeitsoberflächenprogrammierung

Ein herausragendes Merkmal von AIXwindows Desktop ist die Tatsache, daß Funktionalität und Erscheinungsbild des Programms nicht festgelegt sind, sondern durch Gruppen von **Regeldateien** definiert werden, die beschreiben, wie das Desktop und die Objekte auf der Arbeitsoberfläche aussehen und funktionieren. Diese Aktionen sind in der (Skript-basierten) Programmiersprache **Deskshell** geschrieben. **Resources** legen das standardmäßige Erscheinungsbild der Hauptkomponenten der Arbeitsoberfläche fest.

Dienstprogrammdateien sind AIXwindows Desktop-Programme, die über die Befehlszeile oder eine Shellprozedur ausgeführt werden können.

Standardaktion bei Objekten

Ein weiteres herausragendes Merkmal von AIXwindows Desktop ist das Aktivieren der Standardaktion durch Doppelklicken. Durch Doppelklicken mit der Maus kann der Benutzer die Standardaktion an einem Objekt ausführen. Z. B. wird bei der AIXwindows Desktop-Standardkonfiguration normalerweise der Editor **vi** gestartet, wenn auf einer Datendatei doppelt geklickt wird. Durch Standardaktionen kann die Anwendung auch in der Standardumgebung ausgeführt, der Einheitenstatus abgerufen oder eine Anwendung ausgeführt werden.

Automatische Aktivierung von Datendateien

Bei AIXwindows Desktop können Datendateien auch bestimmten Anwendungen zugeordnet werden. Durch Klicken auf einer Datei, die mit dem Editor **xedit** erstellt wurde, kann z. B. AIXwindows Desktop angewiesen werden, zum Editieren der Datei **xedit** automatisch zu starten, anstatt den Editor **vi** aufzurufen.

AIX stellt einen Mechanismus zum Zuordnen zusätzlicher Attribute (z. B. das Programm, durch das sie erstellt wurde) zu einer Datei zur Verfügung. Es ist auch möglich, daß

verschiedene Benutzer unterschiedliche Hilfsprogramme für dieselben Datendateien verwenden wollen und häufig auf dieselben Daten zugreifen müssen. Hat eine Datei z. B. die Erweiterung `.c`, kann eine Maustaste zum Editieren und eine andere Maustaste zum Kompilieren der Datei verwendet werden.

AIXwindows Desktop verfügt über eine Reihe von Vorgehensweisen zum Zuordnen bestimmter Aktionen zu Datendateien. Spezifische Standardaktionen für Dateien können abhängig von ihrem Typ, ihrer Position im Dateiablagensystem, ihrem Namen oder auf individueller Basis festgelegt werden.

Kommunikation mit dem AIXwindows Desktop

Über eine Gruppe von Arbeitsoberflächenbefehlen können andere Anwendungen AIXwindows Desktop anweisen, die erforderlichen Symbole auf die Arbeitsoberfläche zu stellen oder bestimmte Funktionen auszuführen. Mit dem Dienstprogramm `tellxdt3`, mit dem Arbeitsoberflächenbefehle von einem Programm oder einer Shell-Prozedur gesendet werden können, haben andere Anwendungen die Möglichkeit, AIXwindows Desktop zu steuern.

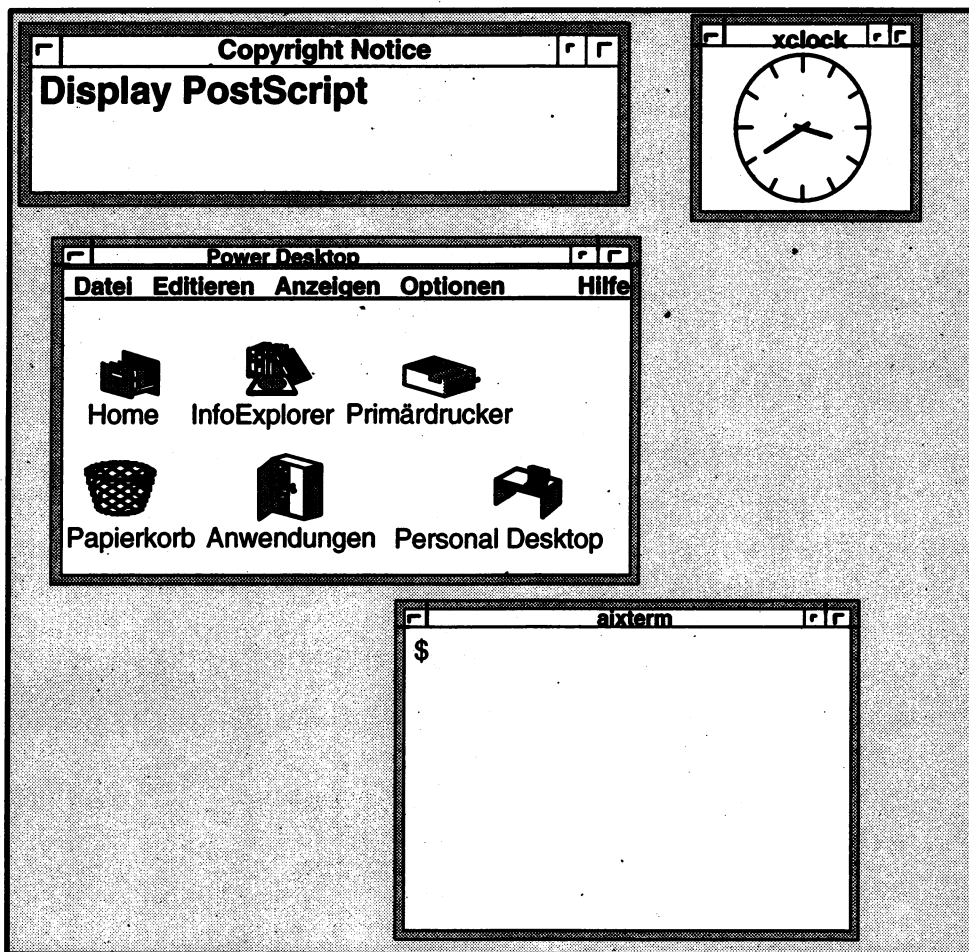
AIXwindows Desktop starten

Beim Starten von AIXwindows Desktop wird auf der Anzeige ein Arbeitsoberflächenfenster mit der Standardarbeitsoberfläche gestartet. Von diesem Arbeitsoberflächenfenster aus können mehrere Arbeitsoberflächen gestartet werden. Das Hauptarbeitsoberflächenfenster ist das erste Fenster, das geöffnet und das letzte Fenster, das geschlossen wird.

Beim Anmelden am System erfolgt das Starten je nach Systemkonfiguration in der Shell, in AIXwindows oder in AIXwindows Desktop.

Standardmäßig wird AIXwindows Desktop automatisch aufgerufen, wenn `xinit` eingegeben und die Eingabetaste gedrückt wird.

Folgende Abbildung zeigt die Arbeitsoberfläche nach dem Start.



Starten der Arbeitsoberfläche

AIXwindows Desktop in der Anmelde-Shell starten

In der Befehlszeile `xinit` eingeben und die Eingabetaste drücken.

AIXwindows wird aufgerufen und es wird ein Fenster mit einer Uhr, einem Startfenster mit dem AIXwindows Desktop-Copyrightvermerk angezeigt. Diese Nachricht wird automatisch wieder geschlossen und die Arbeitsoberfläche wird angezeigt.

Die Startprozedur für AIXwindows Desktop ändern

Die Standardwerte können so geändert werden, daß AIXwindows Desktop nur nach Eingabe eines bestimmten Befehls gestartet wird, oder daß es automatisch gestartet wird. Mit den folgenden Schritten können die Standardwerte geändert werden, so daß der Ablauf des Befehls `xinit` modifiziert wird.

Voraussetzungen

Im **Benutzerverzeichnis** muß die Datei `.xinitrc` vorhanden sein. Ist diese Datei nicht vorhanden, die Datei `xinitrc` aus dem Verzeichnis `/usr/lpp/X11/defaults/xinitrc` in das Benutzerverzeichnis kopieren und dieses in `.xinitrc` umbenennen.

AIXwindows Desktop automatisch starten

Die Datei `.xinitrc` im **Benutzerverzeichnis** editieren und das Pfundzeichen (#) vor der Zeile `/usr/lpp/X11/xdt3 &` entfernen. Das Pfundzeichen verhindert, daß diese Zeile vom System gelesen wird.

Zum Starten von AIXwindows Desktop in der Befehlszeile `xinit` eingeben und die Eingabetaste drücken.

AIXwindows Desktop mit einem Befehl starten

Die Datei `.xinitrc` im **Benutzerverzeichnis** editieren und ein Pfundzeichen (#) am Anfang der Zeile `/usr/lpp/X11/xdt3 &` in der Datei `.xinitrc` einfügen. Das Pfundzeichen verhindert, daß diese Zeile vom System gelesen wird.

Zum Starten von AIXwindows Desktop in der Befehlszeile `xdt3&` eingeben und die Eingabetaste drücken.

AIXwindows Desktop verlassen

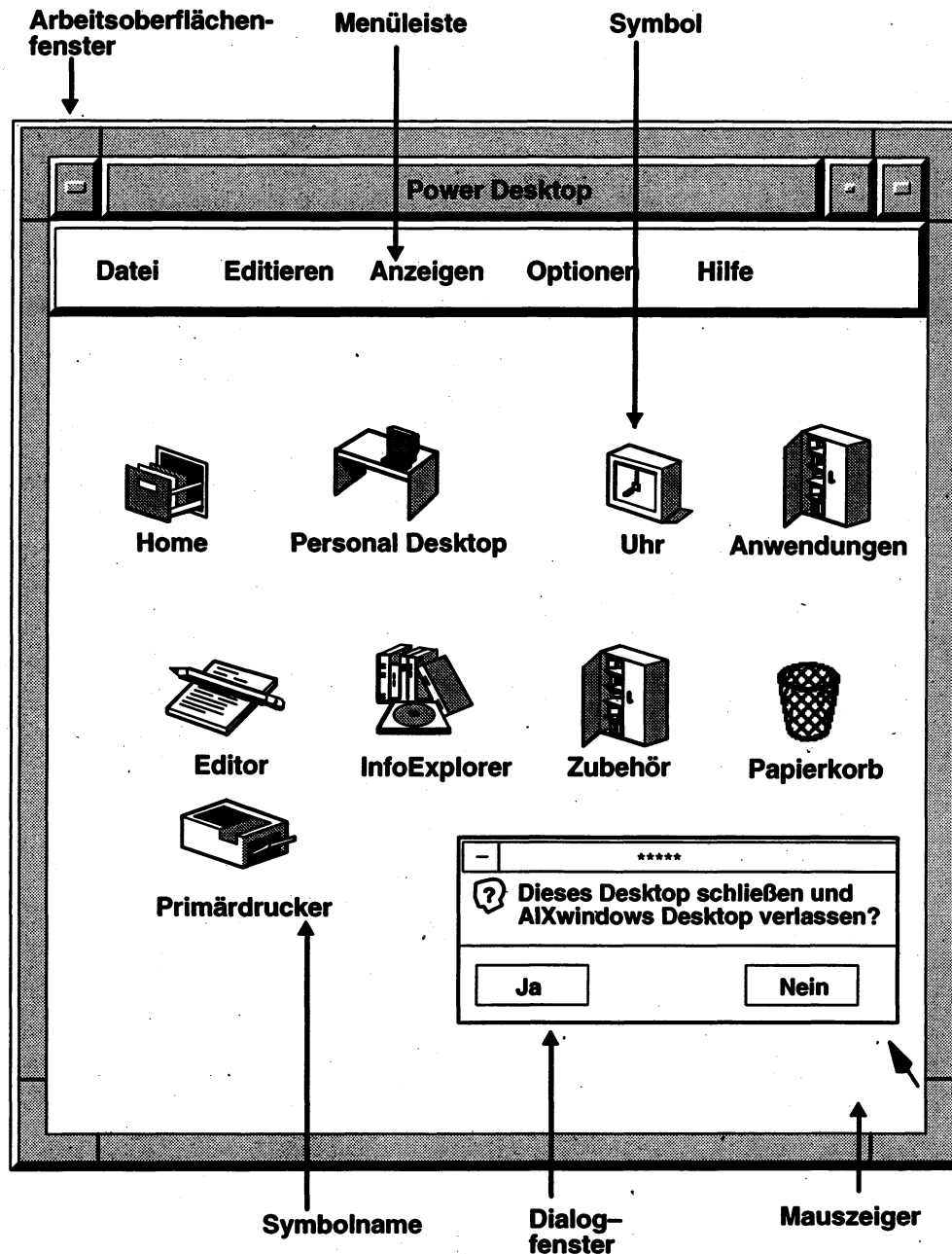
1. Aus der Arbeitsoberflächenmenüleiste **Datei** —> **Dieses Desktop schließen** auswählen.
2. Fordert das System zur Bestätigung auf, mit der linken Maustaste den Knopf **Ja** anklicken.

AIXwindows Desktop-Anzeigekomponenten

Die Arbeitsoberfläche ist der Hauptarbeitsbereich des Benutzers, auf dem die verfügbaren Objekte angezeigt werden. Auf der Arbeitsoberfläche kann der Benutzer über die Maus mit Dokument- und Anwendungssymbolen arbeiten. Außerdem können Befehle mit der Maus in Menüs ausgewählt und gestartet werden.

Arbeitsoberflächenfenster

Im Arbeitsoberflächenfenster wird der Arbeitsbereich des Benutzers angezeigt.



Arbeitsoberflächenkomponenten

Symbole



Uhr

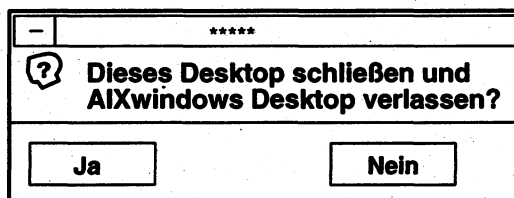
Symbole sind Bilder, die die Objekte auf der Arbeitsoberfläche darstellen. Objekte können Programme, Anwendungen, Dateien und Verzeichnisse sein.

Mauszeiger



Mit dem Mauszeiger werden Objekte, Menüs und Fenster bearbeitet.

Dialoge



Dialogfenster sind Fenster, die Informationen enthalten oder Informationen vom Benutzer anfordern. Das Startfenster wird beim Starten von AIXwindows Desktop angezeigt und automatisch nach einigen Sekunden wieder geschlossen.

Menüleiste



Die Menüleiste enthält die Titel der Arbeitsoberflächenmenüs. Jedes Menü enthält eine Liste der verfügbaren Optionen.

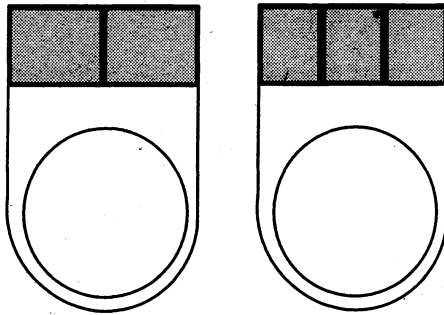
Mit der Maus bei AIXwindows Desktop arbeiten

Mit der Maus kann der Zeiger schnell in alle Bereiche der Bildschirmanzeige bewegt werden. Mit der Maus können Aktionsanweisungen an AIXwindows Desktop übergeben werden. Dies geschieht durch Bearbeiten von Symbolen, Menüs und Fenstern.

Drei- oder Zweitastenmaus verwenden

Für AIXwindows Desktop kann eine Maus mit zwei oder mit drei Tasten verwendet werden.

Hinweis: Für Linkshänder kann die Belegung der Maustasten mit dem **Profileditor** umgekehrt werden.



Maus mit zwei bzw. drei Tasten

Dreitastenmaus

Die am häufigsten mit AIX verwendete Maus hat drei Tasten. Mit jeder Taste wird eine andere Funktion ausgeführt.

Linke Maustaste

Die linke Maustaste zum Auswählen, Kopieren und Umbenennen von Objekten, sowie zum Aktivieren der Standardaktionen für Objekte verwenden.

Mittlere Maustaste

Die mittlere Maustaste zum Umkehren der Auswahl und Versetzen von Objekten sowie zum Aktivieren der alternativen Aktionen für Objekte (falls verfügbar) verwenden.

Rechte Maustaste

Die rechte Maustaste ausschließlich für die Dialogfenstermenüs der Symbole verwenden.

Zweitastenmaus

Bei einer Zweitastenmaus entspricht das gleichzeitige Drücken beider Tasten dem Drücken der mittleren Taste bei der Dreitastenmaus.

Linke Maustaste

Die linke Maustaste zum Auswählen, Kopieren und Umbenennen von Objekten sowie zum Aktivieren der Standardaktionen für Objekte verwenden.

Beide Maustasten

Beide Maustasten zum Umkehren der Auswahl und Versetzen von Objekten, sowie zum Aktivieren der alternativen Aktionen für Objekte verwenden.

Rechte Maustaste

Die rechte Maustaste ausschließlich für die Dialogfenstermenüs der Symbole verwenden.

Basisaktionen der Maustasten

Der folgende Abschnitt enthält Informationen über die mit den Maustasten ausgeführten Aktionen.

Zeigen

Zum Zeigen auf ein Objekt die Maus bewegen, bis sich die Spitze des Mauszeigers auf dem Objekt (Symbol, Menü, Fenster oder Fensterauswahl) befindet. Zum Ausführen der Befehle zunächst auf das Objekt zeigen und dann die Maustasten drücken und gedrückt halten, einfach klicken, doppelt klicken, auf das Objekt ziehen oder auf das Objekt ziehen und an dieses übergeben.

Drücken und Halten

Zum Drücken und Halten auf das Objekt (Symbol, Menü oder Fenster) zeigen und dann die Maustaste drücken halten, ohne die Maus zu bewegen.

Klicken

Zum Klicken auf das Objekt (Symbol, Menü, Fenster oder Fensterauswahl) zeigen und dann die Maustaste drücken und sofort wieder freigeben, ohne die Maus zu bewegen.

Doppelt klicken

Zum Doppelklicken auf das Objekt (Symbol) zeigen und dann die Maustaste schnell zweimal drücken, ohne die Maus zu bewegen.

Ziehen

Ziehen bezieht sich normalerweise auf das Verschieben von Fenstern und die Auswahl einer Menüoption.

Zum Ziehen einer Menüauswahl auf das gewünschte Menü zeigen und die Maustaste drücken und gedrückt halten. Den Zeiger verschieben (ziehen), um die gewünschte Menüoption hervorzuheben, und dann die Maustaste freigeben.

Zum Ziehen eines Fensters an eine andere Position auf eine beliebige Stelle des Fensterrahmens zeigen und die Maustaste drücken und gedrückt halten. Den Zeiger an die neue Position verschieben (ziehen) und dann die Maustaste freigeben.

Ziehen und Übergeben

Ziehen und übergeben bezieht sich normalerweise auf Symbole.

Zum Ziehen und Übergeben auf das Symbol zeigen und die Maustaste drücken und gedrückt halten. Dann mit weiterhin gedrückter Maustaste den Zeiger in die Richtung bewegen (ziehen), in die das Objekt versetzt werden soll. Befindet sich das Objekt an der gewünschten Stelle, die Maustaste freigeben (damit wird das Symbol übergeben).

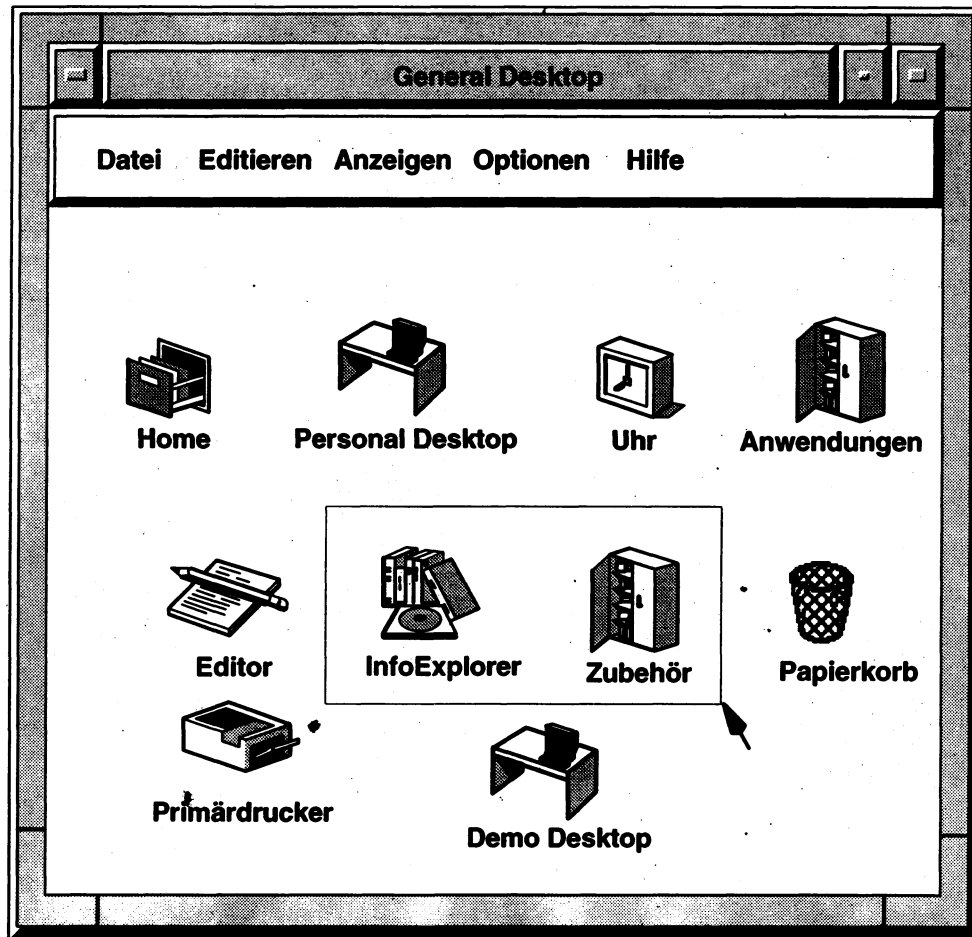
Einpassen

Das Einpassen wird gewöhnlich zum *Umschalten* (Auswählen oder Zurücknehmen der Auswahl) des Auswahlstatus eines Symbols verwendet. Diese Funktion dient auch zum Bearbeiten einer Gruppe von Symbolen. Durch das Umschalten wird ein ausgewähltes Symbol abgewählt bzw. ein nicht ausgewähltes Symbol ausgewählt.

Zum Einpassen auf eine Stelle in der Nähe des/r ausgewählten Symbols/e zeigen, deren Status umgeschaltet werden soll, und (ohne einen Teil des Symbols oder den Symbolnamen zu berühren) die linke Maustaste drücken und gedrückt halten und den Zeiger ziehen. Ein Kästchen wird angezeigt, dessen Rahmen gedehnt werden kann, bis er die gewünschten Symbole umfaßt. Wird die Maustaste losgelassen, wird die Auswahl

aller Symbole innerhalb des dehnbaren Rahmens, bzw. aller Symbole, die diesen berühren, umgeschaltet.

Folgende Abbildung zeigt das Einpassen.



Beispiel für Einpassen

AIXwindows Desktop-Symbole

AIXwindows Desktop verwendet benannte Bilder zur Darstellung von Programmen, Anwendungen, Dateien, Arbeitsoberflächen und Verzeichnissen im Dateisystem. Diese Bilder (sogenannte Symbole) geben dem Benutzer auf einen Blick wichtige Informationen über die Dateien, die sie darstellen. Der Text unterhalb eines Symbols ist der zugehörige Name.

AIXwindows Desktop enthält je einen Satz großer und kleiner Symbole, die monochrom oder in Farbe angezeigt werden können. Als Standardsymbole werden die kleinen Symbole (in Farbe) verwendet, um so mehr Dateien in einem Fenster anzeigen zu können.

Zu den AIXwindows Desktop-Symbolen gehören auch Symbolmenüs mit nützliche Optionen. Die Symbolmenüs erleichtern die Arbeit mit der Arbeitsoberfläche durch die Möglichkeit, in einem Schritt ein Symbol auszuwählen und für dieses eine gewünschte Aktion anzugeben.

Hinweis: Falls vorhanden, werden im folgenden große und kleine Symbole dargestellt.

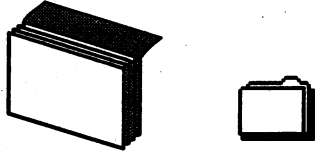
Folgende Abschnitte enthalten weitere Informationen zu AIXwindows Desktop-Symbolen:

- Symboltypen auf Seite 5-13
- Zugriffsmerkmale für Symbole auf Seite 5-15
- Symbolmenüs auf Seite 5-16.
- Symbole auswählen und die Auswahl zurücknehmen auf Seite 5-18.
- Symbole aktivieren auf Seite 5-20.
- Symbole verschieben und kopieren auf Seite 5-21.
- Symbole umbenennen auf Seite 5-22.
- Symbole löschen auf Seite 5-23.

Symboltypen

Eine Reihe von Symbolen werden standardmäßig auf der Arbeitsoberfläche angezeigt. Hierzu gehören das Benutzerverzeichnis (**Home**), einer oder mehrere Zubehörschränke, ein Papierkorb und InfoExplorer.

Verzeichnissymbole



Verzeichnissymbole sind wie Schubladen in einem Schreibtisch und Ordner in diesen Schubladen. Sie enthalten Dokumente, Anwendungen und andere Verzeichnisse. Dieses Symbol wird als Ablageordner angezeigt.

Benutzerverzeichnissymbole



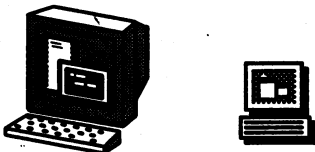
Das Benutzerverzeichnis (Home) entspricht der Hauptablageschublade. Es enthält alle Dateien und Verzeichnisse eines Benutzers. Dieses Verzeichnis wird als Ablagefach dargestellt.

Symbole für Anwendungen und Zubehör



Die Verzeichnisse "Anwendungen" und "Zubehör" enthalten für den Benutzer nützliche Programme. Dieses Symbol wird als Schrank dargestellt.

Anwendungs-/Programmsymbole



Anwendungssymbole stellen Programmdateien dar, die auf der Datenstation ausgeführt werden können. Für viele der Standardanwendungen gibt es spezielle Symbole. Anwendungen ohne spezielles Symbol werden mit diesem generischen Symbol eines Computerbildschirms dargestellt.

Dokument-/Dateisymbole



Dokumentsymbole stellen Dateien dar, die vom Benutzer erstellt wurden. Einige Dokumente werden mit besonderen Symbolen dargestellt, die anzeigen, mit welcher Anwendung sie erstellt wurden. Dieses Symbol wird als leeres Blatt Papier dargestellt.

Spezielle Symbole

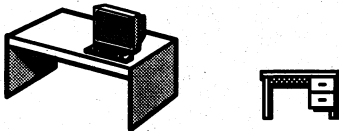


Einige Symbole stehen für Einheiten, von denen spezielle Funktionen oder Aktionen ausgeführt werden. Der **Papierkorb** ist z. B. ein spezielles Symbol, mit dem Dateien und Verzeichnisse entfernt werden können.



Das Symbol **Getting Started** steht für ein interaktives Lernprogramm, das einen Überblick über AIXwindows Desktop enthält. Dieses Programm wird beim ersten Starten von AIXwindows Desktop automatisch gestartet.

Arbeitsoberflächensymbole



Zusätzlich zur Hauptarbeitsoberfläche stellt AIXwindows Desktop dem Benutzer eine persönliche Arbeitsoberfläche zur Verfügung, die durch Doppelklicken auf dem Symbol **Personal Desktop** aufgerufen werden kann. Auf dieser Arbeitsoberfläche können zusammengehörende Programme und Dokumente zusammengefaßt werden. Dieses Symbol wird als Schreibtisch dargestellt.

Geistsymbole



Geistsymbole stehen für Dateien, die vom Desktop entfernt wurden. Dieses Symbol wird als Geist dargestellt.

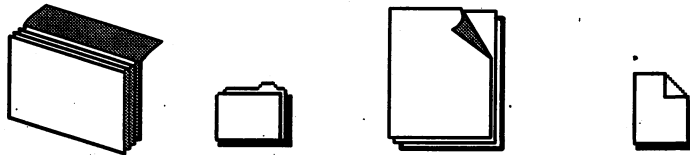
Zugriffsmerkmale für Symbole

Symbole, die Verzeichnisse, Dokumente und Anwendungen darstellen, zeigen auch die Zugriffsmerkmale des Symbols an.

Schreib-/Lesezugriff

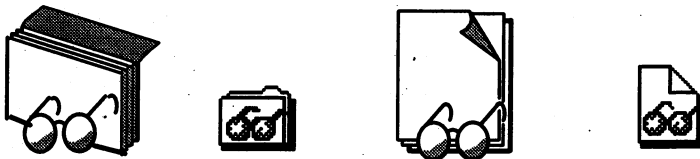
Symbole geben auch den Schreib-/Lesezugriff auf Dokumente und Verzeichnisse an.

Verzeichnisse und Dokumente, die gelesen werden können bzw. in die geschrieben werden kann:



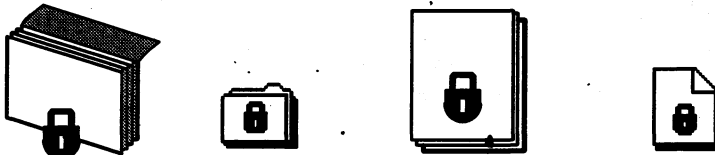
Diese Symbole werden normal angezeigt.

Verzeichnisse und Dokumente, die nur gelesen werden können:



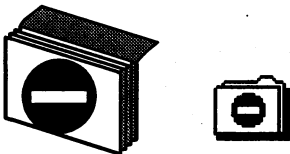
Eine auf der Datei angezeigte Brille zeigt an, daß die Datei gelesen werden kann, daß aber nicht in die Datei geschrieben werden kann.

Verzeichnisse und Dokumente, die nicht gelesen werden können bzw. in die nicht geschrieben werden kann:



Ein Vorhängeschloß auf einem Dateisymbol zeigt an, daß die Datei nicht geöffnet werden kann. Ein Vorhängeschloß auf einem Verzeichnissymbol zeigt an, daß der Inhalt des Verzeichnisses angezeigt werden kann, in die darin enthaltenen Dateien jedoch nicht geschrieben werden kann bzw. die Dateien nicht gelesen werden können.

Verzeichnisse, auf die kein Zugriff besteht:



Das Symbol "Einfahrt verboten" auf einem Verzeichnissymbol zeigt an, daß der Benutzer den Inhalt des Verzeichnisses nicht anzeigen kann.

Ausführungsberechtigung

Bei Anwendungssymbolen wird außerdem angezeigt, ob die Berechtigung zum Ausführen des Programms vorliegt. Wird auf dem Anwendungssymbol ein Vorhängeschloß angezeigt, liegt keine Ausführungsberechtigung für das Programm vor.

Symbolmenüs

Symbolmenüs bieten eine benutzerfreundliche und effiziente Möglichkeit, mit der Arbeitsoberfläche zu arbeiten. Menüoptionen, die unter den jeweils gegebenen Bedingungen ausgeführt werden können, werden normal angezeigt. Kann eine Menüoption nicht ausgeführt werden, wird sie abgeblendet angezeigt und AIXwindows Desktop ignoriert Mausaktionen für diese Menüoption.

AIXwindows Desktop-Symbolmenüs verfügen außerdem über die Option **Hilfe**, über die Anweisungen zum Aktivieren des Symbols angezeigt werden können. Symbolmenüs enthalten neben der Hilfefunktion weitere Funktionen. Nur mit der rechten Maustaste können Symbolmenüoptionen aktiviert werden.

Hinweis: Normalerweise entspricht die erste Option im Symbolmenü dem Doppelklicken mit der linken Maustaste auf dem Symbol. Ist nur eine Option (Hilfe) verfügbar, gilt dies nicht.

Ein Symbolmenü aufrufen

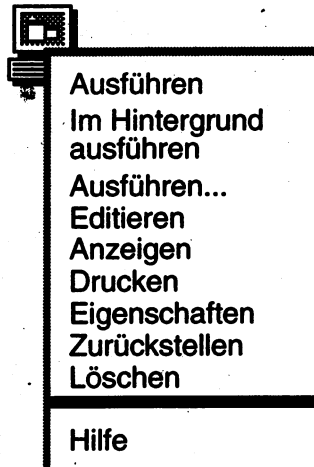
1. Den Mauszeiger auf das Symbol stellen und dann die rechte Maustaste drücken und gedrückt halten.
2. Wird das Menü des Symbols angezeigt, den Zeiger auf das Menü stellen und die Maustaste freigeben, wenn der Zeiger auf der gewünschten Auswahl steht.

Hinweis: Soll keine Auswahl getroffen werden, den Zeiger vom Menü herunterziehen und dabei weiterhin die Taste gedrückt halten, dann die Maustaste freigeben.

Die Hilfefunktion für ein bestimmtes Symbol aktivieren

1. Die rechte Maustaste auf dem Symbol drücken und gedrückt halten.
2. Wird das Symbolmenü angezeigt, den Zeiger im Menü auf **Hilfe** ziehen.
3. Zum Schließen des Hilfefensters auf den Knopf "Schließen" anklicken.

Beispiel für das Symbolmenü bei einem Programmsymbol



Ausführen	Entspricht dem Doppelklicken mit der linken Maustaste. Startet das Programm.
Im Hintergrund ausführen	Startet das Programm im Hintergrund.
Ausführen...	Zeigt ein Eingabefeld an, in das Parameter für den Befehl eingegeben werden können, und startet das Programm im Vordergrund.
Editieren	Öffnet die Datei zum Editieren. Diese Option ist abgeblendet, wenn auf das Symbol nur Lesezugriff besteht.
Anzeigen	Zeigt den Inhalt der Datei an.
Drucken	Druckt den Inhalt der Datei (falls diese eine Shell-Prozedur ist) auf dem Primärdrucker aus.
Eigenschaften	Zeigt Dateinformationen an.
Zurückstellen	Stellt die Datei zurück. Diese Option ist abgeblendet, wenn sich das Symbol auf keiner der Arbeitsoberflächen befindet
Löschen	Stellt die Datei in den Papierkorb. Diese Option ist abgeblendet, wenn auf das Symbol nur Lesezugriff besteht.
Hilfe	Zeigt Anweisungen zum Aktivieren des Symbols an. Diese Option ist abgeblendet, wenn das Symbol nicht Teil des ausgelieferten AIXwindows Desktop ist.

Hinweis: Binärdateien können nicht editiert oder gedruckt werden, daher werden diese abgeblendet dargestellt.

Symbole auswählen und die Auswahl zurücknehmen

Normalerweise wird ein Symbol ausgewählt, indem der Zeiger auf dieses Symbol gestellt und mit der linken Maustaste einfach geklickt wird. Bevor für ein Symbol eine Aktion ausgeführt werden kann, ist es auszuwählen. Symbole können einzeln oder in Gruppen ausgewählt werden. Je nach der Position der Symbole, die ausgewählt werden sollen, kann die Auswahlmethode variieren.

Ein einzelnes Symbol auswählen

Den Mauszeiger auf das Symbol oder den Symbolnamen stellen und auf dem Symbol mit der linken Maustaste klicken. Der Symbolname wird hervorgehoben dargestellt, um anzuzeigen, daß das Symbol ausgewählt wurde. Wenn andere Symbole ausgewählt, wird die Auswahl zurückgenommen.

Mehrere Symbole auswählen

Den Mauszeiger auf das erste Symbol oder den ersten Symbolnamen stellen und das Symbol auswählen. Weitere Symbole durch einfaches Klicken mit der mittleren Maustaste auf jedem Symbol auswählen. Das Klicken mit der mittleren Maustaste hat keine Auswirkung auf die anderen (nicht ausgewählten) Symbole.

Eine Gruppe von Symbolen durch Einpassen auswählen

1. Den Mauszeiger so an eine Ecke einer Gruppe von Symbolen stellen, daß diese keine Symbole oder Symbolnamen berührt.
2. Die linke Maustaste drücken und gedrückt halten und den Mauszeiger zur gegenüberliegenden Ecke ziehen. Um die Gruppe von Symbolen wird ein dehnbarer Rahmen erstellt.
3. Den dehnbaren Rahmen so ziehen, daß er die Symbole, die ausgewählt werden sollen, umschließt. Nach dem Freigeben der Maustaste werden die Symbole ausgewählt.

Symbole durch Einpassen zu einer ausgewählten Gruppe von Symbolen hinzufügen

1. Den Mauszeiger so auf eine Ecke der Gruppe von Symbolen stellen, daß der Zeiger keinen Teil eines Symbols oder Symbolnamens berührt.
2. Die Taste Strg und die Maustaste drücken und gedrückt halten.
3. Während die Tasten Strg und die linke Maustaste gedrückt sind, den Mauszeiger zur gegenüberliegenden Ecke ziehen. Um die Gruppe von Symbolen wird ein dehnbarer Rahmen erstellt.
4. Den dehnbaren Rahmen so ziehen, daß er die hinzuzufügenden Symbole umschließt oder berührt. Nach Freigabe der Taste Strg und der linken Maustaste sind die Symbole ausgewählt.

Die Auswahl eines Symbols zurücknehmen

Den Mauszeiger auf das Symbol oder den Symbolnamen stellen und mit der mittleren Maustaste auf dem Symbol klicken. Der Symbolname wird nicht länger hervorgehoben dargestellt, wodurch angezeigt wird, daß das Symbol nicht mehr ausgewählt ist. Diese Aktion hat keine Auswirkung auf die anderen Symbole.

Die Auswahl für mehrere Symbole zurücknehmen

Auf jedes ausgewählte Symbol, für das die Auswahl zurückgenommen werden soll, mit dem Mauszeiger zeigen und auf dem Symbol oder dem Symbolnamen mit der mittleren Maustaste klicken. Diese Aktion hat keine Auswirkung auf die anderen Symbole.

Die Auswahl einer Gruppe von Symbolen durch Einpassen zurücknehmen

1. Den Mauszeiger so an die Ecke der gewünschten Gruppe von Symbolen stellen, daß der Zeiger keine Symbole oder Symbolnamen berührt.
2. Die linke Maustaste und die Taste Strg drücken und gedrückt halten.
3. Während die Tasten Strg und die linke Maustaste gedrückt sind, den Mauszeiger zur gegenüberliegenden Ecke ziehen. Um die Gruppe von Symbolen wird ein dehnbare Rahmen erstellt.
4. Den dehnbaren Rahmen so ziehen, daß er die Symbole, für die die Auswahl zurückgenommen werden soll, umschließt. Nach Freigabe der Taste Strg und der linken Maustaste sind die Symbole nicht länger ausgewählt.

Auswahl aller Symbole zurücknehmen

Die Abbruchtaste drücken, um die Auswahl aller Symbole zurückzunehmen.

Symbole aktivieren

Ein Symbol kann durch Doppelklicken auf dem Symbol mit der Maustaste oder durch Übergeben eines Symbols an ein anderes aktiviert werden. Durch das Übergeben eines Symbols an ein anderes wird ein Programm aktiviert und diesem mitgeteilt, welche Datei als Quelle zu verwenden ist. Zum Beispiel kann eine Datei namens Bericht an ein Symbol übergeben werden, das einen Texteditor darstellt. Dadurch wird der Texteditor zum Bearbeiten der Datei Bericht gestartet. Durch das Ziehen und Übergeben von Symbolen können auch Anwendungen an Dateien ausgeführt werden. Zum Beispiel wird durch Übergeben einer Textdatei an ein Druckersymbol die Datei gedruckt.

Bei vielen Symbolen können mehrere Aktionen gestartet werden, je nachdem, welche Maustaste verwendet wird. Mit der linken Maustaste wird gewöhnlich die Primäraktion eines Symbols aktiviert.

Auf einem Symbol doppelt klicken

Mit der Maus auf ein Symbol zeigen und die linke Maustaste zweimal kurz hintereinander drücken (doppelt klicken), um die Standardaktion für das Symbol auszuführen.

Dokument- oder Dateisymbole öffnen

Im allgemeinen bewirkt das Doppelklicken mit der linken Maustaste das Ausführen einer \$EDITOR-Aktion und das Doppelklicken mit der mittleren Maustaste das Ausführen einer \$VIEWER-Aktion an der Datei.

Standardmäßig ist \$VIEWER=**pg/more** und \$EDITOR=**vi**.

Bei schreibgeschützten Dateien wird die \$VIEWER-Aktion sowohl bei der Verwendung der linken als auch der mittleren Maustaste aktiviert.

Bei Dateien, auf die kein Zugriff besteht, wird eine Nachricht angezeigt, daß die Datei weder gelesen noch in sie geschrieben werden kann.

Verzeichnissymbole öffnen

Beim Öffnen von Verzeichnissen hängt es von der verwendeten Maustaste ab, wo das Verzeichnis angezeigt wird.

Durch Doppelklicken mit der linken Maustaste auf einem Verzeichnissymbol wird das Verzeichnis geöffnet. Die Dateien im Verzeichnis werden im selben Fenster angezeigt. Diese Funktion ist nützlich, um sich im Dateibaum zu bewegen und dabei stets dasselbe Verzeichnisfenster zu verwenden.

Durch Doppelklicken mit der mittleren Maustaste auf dem Verzeichnissymbol wird das Verzeichnis geöffnet. Die im Verzeichnis enthaltenen Dateien werden in einem anderen Fenster angezeigt. Diese Funktion ist nützlich, wenn Dateien zwischen verschiedenen Fenstern kopiert bzw. verschoben werden sollen.

Programm- oder Anwendungssymbole ausführen

Durch Doppelklicken mit der linken Maustaste wird das Programm ausgeführt. Scheitert die Ausführung, wird die Standardfehlerausgabe angezeigt. Es kann angegeben werden, daß das Programm im Hintergrund ausgeführt werden soll. Dies hat keine Auswirkung auf die Fehlerprotokollierung.

Weitere Einsatzmöglichkeiten der Funktion Ziehen und Übergeben

Die Funktion Ziehen und Übergeben kann auch verwendet werden, um Informationen für AIXwindows Desktop-Dialogfenster zur Verfügung zu stellen. Der Name des an das Dialogfenster übergebenen Symbols wird zurückgegeben.

Symbole verschieben und kopieren

Datei- und Verzeichnissymbole stehen für tatsächliche Dateien und Verzeichnisse. Beim Verschieben oder Kopieren von Symbolen aus einem Verzeichnis in ein anderes wird die physische Position der Datei oder des Verzeichnisses geändert, die/das durch das Symbol dargestellt wird. Diese Funktionen können durch Ziehen und Übergeben der Symbole ausgeführt werden.

Symbole ziehen und übergeben

1. Auf dem zu ziehenden und zu übergebenden Symbol mit der linken Maustaste klicken.
2. Bei gedrückter linker Maustaste den Zeiger (der das Symbol bewegt) bewegen, bis sich das Symbol auf dem zu aktivierenden Symbol befindet.
3. Die Maustaste freigeben. Das Symbol wird an das aktivierte Symbol "übergeben", und eine Aktion wird ausgeführt.

Ein Symbol in ein anderes Verzeichnis verschieben

1. Den Mauszeiger auf das Symbol stellen.
2. Die mittlere Maustaste drücken und gedrückt halten.
3. Das Symbol in ein neues Verzeichnisfenster oder auf ein neues Verzeichnissymbol ziehen.
4. Die Maustaste freigeben.

Mehrere Symbole in ein anderes Verzeichnis verschieben

1. Alle zu verschiebenden Symbole durch einfaches Klicken oder Einpassen auswählen.
2. Auf eines der ausgewählten Symbole mit dem Mauszeiger zeigen.
3. Die mittlere Maustaste drücken und gedrückt halten.
4. Das Symbol in ein neues Verzeichnisfenster oder auf ein neues Verzeichnissymbol ziehen.

Hinweis: Die Cursorform ändert sich, um anzuzeigen, daß mehrere Symbole gleichzeitig gezogen werden.

5. Die Maustaste freigeben.

Ein Symbol in ein anderes Verzeichnis kopieren

1. Den Mauszeiger auf das Symbol stellen.
2. Die linke Maustaste drücken und gedrückt halten.
3. Das Symbol in ein neues Verzeichnisfenster oder auf ein neues Verzeichnissymbol ziehen.
4. Die Maustaste freigeben.

Mehrere Symbole in ein anderes Verzeichnis kopieren

1. Durch einfaches Klicken oder Einpassen alle zu verschiebenden Symbole auswählen.
2. Den Mauszeiger auf eines der ausgewählten Symbole stellen.
3. Die linke Maustaste drücken und gedrückt halten.

4. Das Symbol in ein neues Verzeichnisfenster oder auf ein neues Verzeichnissymbol ziehen.

Hinweis: Die Cursorform ändert sich, um anzuzeigen, daß mehrere Symbole gleichzeitig gezogen werden.

5. Die Maustaste freigeben.

Warnungen beim Verschieben oder Kopieren von Symbolen

Warnungen weisen den Benutzer auf anstehende möglicherweise nicht umkehrbare Aktionen hin:

Bewegen Eine Fehlermeldung informiert den Benutzer, wenn ein Objekt nicht an eine neue Position verschoben werden kann, und stellt nützliche Informationen zur Verfügung. Ist an der Zielposition nicht genügend Speicherplatz vorhanden, wird ebenfalls eine Fehlermeldung ausgegeben und die Operation wird nicht ausgeführt.

Fordert der Benutzer das Verschieben eines Verzeichnisses oder einer anderen Gruppe von Objekten an und die Ausführung der Operation könnte lange dauern, oder wird das Verschieben eines Objekts an die Position eines anderen Objekts mit dem selben Namen angefordert, so daß letzteres verloren geht, fordert das System den Benutzer zur Bestätigung auf.

Kopieren Eine Fehlermeldung informiert den Benutzer, wenn ein Objekt nicht an eine neue Position kopiert werden kann, und stellt hilfreiche Informationen zur Verfügung. Ist an der Zielposition nicht genügend Speicherplatz vorhanden, wird ebenfalls eine Fehlermeldung ausgegeben und die Operation wird nicht ausgeführt.

Fordert der Benutzer das Kopieren eines Verzeichnisses oder einer anderen Gruppe von Objekten an und die Ausführung der Operation könnte lange dauern, oder wird das Kopieren eines Objekts über ein anderes Objekt mit demselben Namen angefordert, fordert das System den Benutzer zur Bestätigung auf.

Auch im Zielverzeichnisfenster wird das Ergebnis der Operation sichtbar. Die Kopierfunktion kann als Multitasking-Operation ausgeführt werden.

Symbole umbenennen

Symbole können nur in Verzeichnisfenstern umbenannt werden.

Erforderliche Schritte

1. Die Taste Alt gedrückt halten und mit der linken Maustaste auf dem Symbol oder dem Symbolnamen klicken. Der Cursor ändert seine Form und folgendes Eingabefeld wird angezeigt:

Neuer Name:

2. Den Symbolnamen eingeben und die Eingabetaste drücken.

Hinweis: Zum Abbrechen der Umbenennungsoperation, den Mauszeiger außerhalb des Symbols stellen und mit der linken Maustaste klicken.

Symbole löschen

Nicht mehr benötigte Symbole können gelöscht werden, indem sie zum Symbol **Papierkorb** gezogen und an dieses übergeben werden. Die gelöschten Dateien und Verzeichnisse werden im Verzeichnis **Papierkorb** auf der Arbeitsoberfläche gespeichert. Diese Symbole bleiben im Symbol **Papierkorb** bis der **Papierkorb** geleert wird oder die Symbole wiederhergestellt werden.

Löschen von Symbolen

1. Die zu löschenden Symbole auswählen.
2. Die Symbole mit der linken Maustaste zum Symbol **Papierkorb** ziehen und an dieses übergeben.

Die Symbole verschwinden von der Arbeitsoberfläche oder aus dem Verzeichnis.

Den Inhalt des Papierkorbs anzeigen

Zum Anzeigen des Inhalts mit der linken Maustaste auf dem Symbol **Papierkorb** doppelt klicken.

Symbole aus dem Papierkorb wiederherstellen

1. Zum Anzeigen des Inhalts mit der linken Maustaste auf dem Symbol **Papierkorb** doppelt klicken.
2. Die wiederherzustellenden Symbole auswählen.
3. Mit der mittleren Maustaste die Symbole in ein offenes Verzeichnisfenster oder auf ein Verzeichnissymbol bewegen (ziehen und übergeben).

Hinweis: Wird versucht, ein Symbol in ein Verzeichnis zu stellen, das bereits ein Symbol mit diesem Namen enthält, wird eine Warnung ausgegeben und die Operation abgebrochen.

Inhalt des Symbols Papierkorb löschen

1. Das Menü des Symbols **Papierkorb** durch Drücken und Halten der rechten Maustaste aufrufen.
2. Die Option **Leer** auswählen.

Eine Warnung wird angezeigt, damit der Benutzer die Gelegenheit hat, die Operation abzubrechen.

AIXwindows Desktop-Menüs

AIXwindows stellt für Arbeitsoberflächen und Verzeichnisse unterschiedliche Fenster zur Verfügung. Arbeitsoberflächen und Verzeichnisfenster werden am besten für unterschiedliche Zwecke verwendet.

Ein Arbeitsoberflächenfenster stellt, ebenso wie ein richtiger Schreibtisch, eine Arbeitsoberfläche dar. Es empfiehlt sich, für jedes Projekt eine eigene Arbeitsfläche einzurichten. AIXwindows ermöglicht dem Benutzer unterschiedlicher Projekte auf verschiedenen Arbeitsoberflächen auszuführen und so den Arbeitsbereich bequem zu organisieren.

Ein Verzeichnisfenster stellt ein physisches Verzeichnis dar. Zum Beispiel ist es möglich, zur Datenorganisation mehrere Ablageschränke zu erstellen. Die Ablageschränke entsprechen Verzeichnissen, Schubladen den Unterverzeichnissen und Papierdokumente den Textdateien. In AIXwindows werden die Daten durch Verzeichnisfenster organisiert. Mit Verzeichnisfenstern und ihrem Inhalt können allgemeine Aufgaben für Verzeichnisse, wie Editieren, Sortieren und Ändern von Dateiberechtigungen ausgeführt werden.

AIXwindows-Menüs ermöglichen den einfachen Zugriff auf häufig verwendete Befehle und Funktionen. Am oberen Rand von Arbeitsoberflächen- und Verzeichnisfenstern befindet sich eine Menüleiste mit den Namen von Menüs. Arbeitsoberflächenfenster stellen die folgenden fünf Menüs zur Verfügung: Datei, Editieren, Anzeigen, Optionen und Hilfe. In Verzeichnisfenstern stehen die folgenden vier Menüs zur Verfügung: Datei, Editieren, Anzeigen und Hilfe.

Zwar haben die entsprechenden Menüs in den beiden Fenstern dieselben Namen, ihr Inhalt ist jedoch unterschiedlich. Optionen des Arbeitsoberflächenmenüs ermöglichen dem Benutzer das Anpassen des Arbeitsbereichs und das Ausführen von allgemeinen Arbeitsoberflächenaufgaben wie das Umschalten zwischen Arbeitsoberflächen und das Zurückstellen von Dateien in ihre Originalverzeichnisse. Vom Arbeitsoberflächenmenü aus können Standardeinheiten für Drucker und Standardoptionen für allgemeine Befehle geändert werden. Verzeichnismenüoptionen ermöglichen das Organisieren von Verzeichnissen, Arbeiten mit Dateien und Ausführen diverser Dateiverwaltungsfunktionen.

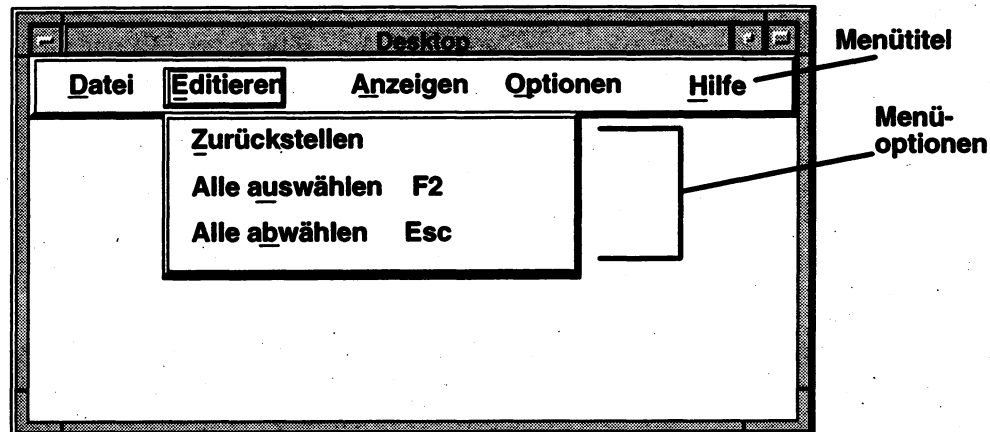
Menüoptionen, die unter den gegebenen Bedingungen ausgeführt werden können, werden normal angezeigt. Kann eine Option unter den gegebenen Bedingungen nicht ausgeführt werden, wird sie abgeblendet dargestellt. Die Option wird zwar angezeigt, aber AIXwindows ignoriert alle Tastatur- oder Mauseaktionen für diese Option.

Folgende Abschnitte enthalten weitere Informationen zu AIXwindows-Menüs:

- Eine AIXwindows-Menüoption auswählen auf Seite 5-25.
- Arbeitsoberflächenmenüs und -optionen bei AIXwindows auf Seite 5-26.
- Verzeichnismenüs und -optionen bei AIXwindows auf Seite 5-30.
- Tastenäquivalente für Menüoptionen bei AIXwindows auf Seite 5-34.

Eine AIXwindows-Menüoption auswählen

Jedes Menü verfügt über einen Titel, der in der Menüleiste angezeigt wird. Unter jedem Titel befinden sich mehrere Menüoptionen mit Befehlen die vom Menü ausgeführt werden können. Diese Optionen werden angezeigt, wenn mit der linken Maustaste auf dem Menütitel geklickt wird.



Optionen des Arbeitsoberflächenmenüs "Editieren"

Durch einfaches Klicken aus dem Menü auswählen

1. Den Mauszeiger auf den Menütitel stellen und mit der linken Maustaste klicken. Die Optionen des Menüs werden angezeigt.
2. Den Mauszeiger auf die auszuwählende Option stellen und mit der linken Maustaste klicken.

Zum Schließen des Menüs, ohne eine Option auszuwählen, den Mauszeiger in einen Bereich außerhalb des Menüs bewegen und mit der linken Maustaste klicken.

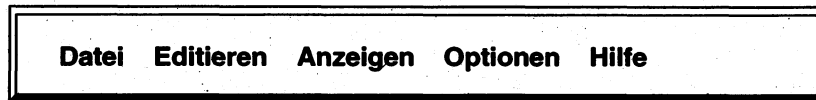
Durch Ziehen eine Menüoption auswählen

1. Den Mauszeiger auf den Menütitel stellen und die linke Maustaste drücken und gedrückt halten. Die Optionen des Menüs werden angezeigt.
2. Die Maustaste weiterhin gedrückt halten und dabei den Mauszeiger auf die gewünschte Option stellen. Dann die Maustaste freigeben.

Zum Schließen des Menüs, ohne eine Option auszuwählen, den Mauszeiger in einen Bereich außerhalb des Menüs bewegen und mit der linken Maustaste klicken.

Arbeitsoberflächenmenüs und -optionen bei AIXwindows

Am oberen Rand des Arbeitsoberflächenfensters wird eine Menüleiste angezeigt. Über die Menüleiste kann auf die Arbeitsoberflächenmenüs zugegriffen werden.



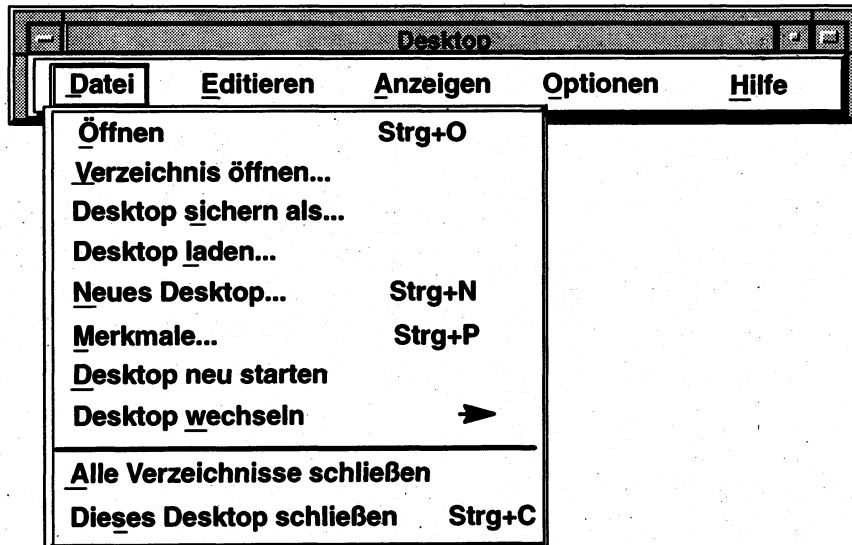
Arbeitsoberflächenmenüleiste

Jedes Menü enthält verschiedene Optionen, die in den folgenden Abschnitten beschrieben werden.

- Optionen des Arbeitsoberflächenmenüs "Datei" auf Seite 5-26.
- Optionen des Arbeitsoberflächenmenüs "Editieren" auf Seite 5-27.
- Optionen des Arbeitsoberflächenmenüs "Anzeigen" auf Seite 5-28.
- Optionen des Arbeitsoberflächenmenüs "Optionen" auf Seite 5-28.
- Optionen des Arbeitsoberflächenmenüs "Hilfe" auf Seite 5-29.

Optionen des Arbeitsoberflächenmenüs "Datei"

Das Menü "Datei" enthält die folgenden Optionen:



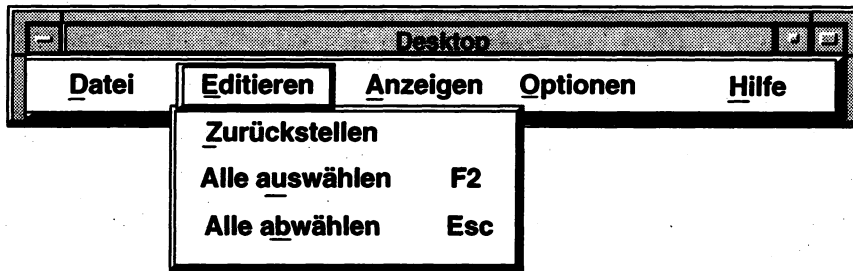
Optionen des Arbeitsoberflächenmenüs "Datei"

- | | |
|-------------------------------|---|
| Öffnen | Öffnet die ausgewählten Symbole. Entspricht dem Doppelklicken auf dem Symbolbild. |
| Verzeichnis öffnen... | Öffnet das angegebene Verzeichnis. |
| Desktop sichern als... | Sichert die Arbeitsoberfläche im Benutzerverzeichnis (Home) unter dem angegebenen Namen. |
| Desktop laden... | Lädt die Arbeitsoberfläche aus der angegebenen Datei und sichert das aktuelle Desktop. |

- Neues Desktop...** Erstellt eine neue Arbeitsoberfläche.
- Desktop neu starten** Sichert alle geöffneten Arbeitsoberflächen und startet dann AIXwindows erneut. Dabei werden die neuesten Änderungen der Standardwerte und Umgebungsvariablen verwendet.
- Desktop wechseln** Ersetzt die aktuelle Arbeitsoberfläche durch eine ausgewählte Arbeitsoberfläche, z. B. durch eines der folgenden:
Allgemein – Öffnet das **General Desktop**.
Power – Öffnet das **Power Desktop**.
Admin – Öffnet das **Admin Desktop**.
- Alle Verzeichnisse schließen** Schließt alle Verzeichnisse auf allen Arbeitsoberflächen. Diese Option wird abgeblendet angezeigt, wenn keine Verzeichnisse geöffnet sind.
- Dieses Desktop schließen** Sichert und schließt die aktuelle Arbeitsoberfläche. Ist diese Arbeitsoberfläche die letzte geöffnete Arbeitsoberfläche, wird der Benutzer vom System zum Bestätigen aufgefordert. Entspricht dem Doppelklicken auf dem Fenstermenüknopf.

Optionen des Arbeitsoberflächenmenüs "Editieren"

Das Menü "Editieren" enthält die folgenden Optionen:

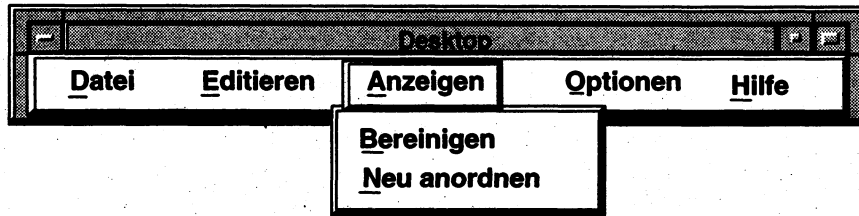


Optionen des Arbeitsoberflächenmenüs "Editieren"

- Zurückstellen** Entfernt die ausgewählten Symbole und stellt sie in das Originalverzeichnis zurück. Diese Option wird abgeblendet angezeigt, wenn kein Symbol ausgewählt ist.
- Alle auswählen** Wählt alle Symbole auf der Arbeitsoberfläche aus.
- Alle abwählen** Nimmt die Auswahl aller Symbole auf der Arbeitsoberfläche zurück. Diese Option wird abgeblendet angezeigt, wenn kein Symbol ausgewählt ist.

Optionen des Arbeitsoberflächenmenüs "Anzeigen"

Das Menü "Anzeigen" enthält die folgenden Optionen:



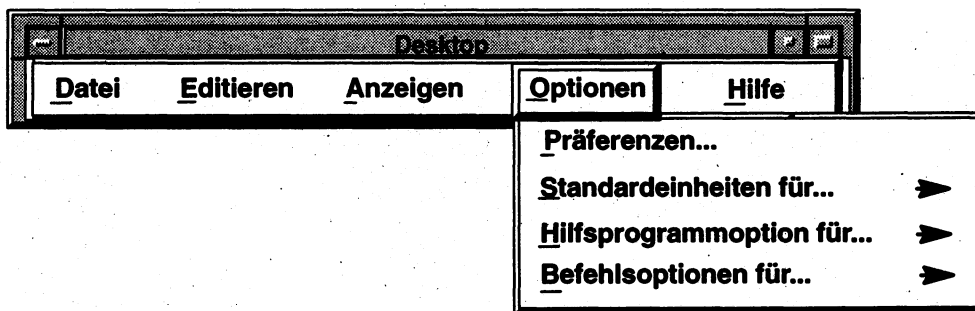
Optionen des Arbeitsoberflächenmenüs "Anzeigen"

Bereinigen Versetzt die Symbole an die nächste Gitterposition (Schnittpunkt zwischen Zeile und Spalte).

Neu anordnen Ordnet die Symbole in regelmäßigen Abständen an.

Optionen des Arbeitsoberflächenmenüs "Optionen"

Das Menü "Optionen" enthält die folgenden Optionen:



Optionen des Arbeitsoberflächenmenüs "Optionen"

Präferenzen... Führt den Profileditor aus.

Standardeinheiten für... Ordnet Standardeinheiten zu. Mögliche Optionen sind:
Primärdrucker – Zeigt die aktuelle Adresse des Primärdruckers an und ermöglicht dem Benutzer das Ändern des Standardwerts.

Alternativdrucker – Zeigt die aktuelle Adresse des Alternativdruckers an und ermöglicht dem Benutzer das Ändern des Standardwerts.

1/4-Zoll-Band – Zeigt die aktuelle Adresse des 1/4-Zoll-Bandlaufwerks an und ermöglicht dem Benutzer das Ändern des Standardwerts

8-mm-Band – Zeigt die aktuelle Adresse des 8-mm-Bandlaufwerks an und ermöglicht dem Benutzer das Ändern des Standardwerts.

Hilfsprogrammoptionen für... Zeigt die Standardhilfsprogramme an und ändert sie. Mögliche Optionen sind:

Editor – Zeigt den aktuellen Editor an und ermöglicht dem Benutzer das Ändern des Standardwerts.

Blättern – Zeigt das aktuelle Programm an, das zum Blättern verwendet wird, und ermöglicht dem Benutzer das Ändern des Standardwerts.

Befehloptionen für...

Zeigt die Standardwerte und Umgebungsvariablen an und ändert sie. Mögliche Optionen sind:

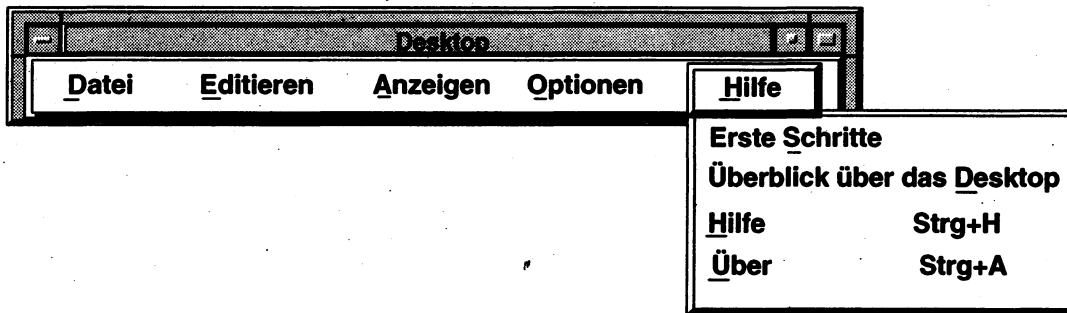
Namensanzeige – Zeigt die Optionen an, die für **Anzeigen** → **Namen** im Verzeichnismenü gesetzt werden müssen, und ermöglicht dem Benutzer das Ändern der Optionen.

Befehl ps – Zeigt die für den Befehl **ps** gesetzten Optionen an und ermöglicht dem Benutzer das Ändern der Optionen.

Befehl cc – Zeigt den aktuellen Compiler an und ermöglicht dem Benutzer das Ändern des Compilers.

Optionen des Arbeitsoberflächenmenüs "Hilfe"

Das Hilfemenü enthält die folgenden Optionen:



Optionen des Arbeitsoberflächenmenüs "Hilfe"

Erste Schritte Startet ein interaktives Lernprogramm über AIXwindows.

Überblick über das Desktop Ruft InfoExplorer auf und öffnet einen Artikel, der einen Überblick über die Arbeitsoberfläche enthält.

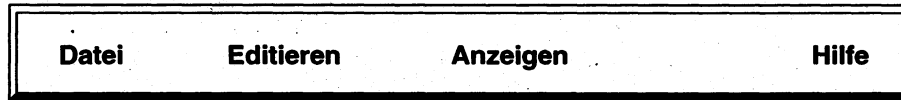
Hilfe Zeigt einen Überblick über die verfügbare Hilfe an.

README Desktop Zeigt eine Textdatei an, die eine Einführung in die gerade verwendete Arbeitsoberfläche und Informationen zur Verwendung von InfoExplorer zum Abrufen von Hilfe über weitere Themen enthält. Jeder Arbeitsoberfläche ist eine eigene Datei **README Desktop** zugeordnet.

Über Zeigt Informationen über Version und Warenzeichen an.

Verzeichnismenüs und -optionen bei AIXwindows

Am oberen Rand des Verzeichnisfensters wird eine Menüleiste angezeigt. Die Menüleiste ermöglicht den Zugriff auf die Verzeichnismenüoptionen.



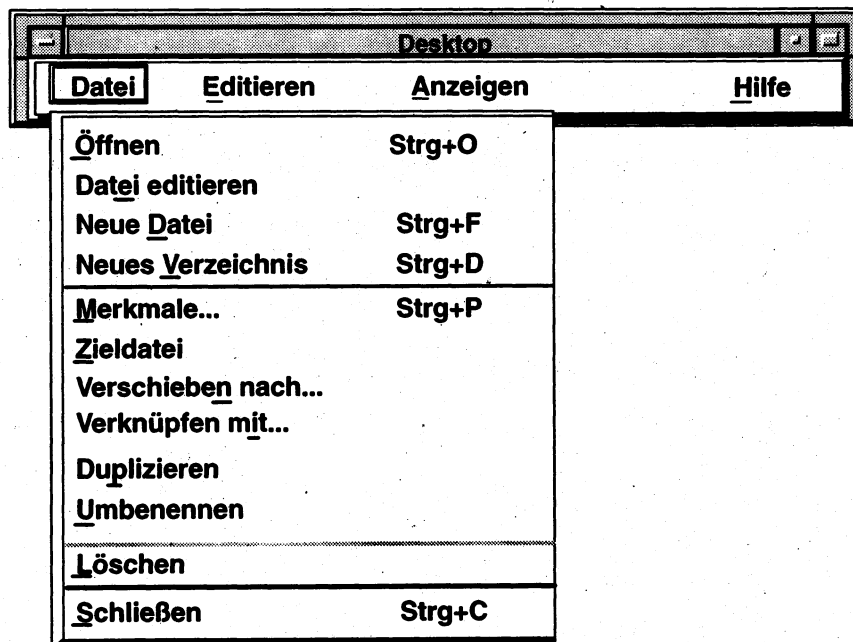
Verzeichnismenüleiste

Die einzelnen Menüs enthalten unterschiedliche Optionen, die in den nachfolgenden Abschnitten näher erläutert werden.

- Optionen im Verzeichnismenü "Datei" auf Seite 5-30.
- Optionen im Verzeichnismenü "Editieren" auf Seite 5-31.
- Optionen im Verzeichnismenü "Anzeigen" auf Seite 5-32.
- Optionen im Verzeichnismenü "Hilfe" auf Seite 5-32.

Optionen im Verzeichnismenü "Datei"

Das Menü "Datei" enthält die folgenden Optionen:



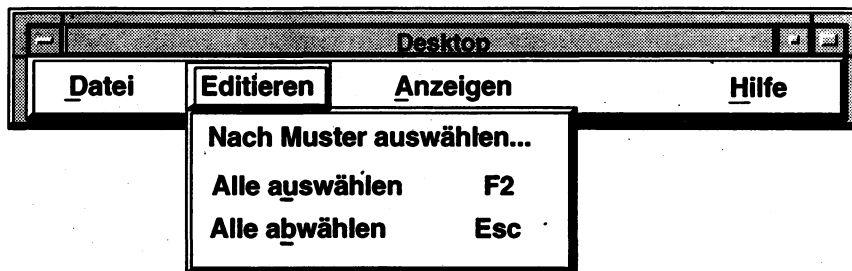
Optionen im Verzeichnismenü "Datei"

- | | |
|--------------------------|---|
| Öffnen | Öffnet die ausgewählte(n) Datei(en). |
| Datei Editieren | Editiert die ausgewählte(n) Datei(en). Diese Option wird abgeblendet angezeigt, wenn kein Symbol ausgewählt ist oder für das ausgewählte Symbol nur Leseberechtigung besteht. |
| Neue Datei | Erstellt eine neue, leere Datei. |
| Neues Verzeichnis | Erstellt ein neues, leeres Verzeichnis. |

Merkmale...	Zeigt Dateiinformationen an. Diese Option wird abgeblendet angezeigt, wenn kein Symbol ausgewählt ist.
Kopieren in...	Kopiert die ausgewählte(n) Datei(en) in ein angegebenes Verzeichnis. Diese Option wird abgeblendet angezeigt, wenn kein Symbol ausgewählt ist oder für das ausgewählte Symbol nur Leseberechtigung besteht.
Verschieben nach...	Verschiebt die ausgewählte(n) Datei(en) in ein angegebenes Verzeichnis. Diese Option wird abgeblendet angezeigt, wenn kein Symbol ausgewählt ist oder für das ausgewählte Symbol nur Leseberechtigung besteht.
Verknüpfen mit...	Stellt eine Verbindung zwischen einer ausgewählten Datei und einem angegebenen Verzeichnis her. Diese Option wird abgeblendet angezeigt, wenn kein Symbol ausgewählt ist oder für das ausgewählte Symbol nur Leseberechtigung besteht.
Duplizieren	Dupliziert die ausgewählte(n) Datei(en). Diese Option wird abgeblendet angezeigt, wenn kein Symbol ausgewählt ist oder für das ausgewählte Symbol nur Leseberechtigung besteht.
Umbenennen...	Benennt die ausgewählte(n) Datei(en) um. Diese Option wird abgeblendet angezeigt, wenn kein Symbol ausgewählt ist oder für das ausgewählte Symbol nur Leseberechtigung besteht.
Löschen	Löscht die ausgewählte(n) Datei(en) im Verzeichnis Papierkorb , ohne diese vorher zu archivieren. Diese Option wird abgeblendet angezeigt, wenn kein Symbol ausgewählt ist, oder für das ausgewählte Symbol nur Leseberechtigung besteht.
Verzeichnis schließen	Schließt das aktuelle Verzeichnis.

Optionen im Verzeichnismenü "Editieren"

Das Menü "Editieren" enthält die folgenden Optionen:



Optionen im Verzeichnismenü "Editieren"

Nach Muster auswählen...

Fordert zur Eingabe eines Textmusters auf und wählt dann alle Dateien im aktuellen Verzeichnis aus (ausgenommen die Punktdateien . und ..), deren Namen das angegebene Muster enthält.

Alle auswählen

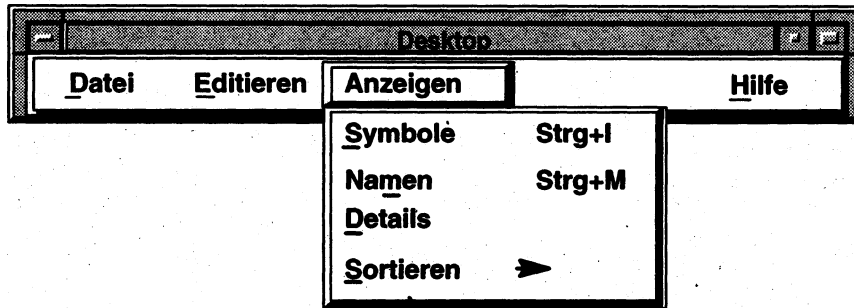
Wählt alle Dateien im aktuellen Verzeichnis mit Ausnahme der Punktdateien (. und ..) aus.

Alle abwählen

Nimmt die Auswahl aller Dateien in diesem Verzeichnis zurück. Diese Option wird abgeblendet angezeigt, wenn kein Symbol ausgewählt ist.

Optionen im Verzeichnismenü "Anzeigen"

Das Menü "Anzeigen" enthält die folgenden Optionen:

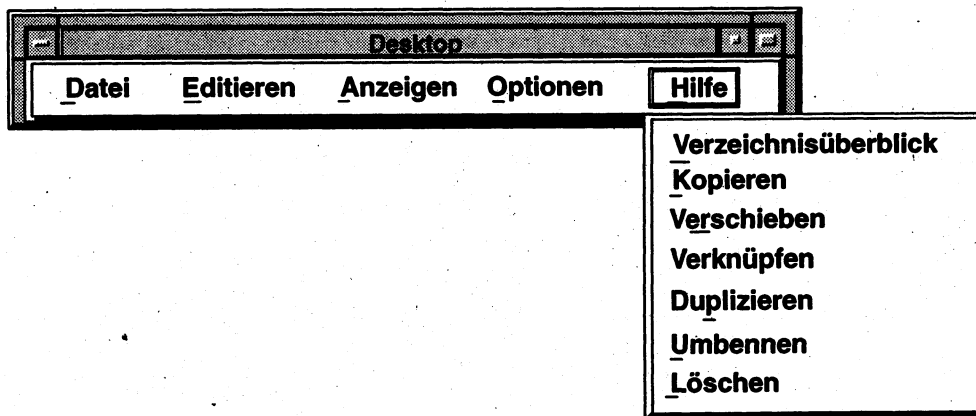


Optionen im Verzeichnismenü "Anzeigen"

- Symbole** Zeigt die Dateien im Verzeichnis als Standardsymbole an.
- Namen** Zeigt den Inhalt des Verzeichnisses (nur Namen) an.
- Details** Zeigt den Verzeichnisinhalt an (Namen, Größe, Änderungsdatum und -zeit von Dateien an).
- Sortieren** Sortiert Dateien nach Namen, Zeit oder Größe. Mögliche Optionen sind:
nach Verzeichnis – Zeigt die Dateien in Standardsymboldarstellung an.
nach Namen – Zeigt nur die Namen an (Standardeinstellung).
nach Zeit – Sortiert nach dem Zeitpunkt der letzten Änderung.
nach Größe – Sortiert nach Dateigröße (absteigende Reihenfolge).

Optionen im Verzeichnismenü "Hilfe"

Das Menü Hilfe enthält die folgenden Optionen:



Optionen im Verzeichnismenü "Hilfe"

- Verzeichnisüberblick** Ruft InfoExplorer auf und zeigt einen Artikel an, der einen Überblick über Verzeichnisse bei AIXwindows bietet.
- Kopieren** Enthält Hilfetexte zum Kopieren von Dateien.
- Verschieben** Enthält Hilfetexte zum Verschieben von Dateien.
- Verknüpfen** Enthält Hilfetexte zum Herstellen von Verbindungen zwischen Dateien.
- Duplizieren** Enthält Hilfetexte zum Kopieren von Dateien.

Umbenennen
Löschen

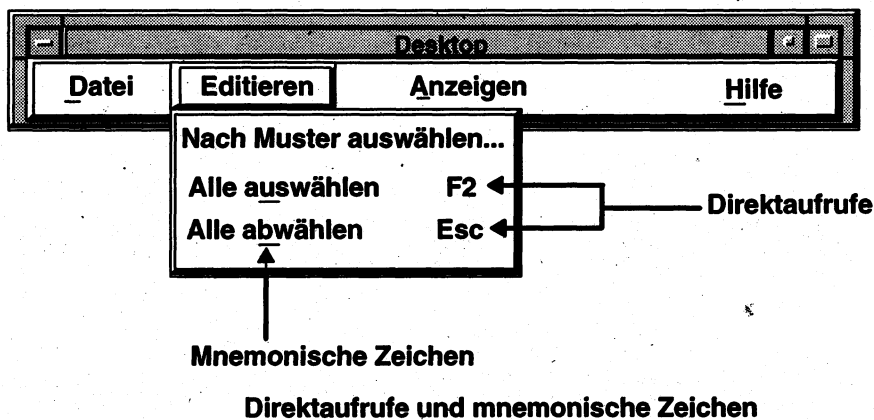
Enthält Hilfetexte zum Umbenennen von Dateien.
Enthält Hilfetexte zum Löschen von Dateien.

Tastenäquivalente für Menüoptionen bei AIXwindows

Es gibt zwei Möglichkeiten, auf Menüs und die zugehörigen Optionen schnell und ohne Verwendung der Maus zuzugreifen.

- Unter Direktaufruf versteht man eine Taste der Tastatur, der von der Anwendung eine Funktion in einem Menü zugeordnet wurde, die durch Drücken der Taste direkt ausgeführt werden kann. Die Direktaufrufe werden in den Menüs neben den Funktionen aufgelistet.
- Mnemonische Zeichen bieten ähnliche Möglichkeiten, indem sie dem Benutzer ermöglichen, innerhalb eines Menüs durch Eingeben bestimmter Zeichen eine Auswahl zu treffen. Mnemonische Zeichen werden im Menü als Buchstaben mit Unterstreichung gekennzeichnet.

Folgende Abbildung veranschaulicht ein Menü mit Direktaufrufen und mnemonischen Zeichen.



Es ist zu beachten, daß mit Direktaufrufen eine Funktion ausgeführt werden kann, ohne das Menü anzuzeigen. Mnemonische Zeichen bieten lediglich eine einfache Methode zum Anzeigen von Menüs und zum Auswählen von Menüoptionen ohne Verwendung einer Maus.

Eine Menüoption mit einem Direktaufruf auswählen

Die zugeordnete Funktionstaste drücken. Das Menü muß nicht angezeigt werden.

Im Menü "Editieren" werden durch Drücken der Taste F2 alle Symbole auf der Arbeitsoberfläche ausgewählt.

Ein Menü über das zugeordnete mnemonische Zeichen auswählen

Die Taste Alt drücken und gedrückt halten und dann den Buchstaben drücken, der im Menünamen unterstrichen ist.

Beim Menü "Editieren" wird z. B. ein Direktaufruf in der Menüleiste angezeigt. Dies bedeutet, daß der Benutzer durch Drücken von Alt-E auf das Menü "Editieren" zugreifen kann.

Eine Menüoption über das zugeordnete mnemonische Zeichen auswählen

Nachdem ein Menü angezeigt wird, kann mit einem mnemonischen Zeichen eine Funktion ausgeführt werden. Hierzu den im Funktionsnamen unterstrichenen Buchstaben eingeben.

Im Menü "Editieren" werden durch Drücken der Taste U alle Symbole auf der Arbeitsoberfläche ausgewählt.

Folgende Tabelle enthält eine Liste der mnemonischen Zeichen und Direktaufrufe des Arbeitsoberflächenmenüs.

Menüoption	Mnemonisches Zeichen	Direktaufruf
Menü "Datei"	D	
Öffnen	Ö	Strg-O
Verzeichnis öffnen	V	
Desktop sichern als...	i	
Desktop laden..	I	
Neues Desktop...	N	Strg-N
Merkmale...	M	Strg-P
Desktop neu starten	D	
Desktop wechseln	w	
Allgemein	I	
Power	P	
Admin	A	
Alle Verzeichnisse schließen	A	
Dieses Desktop schließen	s	
Menü "Editieren"	E	
Zurückstellen	Z	
Alle auswählen	u	F2
Alle abwählen	b	Esc
Menü "Anzeigen"	A	Alt-V
Bereinigen	B	Strg-C
Neu anordnen	N	Strg-R
Menü "Optionen"	O	
Präferenzen	P	
Standardeinheiten für...	S	
Primärdrucker	P	
Alternativdrucker	A	
8-mm-Band	8	
1/4-Zoll-Band	1	
Hilfsprogrammoptionen für...	H	
Editor	E	
Blättern	B	
Befehloptionen für...	B	
Namensanzeige	N	
Basisname	B	
Symbolname	S	
Größe, Datum, Basisname	G	
Klasse, Größe, Datum, Symbolname	D	
Klasse, Datum, Größe, Basisname	K	
Übrige	Ü	

Menüoption	Mnemonisches Zeichen	Direktaufruf
Menü "Optionen" Befehl ps Befehl cc	O p c	
Menü "Hilfe" Erste Schritte Überblick über das Desktop Hilfe Über	H S D H Ü	Strg-H Strg-A

Folgende Tabelle enthält eine Auflistung der Direktaufrufe und mnemonischen Zeichen des Verzeichnismenüs von AIXwindows Desktop.

Menüoption	Mnemonisches Zeichen	Direktaufruf
Menü "Datei" Öffnen Datei editieren Neue Datei Neues Verzeichnis' Merkmale... Zieldatei Verschieben nach... Verknüpfen mit... Duplizieren Umbenennen... Löschen Schließen	D Ö e D V M Z n i p U L S	Strg-O Strg-F Strg-D Strg-P Strg-C
Menü "Editieren" Nach Muster auswählen Alle auswählen Alle abwählen	E u b	 F2 Esc
Menü "Anzeigen" Symbole Namen Details Sortieren nach Verzeichnis nach Namen nach Zeit nach Größe	A S m D S N Z G	Strg-I Strg-M

Menüoption	Mnemorisches Zeichen	Direktaufruf
Menü "Hilfe"	H	
Verzeichnisüberblick	V	
Kopieren	K	
Verschieben	r	
Duplizieren	p	
Umbenennen	U	
Löschen	L	

Arbeitsoberflächentypen bei AIXwindows Desktop

Bei AIXwindows Desktop stehen drei (vorkonfigurierte) Arbeitsoberflächentypen mit jeweils unterschiedlichen Gruppen von Funktionen zur Verfügung, die speziell an die Anforderungen einzelner Benutzer angepaßt sind. Als Benutzerarbeitsoberflächen sind das *General Desktop* und das *Power Desktop* verfügbar. Eine dieser Arbeitsoberflächen wird als Standardarbeitsoberfläche installiert. Eine dritte Arbeitsoberfläche, das *Admin Desktop*, ist den Erfordernissen von Systemadministratoren angepaßt. Der Name der jeweiligen Arbeitsoberfläche wird in der Titelleiste angezeigt.

Von den drei Arbeitsoberflächentypen ist das **General-Desktop** für die Erfordernisse von Benutzern mit geringen Vorkenntnissen ausgelegt. Es sind keine umfassenden EDV-Kenntnisse erforderlich, denn der Benutzer wird beim Ausführen von Aktionen durch die Abbildungen auf der Arbeitsoberfläche unterstützt.

Über das **Power Desktop** hingegen erhält der erfahrene Programmierer einfachen Zugriff auf leistungsstarke Betriebssystemfunktionen. Die umfassendste Arbeitsoberfläche ist **Admin Desktop**. Sie enthält alle Funktionen des **Power Desktop** und zusätzliche Systemverwaltungsfunktionen zum Ausführen von Aufgaben wie das Verwalten von Druckern und anderen peripheren Einheiten, das Zuordnen von Ressourcen und das Definieren neuer Benutzer.

Jede Arbeitsoberfläche wird in einer eigenen Datei definiert, in der die Symbole angegeben sind, die am Anfang auf der Arbeitsoberfläche vorhanden sind. Diese Datei enthält Regeln zum Definieren bestimmter Aktionen für Symbole. Die Arbeitsoberflächen können so verwendet werden, wie sie in der Standardumgebung definiert sind. Sie können aber auch an die Anforderungen des jeweiligen Benutzers angepaßt werden.

Für jeden Benutzer wird mit dem Hilfsprogramm **xdt3newuser** eine der drei Standardarbeitsoberflächen installiert. Der Systemadministrator gibt mit dem Hilfsprogramm **xdt3newuser** an, auf welche Arbeitsoberfläche ein Benutzer Zugriff hat. Benutzer können durch Auswahl der Option **Desktop wechseln** in einem Arbeitsoberflächenmenü "Datei" zu einer anderen Arbeitsoberfläche wechseln.

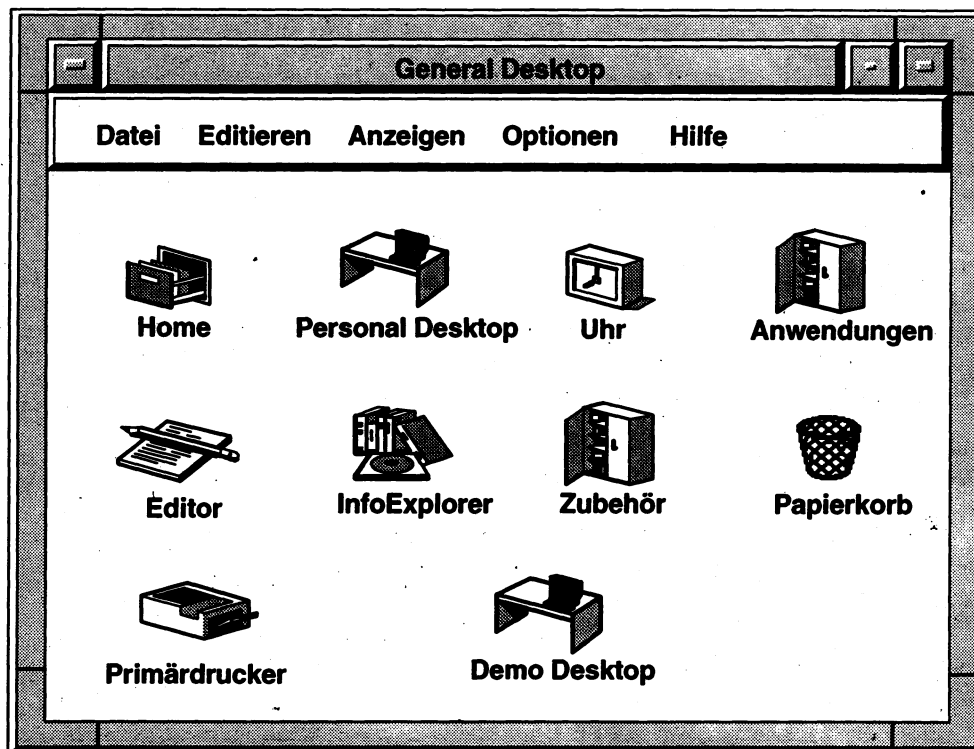
In den folgenden Abschnitten werden die Arbeitsoberflächen **General Desktop**, **Power Desktop** und **Admin Desktop** und die zugehörigen Hilfsprogramme beschrieben.

- General Desktop auf Seite 5-39.
- Power Desktop auf Seite 5-45.
- Admin Desktop auf Seite 5-51.
- Neue Benutzer definieren auf Seite 5-57.

General Desktop

Über die Arbeitsoberfläche General Desktop können eine Reihe von grundlegenden Operationen von Benutzern ohne Vorkenntnisse oder mit geringen EDV- Kenntnissen ausgeführt werden. Diese Arbeitsoberfläche bietet besondere Sicherheitsvorkehrungen für den Benutzer. Das System zeigt z. B. eine Aufforderung zum Bestätigen an, wenn der Benutzer versucht, eine Datei über eine bereits vorhandene Datei zu kopieren, oder ein vollständiges Verzeichnis zu kopieren. Obwohl die Standardarbeitsoberfläche die Funktionen zum Ausführen der meisten grundlegenden Aufgaben enthält, können vom Systemadministrator zusätzliche Funktionen oder weitere Einschränkungen hinzugefügt werden.

Das General Desktop ist die Basisarbeitsoberfläche für alle Benutzer von AIXwindows Desktop. Beherrscht der Benutzer die Funktionen dieser Arbeitsoberfläche, kann er die Arbeitsoberflächen mit erweiterten Funktionen verwenden. Das General Desktop und einige der zugehörigen Symbole werden in der folgenden Abbildung dargestellt.



General Desktop mit großen Symbolen

Die Arbeitsoberfläche General Desktop wird in folgenden Abschnitten näher beschrieben:

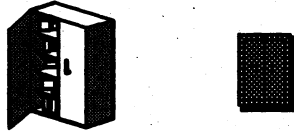
- Symbole des General Desktop auf Seite 5-40.
- Sonderdateien des General Desktop auf Seite 5-42.
- Sichtbare Dateien des General Desktop auf Seite 5-42.
- Zubehörsymbole des General Desktop auf Seite 5-43.

Symbole des General Desktop

Über das General Desktop besteht Zugriff auf Hilfsprogramme und Dateien, die von Benutzern ohne Vorkenntnisse oder mit geringen Vorkenntnissen benötigt werden. Es folgt eine Beschreibung einiger Symbole, die auf der Arbeitsoberfläche General Desktop zur Verfügung stehen.

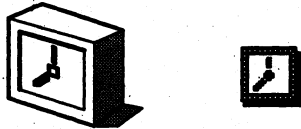
Hinweis: Manche Funktionen können sowohl durch ein großes als auch durch ein kleines Symbol dargestellt werden.

Anwendungen



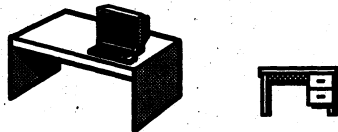
Das Symbol **Anwendungen** ermöglicht den Zugriff auf diverse Lizenzprogramme, die zum Ausführen bestimmter Arbeiten benötigt werden.

Uhr



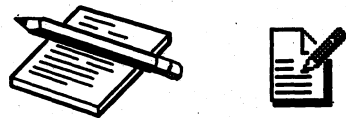
Mit dem Symbol **Uhr** wird das Programm **xclock** ausgeführt und eine Uhr angezeigt.

Demo Desktop



Mit dem Symbol **Demo Desktop** wird die Arbeitsoberfläche Demo Desktop geöffnet, die den Zugriff auf Demonstrationssoftware ermöglicht.

Editor



Mit dem Symbol **Editor** können Benutzer Textdateien erstellen und editieren.

Getting Started



Mit dem Symbol **Getting Started** wird ein interaktives Lernprogramm auf der Arbeitsoberfläche ausgeführt.

Home



Mit dem Symbol **Home** wird der Zugriff auf das Verzeichnis **Home** ermöglicht.

InfoExplorer



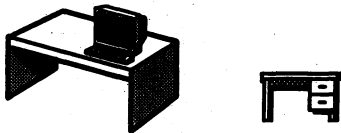
Mit dem Symbol **InfoExplorer** wird der Zugriff auf InfoExplorer, eine Online-Version der Betriebssystemdokumentation, ermöglicht.

Mail



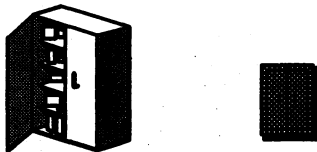
Mit dem Symbol **Mail** wird eine einfache Methode zum Empfangen von Nachrichten und Dateien von anderen Benutzern und zum Senden von Nachrichten und Dateien an andere Benutzer zur Verfügung gestellt.

Personal Desktop



Mit dem Symbol **Personal Desktop** wird der Zugriff auf eine andere Arbeitsoberfläche ermöglicht, auf die Dateien und Verzeichnisse zu einem bestimmten Projekt gestellt werden können.

Zubehör



Mit dem Symbol **Zubehör** wird der Zugriff auf Standardanwendungen wie Uhr, Taschenrechner und Editor zur Verfügung gestellt.

Papierkorb



Mit dem Symbol **Papierkorb** wird der Zugriff auf archivierte Dateien und Verzeichnisse sowie auf eine Funktion zum endgültigen Löschen von Dateien und Verzeichnissen ermöglicht.

Primärdrucker



Mit dem Symbol **Primärdrucker** können Benutzer Dateien drucken.

Sonderdateien des General Desktop

Dateien mit den folgenden Erweiterungen haben besondere Funktionsweisen und Symbole:

Datei	Beschreibung
.Z	Bei Aktivierung dekomprimieren.
.px	Bei Aktivierung in den Pixel-Abbildeditor geladen. Zeigt den Inhalt der Datei an.
.c, .o	Zeigen nur spezielle Symbole an.

Sichtbare Dateien des General Desktop

Zur besseren Übersichtlichkeit des General Desktop werden einige der auf dem System vorhandenen Dateien nicht angezeigt. Diese nicht sichtbaren Dateien werden für erweiterte Funktionen verwendet, die ein Benutzer mit geringen Kenntnissen gewöhnlich nicht ausführt. Dateinamen, die mit "." beginnen, z. B. **.mwmrc**, werden im allgemeinen von fortgeschrittenen Benutzern verwendet.

Folgende Dateien sind für den Benutzer des General Desktop nicht sichtbar:

- Geistdateien



Wurde eine Datei seit dem letzten Aufrufen der Arbeitsoberfläche gelöscht, erscheint das Geistsymbol.

- Dateien, deren Namen mit "." beginnen

Es gibt zwei Ausnahmen. Die Datei "." im **Benutzerverzeichnis** wird "Punktverzeichnis" genannt. Das Fenster enthält außerdem zwei Symbole mit der Bezeichnung . (Punkt) bzw. ... Das Symbol . steht für das Verzeichnis selbst, das Symbol .. für das Väterverzeichnis. Im **Benutzerverzeichnis** wird außerdem ".." nicht angezeigt, um den Zugriff auf das Stammdateisystem zu verhindern.

Zubehörsymbole des General Desktop

Zubehör ist ein Verzeichnis, das nützliche AIXwindows Desktop-Anwendungen enthält. Nachfolgend werden Beispiele für Anwendungen beschrieben, die sich im allgemeinen im Verzeichnis **Zubehör** befinden:

Taschenrechner



Mit dem Symbol **Taschenrechner** kann ein Taschenrechner aufgerufen werden, mit dem sämtliche Rechenoperationen ausgeführt werden können.

Details



Mit dem Symbol **Details** können Informationen über die aktuellen Arbeitsflächenbenutzer und die aktuelle Umgebung abgerufen werden.

Diskette



Mit dem Symbol **Diskette** können Daten auf Disketten geschrieben und von Diskette zurückgeholt werden. Außerdem kann der Inhalt der Diskette angezeigt werden.

Datei suchen



Mit dem Symbol **Datei suchen** können eine oder mehrere Dateien im Dateiablagensystem gesucht und auf einer Arbeitsoberfläche oder in einem Fenster angezeigt werden.

Text suchen



Mit dem Symbol **Text suchen** können angegebene Zeichenfolgen in einer oder mehreren Dateien gesucht und ihre Positionen in einem Fenster angezeigt werden.

Who



Mit dem Symbol **Who** werden Benutzer identifiziert, die am System angemeldet oder für das System konfiguriert sind und als Symbole auf einer neuen Arbeitsoberfläche angezeigt. Über jedes dieser Symbole können verschiedene dem Benutzer zugeordnete Aufgaben ausgeführt werden. Außerdem können diese Symbole als Parameter für weitere Anwendungen verwendet werden.

8-mm-Band

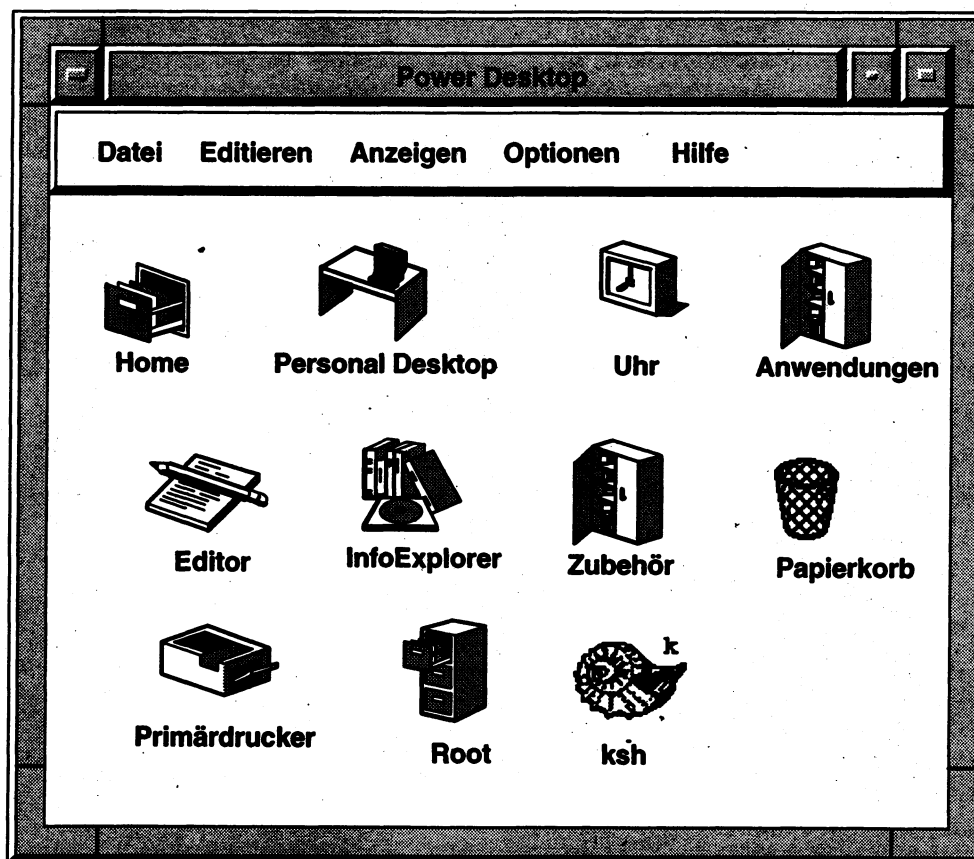


Mit dem Symbol **8-mm-Band** können Daten auf ein 8-mm-Band geschrieben und von einem 8-mm-Band zurückgeholt werden und der Inhalt eines 8-mm-Bands angezeigt werden.

Power Desktop

Die Arbeitsoberfläche Power Desktop erfüllt die Anforderungen von Benutzern mit umfassenden EDV- Kenntnissen, die mit dem UNIX-Dateisystem und dessen Möglichkeiten vertraut sind. Das Power Desktop umfaßt alle Funktionen des General Desktop und bietet darüber hinaus zusätzliche Funktionen für den fortgeschrittenen Benutzer. Das Power Desktop erleichtert den Zugriff auf eine Shell und beinhaltet Regeln für viele in `/usr/bin` und dem Stammverzeichnis (**Root**) enthaltene Dienstprogramme.

Die Arbeitsoberfläche Power Desktop und einige seiner Symbole werden in der folgenden Abbildung dargestellt.



Power Desktop mit großen Symbolen

Die folgenden Abschnitte enthalten detaillierte Informationen zum Power Desktop:

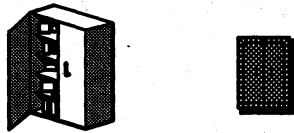
- Symbole des Power Desktop auf Seite 5-46.
- Sonderdateien des Power Desktop auf Seite 5-48.
- Sichtbare Dateien des Power Desktop auf Seite 5-49.
- Symbole des Verzeichnisses "Power" auf Seite 5-49.

Symbole des Power Desktop

Das Power Desktop erfüllt Anforderungen von Benutzern mit detaillierten EDV- Kenntnissen. Es stellt diesen leistungsstarke Hilfsprogramme und Funktionen zur Verfügung. Es folgt eine Beschreibung einiger der Symbole des Power Desktop.

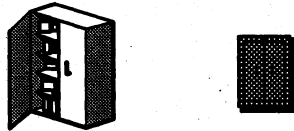
Hinweis: Manche Funktionen können sowohl durch ein großes als auch durch ein kleines Symbol dargestellt werden.

Anwendungen



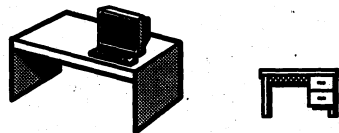
Das Symbol **Anwendungen** ermöglicht den Zugriff auf diverse Lizenzprogramme, die zum Ausführen bestimmter Arbeiten benötigt werden.

C-Programm



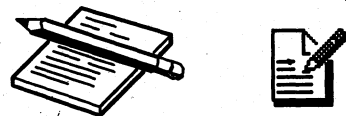
Mit dem Symbol **C-Programm** wird der Zugriff auf ein Verzeichnis zur Verfügung gestellt, das Hilfsprogramme für das Schreiben von C-Programmen enthält.

Demo Desktop



Mit dem Symbol **Demo Desktop** wird die Arbeitsoberfläche Demo Desktop geöffnet, über die Zugriff auf Demonstrationssoftware besteht.

Editor



Mit dem Symbol **Editor** können Benutzer Textdateien erstellen und editieren.

Getting Started



Mit dem Symbol **Getting Started** wird ein interaktiver Überblick über das Programm AIXwindows Desktop aufgerufen.

Home



Mit dem Symbol **Home** wird der Zugriff auf das **Benutzerverzeichnis** ermöglicht.

InfoExplorer



Über das Symbol **InfoExplorer** besteht eine Möglichkeit zum einfachen Zugriff auf InfoExplorer, eine Online-Version der Betriebssystemdokumentation.

ksh



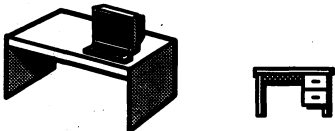
Mit dem Symbol **ksh** wird der Zugriff auf den Befehl **ksh** ermöglicht, mit dem das Programm **Korn Shell** in einem Datenstationsfenster ausgeführt wird.

Mail



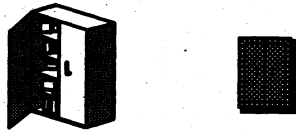
Mit dem Symbol **Mail** wird eine einfache Methode zum Empfangen von Nachrichten und Dateien von anderen Benutzern und Senden von Nachrichten oder Dateien an andere Benutzer zur Verfügung gestellt.

Personal Desktop



Mit dem Symbol **Personal Desktop** besteht Zugriff auf eine weitere Arbeitsoberfläche, auf die der Benutzer die Dateien und Verzeichnisse zu einem bestimmten Projekt stellen kann.

Power



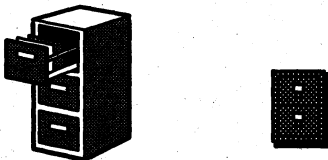
Das Symbol **Power** enthält leistungsstarke Systemhilfsprogramme und Anwendungen.

Primärdrucker



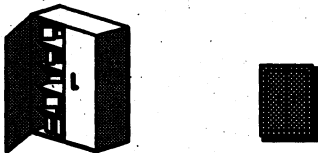
Mit dem Symbol **Primärdrucker** können Benutzer Dateien drucken.

Root



Mit dem Symbol **Root** wird der Zugriff auf das Stammverzeichnis (**Root**) ermöglicht.

Zubehör



Mit dem Symbol **Zubehör** wird der Zugriff auf Standardanwendungen wie Taschenrechner, Uhr und Editor ermöglicht.

Papierkorb



Das Symbol **Papierkorb** ermöglicht den Zugriff auf archivierte Dateien und Verzeichnisse. Außerdem können damit Dateien und Verzeichnisse endgültig gelöscht werden.

Sonderdateien des Power Desktop

Die Arbeitsoberfläche Power Desktop verfügt über besondere Symbole für **.c-** und **.o-**Dateien. Die folgenden speziellen Aktivierungsoptionen stehen für **.c-**Dateien zur Verfügung:

- Zum Kompilieren einer **.c-**Datei, mit der mittleren Maustaste auf der **.c-**Datei doppelt klicken.
- Zum Kompilieren einer **.c-**Datei und Angeben von Optionen mit der rechten Maustaste auf der **.c-**Datei doppelt klicken.

Sichtbare Dateien des Power Desktop



Dem Benutzer des Power Desktop werden *alle* Dateien angezeigt. Geistdateien werden mit dem **Geistsymbol** dargestellt.

Symbole des Verzeichnisses "Power"

Das Verzeichnis **Power** enthält nützliche vordefinierte Anwendungen. Nur der Systemadministrator hat Schreibberechtigung für dieses Verzeichnis. Im Verzeichnis **Power** befinden sich u. a. die folgenden Anwendungen:

Komprimieren



Mit dem Symbol **Komprimieren** werden Dateien komprimiert, die an dieses Symbol übergeben werden. Die Namen komprimierter Dateien erhalten das Suffix **.Z** und ihre Symbole werden in das Symbol **Komprimieren** umgewandelt.

Symboleditor



Mit dem Symbol **Symboleditor** besteht Zugriff auf den Pixel-Abbildeditor, mit dem Symbole erstellt oder geändert werden können.

man



Mit dem Symbol **man** besteht Zugriff auf die Dokumentation zu Befehlen.

Objekt öffnen



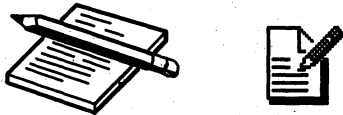
Mit dem Symbol **Objekt öffnen** wird ein Objekt "obj." als Verzeichnisfenster geöffnet. Der Benutzer kann neue Objekte erstellen oder vorhandene Objekte ändern.

Prozesse



Mit dem Symbol **Prozesse** wird eine neue Arbeitsoberfläche im Verzeichnis **/tmp** erstellt und für jeden Prozeß ein Symbol auf der Arbeitsoberfläche angezeigt. Auf der neuen Arbeitsoberfläche wird ein Symbol **kill** angezeigt. Prozesse, die an das Symbol **kill** übergeben werden, werden gestoppt.

vi



Das Symbol **vi** ermöglicht den Zugriff auf den Editor **vi**, mit dem der Benutzer Textdateien erstellen und editieren kann.

X-Programme



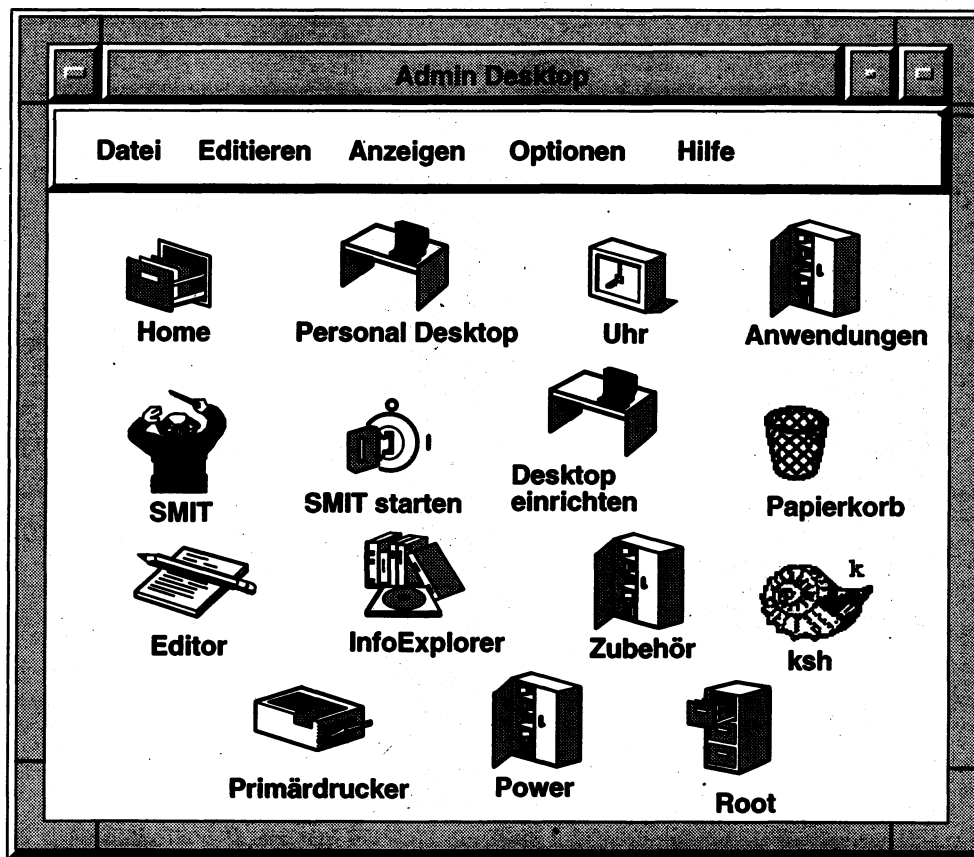
Mit dem Symbol **X-Programme** wird ein Verzeichnisfenster geöffnet, das **X11-Programme** enthält (**/usr/bin/X11** oder ein äquivalentes Programm).

Admin Desktop

Die Arbeitsoberfläche Admin Desktop enthält Hilfsprogramme zum Ausführen von Systemverwaltungsfunktionen. Sie ermöglicht dem Systemadministrator das Definieren von Benutzern und Verwalten von Druckern, Einheiten, Ressourcen und Dateien bei AIXwindows Desktop.

Über das Admin Desktop besteht Zugriff auf alle Hilfsprogramme und Dateien zum Verwalten von AIXwindows Desktop und zum Erstellen neuer, benutzerspezifischer Umgebungen. Diese Arbeitsoberfläche hat Informationen über die verwendete Sprachvariable **\$LANG** und vereinfacht das Erstellen von Regeln, da sie den Zugriff auf die richtigen Dateien ermöglicht und diese anzeigt.

Das Admin Desktop wird mit einigen der zugehörigen Symbole in folgender Abbildung dargestellt.



Admin Desktop mit großen Symbolen

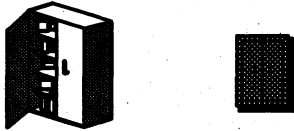
Nähere Informationen zum Admin Desktop enthalten die folgenden Abschnitte:

- Symbole des Admin Desktop auf Seite 5-52.
- Sichtbare Dateien des Admin Desktop auf Seite 5-55.
- Symbole des Verzeichnisses "Admin" auf Seite 5-55.
- Neue Benutzer definieren auf Seite 5-57.

Symbole des Admin Desktop

Über die Arbeitsoberfläche Admin Desktop besteht Zugriff auf Hilfsprogramme und Dateien, die zum Verwalten von AIXwindows Desktop benötigt werden. Außerdem wird dem Systemadministrator das Anpassen neuer Arbeitsoberflächen ermöglicht. Es folgt eine Beschreibung einiger Symbole des Admin Desktop.

Admin



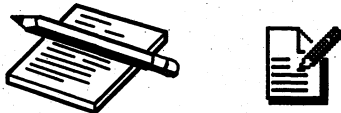
Mit dem Symbol **Admin** besteht Zugriff auf Anwendungen für den Systemadministrator.

Anwendungen



Mit dem Symbol **Anwendungen** besteht Zugriff auf verschiedene Lizenzprogramme, die zum Ausführen bestimmter Arbeiten benötigt werden.

Editor



Mit dem Symbol **Editor** können Benutzer Textdateien erstellen und editieren.

Getting Started



Mit dem Symbol **Getting Started** wird ein interaktiver Überblick über das Programm AIXwindows Desktop aufgerufen.

Home



Mit dem Symbol **Home** wird der Zugriff auf das Benutzerverzeichnis (**Home**) ermöglicht.

InfoExplorer



Über das Symbol **InfoExplorer** besteht Zugriff auf InfoExplorer, eine Online-Version der Betriebssystemdokumentation.

ksh



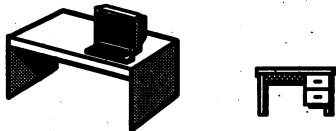
Das Symbol **ksh** ermöglicht den Zugriff auf den Befehl **ksh**, der das Programm **Korn Shell** in einem Datenstationsfenster ausführt.

Mail



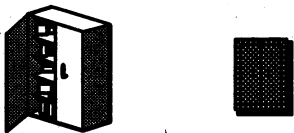
Das Symbol **Mail** erleichtert das Senden von Nachrichten und Dateien an andere Benutzer.

Personal Desktop



Über das Symbol **Personal Desktop** besteht Zugriff auf eine andere Arbeitsoberfläche, auf die der Benutzer Dateien und Verzeichnisse zu einem bestimmten Projekt stellen kann.

Power



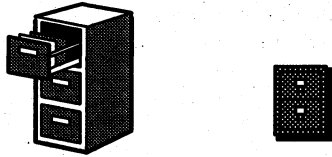
Das Symbol **Power** enthält leistungsstarke Hilfsprogramme und Anwendungen.

Primärdrucker



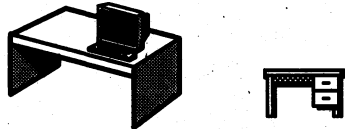
Mit dem Symbol **Primärdrucker** können Benutzer Dateien drucken.

Root



Über das Symbol **Root** besteht Zugriff auf das Stammverzeichnis (**Root**).

Desktop einrichten



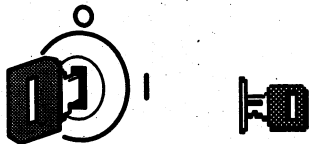
Das Hilfsprogramm **Desktop einrichten** ermöglicht Systemadministratoren das Definieren neuer Benutzer.

SMIT



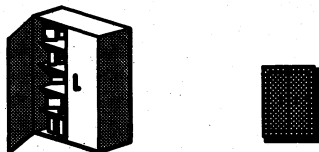
Das Symbol **SMIT** (System Management Interface Tool) unterstützt Systemadministratoren beim Ausführen von Systemverwaltungsaufgaben wie der Verwaltung von Dateisystemen und Einheiten, dem Hinzufügen neuer Benutzer und der Installation und Pflege von Software.

SMIT starten



Mit dem Symbol **SMIT starten** wird ein Lizenzprogramm installiert.

Zubehör



Über das Symbol **Zubehör** besteht Zugriff auf Standardanwendungen wie Taschenrechner, Uhr und Editor.

Papierkorb



Das Symbol **Papierkorb** ermöglicht den Zugriff auf archivierte Dateien und Verzeichnisse. Außerdem bietet es eine Funktion zum endgültigen Löschen von Dateien und Verzeichnissen.

xdtuserinfo



Über das Symbol **xdtuserinfo** kann auf die Datei **xdtuserinfo** zugegriffen werden.

Sichtbare Dateien des Admin Desktop

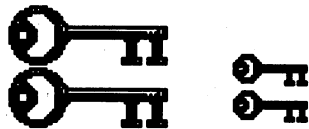


Dem Benutzer des Admin Desktop werden *alle* Dateien angezeigt. Geistdateien werden mit dem **Geistsymbol** angezeigt.

Symbole des Verzeichnisses "Admin"

Das Verzeichnis **Admin** enthält Anwendungen für den Systemadministrator. Nur der Systemadministrator hat Schreibberechtigung für dieses Verzeichnis. Im Verzeichnis **Admin** befinden sich u. a. folgende Hilfsprogramme:

Kennwort



Mit dem Symbol **Kennwort** wird das Programm **password** ausgeführt. Es erleichtert dem Systemadministrator das Ändern aktueller Benutzerkennwörter oder das Anlegen eines Kennworts für einen neuen Benutzer.

Benutzer hinzufügen



Mit dem Symbol **Benutzer hinzufügen** kann ein neuer Benutzer zum System hinzugefügt werden.

Regeln



Mit dem Symbol **Regeln** wird das aktuelle Regeldateiverzeichnis für die aktuelle **\$LANG** geöffnet.

Neue Benutzer definieren

Mit dem Befehl `xdt3newuser` kann der Systemadministrator neue Benutzer definieren. Durch Verwenden des Befehls `xdt3newuser` wird sichergestellt, daß Systemadministratoren bei der Definition neuer Benutzer keine wichtigen Schritte auslassen. Soll ein neuer Benutzer definiert werden, `xdt3newuser` in der Befehlszeile eingeben oder auf dem Symbol **Desktop einrichten** im Verzeichnis **Admin** doppelt klicken.

Erforderliche Schritte

Zum Definieren eines neuen Benutzers und Einrichten der Umgebung für die Standardarbeitsoberfläche führt das System den Systemadministrator durch eine Reihe von Fragen. Die drei Umgebungstypen für die Arbeitsoberfläche sind:

- General Desktop** Für Benutzer mit geringen UNIX-Kenntnissen.
- Power Desktop** Für erfahrene Programmierer. Über diese Arbeitsoberfläche besteht Zugriff auf leistungsfähige Betriebssystemfunktionen.
- Admin Desktop** Für Systemadministratoren. Diese Arbeitsoberfläche umfaßt auch alle Funktionen des Power Desktop.

Abschließend wird zum Eingeben der Namen der neuen Benutzer aufgefordert:

Namen der einzurichtenden Benutzer eingeben (Eingabetaste drücken, wenn keine Benutzer eingerichtet werden sollen).

An dieser Stelle eine Liste der Anmeldenamen der Benutzer eingeben, die für AIXwindows Desktop definiert werden sollen, und die Eingabetaste drücken. Die einzelnen Namen in der Liste durch Leereichen voneinander trennen. Wird die Eingabetaste gedrückt, ohne daß ein Benutzername eingegeben wurde, wird das Programm beendet. Andernfalls fordert das System zum Eingeben weiterer Namen auf.

Den codierten Zeichensatz ISO8859 installieren

Die folgenden Schritte können nur von Benutzern mit Superuser-Berechtigung ausgeführt werden. Zum Installieren des codierten Zeichensatzes ISO8859 `xdt3newuser` eingeben oder auf dem Symbol **Desktop einrichten** im Verzeichnis **Admin** doppelt klicken.

Erforderliche Schritte

1. `xdt3newuser` zeigt folgende Eingabeaufforderung an:

Sollen die AIXwindows Desktop-Dateien für den Codesatz 8859 erstellt/neu installiert werden?

2. ja auswählen. Das System zeigt dann folgende Eingabeaufforderung an:

Welche Sprache soll erstellt/neu installiert werden?:

Es wird eine Liste der am System installierten Sprachen angezeigt.

`xdt3newuser` installiert die Arbeitsoberfläche und zeigt die Verzeichnisse an, während sie umgewandelt werden. Die Prozedur `xdt3newuser` wird fortgesetzt.

Kapitel 6. Netze und TCP/IP

Dieses Kapitel definiert Begriffe, Komponenten und Anwendungen für physische und logische Netze und liefert des weiteren Informationen und Prozeduren zum TCP/IP (Transmission Control Protocol/Internet Protocol).

Ein Netz ist die Kombination von zwei oder mehr Computern und den Verbindungsleitungen dazwischen. Das *physische* Netz ist die Hardware (Geräte wie Computer, Kabel und Telefonleitungen), aus der ein Netz besteht. Die Software und andere Steuereinheiten und Konventionen bilden das *logische* Netz.

Das TCP/IP ist eine Protokollgruppe, die Datenfernverarbeitungsstandards zwischen Computern und Detailvereinbarungen für die Leitwegwahl und die Verbindung zwischen den Netzen angibt. Es wird hauptsächlich auf dem DARPA-Internet verwendet und ermöglicht somit Forschungsinstituten, Colleges, Universitäten, der Regierung und der Industrie, miteinander zu kommunizieren.

Die folgenden Abschnitte bieten einen Überblick über Netze und das TCP/IP:

- Überblick über Netze
- Überblick über das TCP/IP
- Einführung in das TCP/IP und Internet
- Einführung in die Verwendung von TCP/IP-Netzen
- Einführung in TCP/IP-Befehle für Endbenutzer
- Einführung in den Befehl telnet
- Einführung in den Befehl rlogin
- Einführung in die Befehlsausführung auf einem fernen System

Die folgenden Abschnitte erläutern die Prozeduren für netzbezogene Aufgaben:

- Informationen über angemeldete Benutzer anzeigen
- Mit dem Befehl ftp Dateien kopieren
- Von einem fernen System aus drucken
- An einem fernen System drucken
- Dialog mit einem Benutzer auf einem fernen System
- Einen lokalen Host mit einem fernen Host verbinden
- Mit dem Befehl tftp Dateien kopieren

Das Handbuch *Communication Concepts and Procedures* enthält weitere Informationen über TCP/IP und anderen mit Netzen zusammenhängenden Themen.

Überblick über Netze

Ein Netz ist die Kombination von zwei oder mehr Computern und den Verbindungsleitungen dazwischen. Das *physische* Netz ist die Hardware (Geräte wie Computer, Kabel und Telefonleitungen), aus der ein Netz besteht. Die Software und andere Steuereinheiten und Vereinbarungen bilden das *logische* Netz.

Dieser Überblick bietet die folgenden Informationen zu Netzen:

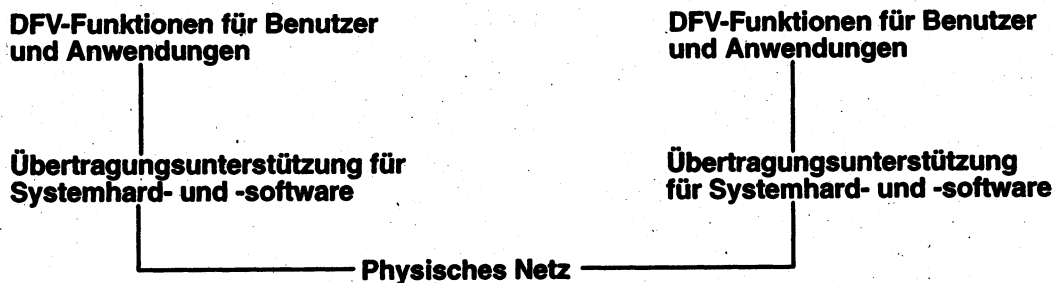
- Einführung in Netze
- Physische Netze (auf Seite 6-3)
- Übertragungsunterstützung für Systemhard- und -software (auf Seite 6-3)
- DFV-Funktionen für Benutzer und Anwendungen (auf Seite 6-6)
- Mit anderen Betriebssystemen kommunizieren (auf Seite 6-6).

Einführung in Netze

Netz ist ein häufig verwendeter, oft doppeldeutiger Begriff, dessen Bedeutung im allgemeinen kontextabhängig ist. Dieser Netzüberblick unterscheidet zwischen drei generellen Ebenen:

- Auf der obersten Ebene DFV-Funktionen für Benutzer und Anwendungen
- Auf der mittleren Ebene Übertragungsunterstützung für Systemhard- und -software
- Auf der unteren Ebene physische Netze

In Kombination bilden diese Ebenen ein Kommunikationsnetz.



Sowohl einzelne Benutzer als auch Organisationen verwenden Netze aus vielen Gründen. Einige Verwendungsmöglichkeiten sind:

- Datenerfassung
- Abfrage
- Datensatzaktualisierung
- Stapelferneingabe
- gemeinsamer Ressourcenzugriff
- gemeinsame Datenbenutzung
- elektronische Post

Die Datenerfassung besteht aus der Eingabe von Daten direkt entweder in lokale oder ferne Datendateien, wodurch notwendige Zwischenschritte wie Übergabe, Aufzeichnen und Übertragen wegfallen. Höhere Genauigkeit und Effizienz sind natürliche Nebenprodukte einer solchen Datenübertragung in einem Schritt. Die Abfrage beinhaltet das Suchen in Datendateien nach bestimmten Informationen. Die Datensatzaktualisierung betrifft das Ändern, Hinzufügen oder Löschen von Daten, die in lokalen oder fernen Dateien gespeichert sind. Die Stapelferneingabe besteht aus der Eingabe von Stapeldaten an einem fernen Standort. Wegen des weitreichenden Leistungsspektrums sind Datenfernverarbeitung und Netze nicht nur wünschenswert, sondern notwendig.

Der gemeinsame Ressourcenzugriff ist eine weitere Funktion von Netzen. Die Benutzer können nicht nur Daten, sondern auch Programme, Dateispeicherplatz und periphere Einheiten (wie z. B. Drucker, Modems, Datenstationen, Festplatten und Fernschreiber) gemeinsam benutzen. In einer verteilten Umgebung ist es häufig untragbar, für jeden Benutzer einen Drucker bereitzustellen, oder (wegen des begrenzten Speicherplatzes) dieselbe Datei auf jeden Knoten zu stellen.

Ein Kommunikationsnetz ermöglicht vielen Benutzern, Dateien, die auf einem Datei-Server gespeichert sind, wirtschaftlich und effizient zu nutzen und wenn nötig, einen Einzeldrucker oder ein Modem zur Datenübertragung zu verwenden. Die elektronische Post ermöglicht zwei Benutzern auf unterschiedlichen Systemen, über das physische Netz miteinander zu kommunizieren.

Physische Netze

Im physischen Netz werden die unterschiedlichen Hardwarekomponenten des Netzes mit Kabeln (Standard-Ethernet, IEEE 802.3 Ethernet, Token-Ring und Telefonleitungen) miteinander verbunden.

Physische Netze sind sowohl in der Größe unterschiedlich als auch in der verwendeten Hardware. Die zwei häufigsten Arten von Netzen sind das *lokale Netz* (LAN) und das *öffentliche Netz* (WAN). Ein LAN ist ein Netz, in dem die Datenfernverarbeitung auf ein relativ kleines geographisches Gebiet (1-10 km) begrenzt ist, z. B. ein Bürogebäude, Warenhaus oder Universitätsgelände. Ein WAN ist ein Netz, dessen Fähigkeiten zur Datenübertragung über größere geographische Gebiete reichen als beim LAN; z. B. eine Stadt, ein Bundesland oder Staat.

LANs verwenden normalerweise Standard-Ethernet-, IEEE 802.3 Ethernet- oder Token-Ring-Hardware für das physische Netz, wohingegen WANs und asynchrone Netze Kommunikationsnetze verwenden, die von Fernmeldegesellschaften angeboten werden. Der Betrieb eines physischen Netzes ist normalerweise durch Netzbetriebsstandards geregelt, zum Beispiel EIA232D oder CCITT V.35.

Übertragungsunterstützung für Systemhard- und -software

Jede Datenfernverarbeitung im Netz beinhaltet die Verwendung von Hardware und Software. Die *Hardware* besteht aus den Geräten, die zum physischen Netz verbunden sind. Die *Software* besteht aus den Programmen, Prozeduren, Regeln oder Protokollen und der dazugehörigen Dokumentation, die sich auf den Betrieb eines bestimmten Systems beziehen. Die Übertragungsunterstützung für die Hard- und Software hängt von der verwendeten Hardware ab und von der Software, die zum Betrieb dieser Hardware und der Schnittstelle mit dem Netz erforderlich ist.

Zur Systemhardware gehören Adapterkarten, die den Pfad oder die Schnittstelle zwischen der Systemsoftware und dem physischen Netz bereitstellen. Eine Adapterkarte benötigt einen Ein-/Ausgabekartensteckplatz am System. Die Adapterkarte verbindet die *Datenendeinrichtung* (DTE) mit der *Datenübertragungseinrichtung* (DCE); d. h., sie liefert die physische lokale Adresse für einen DTE-Anschluß.

Eine Adapterkarte bereitet alle hereinkommenden und abgehenden Daten auf, führt Suchen nach Adressen aus, stellt Treiber, Empfänger und Überspannungsschutz zur Verfügung, unterstützt unterschiedliche Schnittstellen und übernimmt im allgemeinen viele DFV-Aufgaben des Systemprozessors. Adapterkarten unterstützen die Standards, die für das physische Netz erforderlich sind (z. B. EIA232D, Smart Modem, V.25 bis, EIA422A, X.21 oder .35), und gleichzeitig unterstützen sie *Softwareprotokolle* (z. B. SDLC-, HDLC- und bisynchrone Protokolle).

Protokolle

Alle DFV-Softwarekomponenten verwenden Protokolle, d. h. Gruppen von semantischen und syntaktischen Regeln, die das Verhalten der Funktionseinheiten beim Datenaustausch festlegen. Protokolle definieren, wie Informationen geliefert werden, wie sie ihre Zieladresse sicher erreichen, und welchem Pfad sie folgen. Protokolle regeln außerdem den Fluß von Nachrichten und deren Empfangsbestätigungen; z. B., ob eine Antwort erwartet wird.

Datenaustauschprotokolle legen die Sendepriorität in einem System fest, so daß Informationen nicht verlorengehen. Es ist wichtig zu wissen, wer gerade sendet und wer als nächstes senden wird. Wenn Benutzer zur gleichen Zeit senden, können Informationen verlorengehen.

Schließlich gibt es noch Protokolle, die festlegen, wie Daten formatiert und ausgegeben werden. Verwenden beispielsweise Absender und Empfänger die gleiche Verschlüsselung? Sprechen sie dieselbe Sprache oder ist eine Umsetzung erforderlich?

Protokolle existieren auf unterschiedlichen Ebenen innerhalb des Kernel und können nicht direkt bearbeitet werden. Sie werden jedoch indirekt bearbeitet durch die Auswahl, die der Benutzer auf der Ebene der Anwendungsprogrammchnittstelle (API) trifft. Die Auswahl, die der Benutzer trifft, wenn er die Dateiübertragung, die Fernanmeldung oder das Emulationsprogramm für die Datenstation aufruft, definiert die Protokolle, die zur Ausführung dieser Programme verwendet werden.

Adressen

Ein drittes Merkmal, das Kommunikationsnetze haben, sind *Adressen*. Adressen sind sowohl Soft- als auch Hardware zugeordnet. Die Adresse ist das Mittel, über das die sendende oder Leitstation die Station auswählt, an die die Daten gesendet werden. Im Prinzip sind Adressen das Mittel, um Empfangsstationen und Speicherstellen zu identifizieren. Eine physischen Adresse ist ein eindeutiger Code, der jeder Einheit oder Datenstation, die mit einem Netz verbunden ist, zugeordnet ist.

In einem Token-Ring-Netz beispielsweise zeigt der Befehl **netstat -iv** die Token-Ring-Kartenadresse an. Dies ist die physische Netzadresse. Der Befehl **netstat -iv** zeigt außerdem Adreßinformationen auf Klassen- und Benutzerebene an. Adressen werden oft durch die Software definiert, können aber auch vom Benutzer selbst erstellt werden.

Domänen

Ein Aspekt von Adressen, der vielen Kommunikationsnetzen gemeinsam ist, ist das Konzept der *Domänen*. Beispielsweise zeigt die Struktur des Internet, wie über Domänen die Internet-Protokolladressen definiert werden. Das Internet ist ein umfangreiches Netz, das viele verschiedene kleinere Netze umfaßt. Um die Leitwegwahl und Adressierung zu erleichtern, sind Internet-Adressen hierarchisch in Domänen strukturiert, mit sehr weit gefaßten Kategorien an der Spitze, wie z. B. **com** für kaufmännische Benutzer, **edu** für Benutzer aus dem Bildungssektor und **gov** für Benutzer aus Regierungskreisen.

Innerhalb der Domäne **com** befinden sich viele kleinere Domänen, die sich auf einzelne Geschäftsbetriebe beziehen; zum Beispiel **ibm**. Innerhalb der Domäne **ibm.com** befinden sich noch kleinere Domänen, die sich ihrerseits auf die Internet-Adressen der verschiedenen Standorte beziehen, wie **austin[150q.ibm.com** oder **raleigh.ibm.com**. Ab dieser

Ebene sind die Namen von *Hosts* zu finden. Ein Host ist in diesem Zusammenhang jeder Computer, der mit dem Netz verbunden ist. Innerhalb der Domäne `austin.ibm.com` können sich Hosts mit den Namen `hamlet` und `lear` befinden, die mit `hamlet.austin.ibm.com` und `lear.austin.ibm.com` adressiert werden.

Gateways und Brücken

Auf dem Internet befindet sich eine Vielzahl von Netzen, die oft unterschiedliche Hard- und Softwarekomponenten verwenden. *Gateways* und *Brücken* ermöglichen diesen unterschiedlichen Netzen, miteinander zu kommunizieren. Brücken werden auf LANs verwendet. Eine Brücke ist eine Funktionseinheit, die zwei LANs miteinander verbindet, die möglicherweise dieselbe Prozedur zur Steuerung logischer Verbindungen (LLC), wie z. B. Ethernet verwenden, aber eine unterschiedliche Prozedur zur Datenträgerzugriffssteuerung (MAC). Ein Gateway hat eine größere Reichweite als eine Brücke. Es arbeitet oberhalb der Verbindungsschicht und setzt, falls erforderlich, die Schnittstelle und das Protokoll, das von dem einen Netz verwendet wird, in die vom anderen eindeutigem Netz verwendeten um. Gateways ermöglichen die Datenübertragung zwischen den verschiedenen Netzen, die das Internet bilden.

Leitwegwahl

Die Verwendung von Domännennamen für die Adressierung und von Gateways für die Umsetzung erleichtert die *Leitwegwahl* der Daten, die übertragen werden. Leitwegwahl ist die Zuordnung eines Pfades, über den eine Nachricht ihre Zieladresse erreicht. Über den Domännennamen wird die Zieladresse der Nachricht genau definiert. In einem großen Netz wie dem Internet werden die Daten von einem Kommunikationsnetz zum nächsten geleitet, bis sie die Zieladresse erreicht haben. Jedes Kommunikationsnetz überprüft den Domännennamen und leitet die Daten auf der Grundlage der Domänen, die diesem Netz bekannt sind, weiter bis zum nächsten logischen Stopp. Auf diese Weise trägt jedes Kommunikationsnetz, das die Daten empfängt, zum Prozeß der Leitwegwahl bei.

Lokale und ferne Knoten

Ein physisches Netz wird von den Hosts verwendet, die sich auf diesem Netz befinden. Jeder Host kann als *Knoten* in diesem Netz angesehen werden. Ein Knoten ist ein adressierbarer Standort in einem Kommunikationsnetz, der Hostverarbeitungsservices bietet. Die Verbindung dieser verschiedenen Knoten untereinander führt zu dem Konzept von *lokal* und *fern*. *Lokal* bezieht sich auf eine Einheit, eine Datei oder ein System, auf die/das direkt vom System des Benutzer aus zugegriffen wird, ohne eine DFV-Leitung zu verwenden. *Fern* bezieht sich auf eine Einheit, eine Datei oder ein System, auf die/das das System des Benutzers über eine DFV-Leitung zugreift. Lokale Dateien befinden sich auf dem System des Benutzers, während sich ferne Dateien auf einem Datei-Server befinden oder auf einem anderen Knoten, mit dem über ein physisches Netz kommuniziert wird, z. B. eine Ethernet-, Token-Ring- oder Telefonleitung.

Server und Klient

Verbunden mit dem Konzept von lokal und fern ist das Konzept von *Klient* und *Server*. Ein Server ist ein Computer, der Daten enthält oder Einsatzmittel bietet, auf die von anderen Computern im Netz zugegriffen werden kann. Die häufigsten Server-Typen sind Datei-Server, die Dateien speichern, Namens-Server, die Namen und Adressen speichern, und Code-Server, die Programme und Anwendungen speichern.

Ein Klient ist ein Computer, der Service oder Daten von einem Server anfordert. Ein Klient kann beispielsweise aktualisierten Programmcode oder die Verwendung von Anwendungen von einem Code-Server anfordern. Um Namen und Adressen zu erhalten, wendet sich der Klient an den Namens-Server. Ein Klient kann außerdem Dateien und Daten zur Datenerfassung, Abfrage oder Datensatzaktualisierung vom Datei-Server anfordern.

DFV-Funktionen für Benutzer und Anwendungen

Netze ermöglichen DFV-Funktionen für mehrere Benutzer und Anwendungen. Dem Benutzer stehen folgende Funktionen zur Verfügung:

- Elektronische Post senden
- Eine andere Datenstation emulieren oder an einem anderen Computer anmelden
- Daten übertragen
- Programme ausführen, die sich auf einem fernen Knoten befinden

Über das Kommunikationsnetz kann ein Benutzer mit einem anderen Benutzer mittels der elektronischen Post kommunizieren. Die tieferliegenden Schichten der Soft- und Hardware sowie das physische Netz ermöglichen dem Benutzer, Nachrichten, Briefe, Kurzinfos, Einladungen und Datendateien zu generieren und an einen beliebigen Benutzer auf dem physischen Netz zu senden oder sie von ihm zu empfangen und zu verarbeiten. Die elektronische Post verfügt über folgendes Leistungsspektrum: Nachrichtenanmerkung, Nachrichtensortierung, Packen von Nachrichten, Sortieren nach Datum und Postordnerfähigkeiten.

Das Emulieren eines anderen Computers ermöglicht dem Benutzer, auf Anwendungen in anderen Systemen in der Weise zuzugreifen, als ob er direkt an diesem System angeschlossen wäre. Über die Fernanmeldung ist eine ferne Anmeldung zwischen einem System unter AIX und einem anderen UNIX-System möglich.

Über ein File Transfer Protocol kann der Benutzer auf Dateien und Verzeichnisse auf einem fernen Host zugreifen. Dateien können vom und zum fernen Host übertragen werden. Ein Teil des Protokolls ist normalerweise der Kennwortschutz. Bei der Dateiübertragung besteht ein Klient/Server-Verhältnis zwischen dem Benutzer, der die Anforderung eingeleitet hat, und dem fernen System, auf das der Benutzer zugreift. Oft beinhaltet ein File Transfer Protocol auch Anzeige- und Steuerfunktionen, so daß ein Benutzer mit Schreib-/Lesezugriff Dateien und Verzeichnisse anzeigen, definieren oder löschen kann.

Einige unterschiedliche Protokolle sind entwickelt worden, um Benutzern und Anwendungen auf einem System zu ermöglichen, auf einem anderen System Prozeduren aufzurufen und Anwendungen auszuführen. Das kann für eine Reihe von Umgebungen sinnvoll sein, zum Beispiel, wenn es darum geht, in technischen und naturwissenschaftlichen Anwendungen viele Routinen mit hoher Computerauslastung zu entlasten.

Mit anderen Betriebssystemen kommunizieren

Unterschiedliche Computertypen können in einem Netz verbunden werden, d. h., die Computer können von unterschiedlichen Herstellern sein oder unterschiedliche Modelle eines Herstellers. Die Unterschiede in den Betriebssystemen von zwei oder mehr Computertypen können mit Programmen überbrückt werden.

Manchmal setzen diese Programme voraus, daß vorher ein anderes Programm auf dem Netz installiert wurde. Andere Programme wiederum setzen voraus, daß DFV-Konnektivitätsprotokolle wie das TCP/IP oder SNA auf dem Netz existieren.

Empfohlene Dokumentation

Zugehörige Informationen

Das Kapitel "TCP/IP Overview for System Management" im Handbuch *Communication Concepts and Procedures*.

Das Kapitel "BNU Overview for System Management" im Handbuch *Communication Concepts and Procedures*.

Das Kapitel "X.25 Communications Overview for System Management" im Handbuch *Communication Concepts and Procedures*.

Das Kapitel "Network File System (NFS) Overview for System Management" im Handbuch *Communication Concepts and Procedures*.

Das Kapitel "Mail Overview for System Management" im Handbuch *Communication Concepts and Procedures*.

Überblick über das TCP/IP

Das TCP/IP ist eine Protokollgruppe, die Datenfernverarbeitungsstandards zwischen Computern und Detailvereinbarungen für die Leitwegwahl und die Verbindung zwischen den Netzen angibt. Es wird hauptsächlich auf dem DARPA-Internet verwendet und ermöglicht somit Forschungsinstituten, Colleges, Universitäten, der Regierung und der Industrie, miteinander zu kommunizieren.

Wenn Computer miteinander kommunizieren, wird ihnen durch bestimmte Regeln oder *Protokolle* ermöglicht, Daten auf geregelte Art und Weise zu übertragen und zu senden. Die in der ganzen Welt am häufigsten verwendete Reihe von Protokollen ist das Transmission Control Protocol/Internet Protocol (TCP/IP). (In Europa ist jedoch das X.25-Protokoll verbreiteter.) Einige allgemeine Funktionen bei der Verwendung vom TCP/IP sind die elektronische Post, Dateiübertragung zwischen Computern und die Fernanmeldung.

TCP/IP und Internet

Lokale Netze sind darauf ausgelegt, den Bedürfnissen einer bestimmten Benutzergruppe gerecht zu werden. Zum Beispiel wird eine Universität ein System mit Hunderten von Hosts benötigen, damit Studenten ihre Arbeiten erledigen können, während Professoren ihre Forschungen fortsetzen. Dahingegen wird eine Computerfirma mit Büros im ganzen Land wohl eine Reihe hochleistungsfähiger Datenstationen benötigen, wenn sie einen neuen Computer entwickeln will. In beiden Fällen werden Systemmanager und Programmierer die Hardware und Übertragungsprotokolle entsprechend den bestimmten Bedürfnissen auswählen.

Dennoch wird es, gerade weil die Bedürfnisse bei Computernetzen so unterschiedlich sind, kein endgültiges Netzsystem geben. Im allgemeinen werden Netze in zwei Kategorien eingeteilt: lokale Netze und öffentliche Netze. *Lokale Netze (LANs)* sind normalerweise schneller als andere Typen von Netzen, aber in ihrer Reichweite begrenzt. *Öffentliche Netze* sind normalerweise langsamer als LANs, können aber Hosts in der ganzen Welt miteinander verbinden. Um einem Benutzer auf einem lokalen Host zu ermöglichen, trotz unterschiedlicher Hardware oder Netzkonfiguration mit fernen Netzen zu kommunizieren, wurde das *Internet*-Protokoll erstellt.

Defense Advanced Research Projects Agency (DARPA)

Während der siebziger Jahre und am Anfang der achtziger investierte die Defense Advanced Research Projects Agency (DARPA) in Forschungsprojekte, die Standards im Bereich der Computerkommunikation erstellen sollten und zwar durch das Festlegen von Vereinbarungen, wie Netze miteinander verbunden werden, wie Informationen gesendet und empfangen werden und wie Daten übertragen werden. Seitdem haben führende Forschungsinstitute, wie die National Science Foundation, das Department of Energy und die National Aeronautics and Space Administration die DARPA-Standards übernommen und fordern von ihren Vertragsnehmern, sie auch zu befolgen. Diese Standards, bekannt als das Transmission Control Protocol/Internet Protocol, sind heutzutage fast auf der ganzen Welt verbreitet.

Die folgenden Merkmale unterscheiden das TCP/IP von vorangegangenen Protokollen für miteinander verbundene Netze, die in ihrem Leistungsspektrum begrenzter waren:

- Unabhängigkeit von Netztechnologie. Das TCP/IP wurde so angepaßt, daß es mit fast jeder Hardware (Computertypen, Kabel, Telefonleitungen usw.) ausführbar ist, die nationale Standards erfüllen.

- Universelle Verbindung. Ein Computer in einem lokalen Netz kann mit einem anderen, entweder in dem gleichen oder in einem anderen Netz, kommunizieren, solange das lokale Netz entsprechend konfiguriert ist.
- Anwendungsprotokollstandards. Die Internet-Standards decken viele einheitliche Anwendungen ab, wie z. B. die elektronische Post, Dateiübertragung und die Fernanmeldung.

Das TCP/IP wurde erstellt, um die Lücken der vorangegangenen Protokolle zu schließen. Wenn sich die Anforderungen an Netze ändern, wird das TCP/IP sicherlich neue oder modifizierte Protokolle entwickeln.

Einführung in TCP/IP-Befehle für Endbenutzer

Das TCP/IP ist ein Teil der tieferliegenden Struktur des Benutzersystems. Es ermöglicht dem Benutzer, durch die bloße Ausführung eines Befehls oder Programms mit einer anderen Datenstation oder einem anderen System zu kommunizieren. Das System sorgt für die reibungslose Ausführung der Funktion.

Die TCP/IP-Befehle für Endbenutzer können grob in vier Kategorien eingeteilt werden:

- Dateiübertragung
- Fernanmeldung
- Status
- Fernverbindung .

Befehle zur Dateiübertragung

Obwohl über die elektronische Post die Möglichkeit besteht, relativ kleine Dateien zu senden, gibt es effizientere Methoden, größere Dateien zu übertragen. Die Befehle **ftp**, **rcp** und **tftp** basieren auf dem TCP/IP bei der Herstellung einer Direktverbindung vom lokalen Host zu einem fernen Host. Basic Network Utilities (BNU) können auch das TCP/IP verwenden, um Direktverbindungen zu fremden Hosts herzustellen.

Das TCP/IP enthält drei Befehle zur Dateiübertragung:

- | | |
|------------------------|---|
| ftp | Dieser Befehl verwendet das File Transfer Protocol (FTP), um Dateien zwischen Hosts zu übertragen, die ein unterschiedliches Dateisystem verwenden oder eine unterschiedliche Zeichendarstellung, z. B. EBCDIC und ASCII. Er stellt eine Sicherheitsfunktion zur Verfügung, indem Kennwörter zum fernen Host gesendet werden, und läßt die automatische Anmeldung, Dateiübertragung und Abmeldung zu. |
| rcp | Dieser Befehl kopiert eine oder mehrere Dateien von einem lokalen zu einem fernen Host oder von einem fernen zu einem anderen fernen Host oder auch innerhalb desselben fernen Hosts. Dieser Befehl ähnelt dem Befehl cp , außer daß er nur auf die ferne Dateiverarbeitung anwendbar ist. Falls für das Netz besondere Sicherheitsvorkehrungen erforderlich sind, kann der Befehl vom Systemmanager inaktiviert werden. |
| tftp oder utftp | Diese Befehle verwenden das Trivial File Transfer Protocol (TFTP), um Dateien von und zu Hosts zu übertragen. Da das TFTP ein Übertragungsprotokoll für Einzeldateien ist, bieten die Befehle tftp und utftp nicht alle Funktionsmerkmale des Befehls ftp . Falls für das Netz besondere Sicherheitsvorkehrungen erforderlich sind, kann der Befehl vom Systemmanager inaktiviert werden. |

Fernanmeldung

Aus diversen Gründen kann es erforderlich sein, auf einen anderen Computer zuzugreifen als auf den eigenen. Zum Beispiel könnte ein Systemadministrator die Berechtigung für eine kritische Datei, die vom Benutzer kürzlich bearbeitet wurde, neu zuordnen wollen, oder der Benutzer muß auf eine persönliche Datei auf der Datenstation eines anderen Benutzers zugreifen. Die Funktionen der Fernanmeldung, wie z. B. die Befehle **rlogin**, **rexec** und **telnet** ermöglichen dem lokalen Host, als Ein-/Ausgabedatenstation für den fernen Host zu dienen. Tastenanschläge werden zum fernen Host gesendet und die Ergebnisse werden am lokalen Bildschirm angezeigt. Wird die Sitzung der Fernanmeldung beendet, werden alle Funktionen zum lokalen Host zurückgegeben.

Das TCP/IP enthält die folgenden Befehle zur Fernanmeldung und zur Befehlsausführung:

- rexec** Der Befehl **rexec** bietet die Möglichkeit, Befehle interaktiv an verschiedenen fernen Hosts auszuführen, wenn sich der Benutzer mit dem Befehl **rlogin** an einem fernen Host angemeldet hat. Falls für das Netz besondere Sicherheitsvorkehrungen erforderlich sind, kann der Befehl vom Systemmanager inaktiviert werden.
- rlogin** Der Befehl **rlogin** bietet die Möglichkeit, sich an einem ähnlichen fremden Host anzumelden. Anders als der Befehl **telnet**, der auf unterschiedliche ferne Hosts anwendbar ist, kann der Befehl **rlogin** nur bei UNIX-Hosts verwendet werden. Falls für das Netz besondere Sicherheitsvorkehrungen erforderlich sind, kann der Befehl vom Systemmanager inaktiviert werden.
- rsh** oder **remsh** Die Befehle **rsh** und **remsh** bieten die Möglichkeit, Befehle auf ähnlichen fremden Hosts auszuführen. Alle erforderlichen Eingaben müssen vom fernen Host ausgeführt werden. Falls für das Netz besondere Sicherheitsvorkehrungen erforderlich sind, kann der Befehl vom Systemmanager inaktiviert werden.
- telnet, tn** oder **tn3270** Der Befehl **telnet** ruft ein Emulationsprogramm für die Datenstation auf, das das TELNET-Protokoll implementiert und dem Benutzer ermöglicht, sich an einem ähnlichen oder nicht ähnlichen fremden Host anzumelden.

Statusbefehle

Das TCP/IP enthält die folgenden Befehle, mit denen der Status von lokalen und fernen Host und deren Netzen festgestellt werden kann:

- finger** oder **f** Diese Befehle zeigen Informationen über die aktuellen Benutzer eines bestimmten Systems an. Diese Informationen können den Anmeldenamen des Benutzer, seinen vollständigen Namen und den Datenstationsnamen beinhalten sowie das Datum und die Zeit der Anmeldung.
- host** Dieser Befehl wandelt einen Hostnamen in eine Internet-Adresse um oder umgekehrt.
- ping** Dieser Befehl hilft bei der Bestimmung des Status eines Netzes oder Hosts. Er wird meistens verwendet, um zu prüfen, ob ein Netz oder ein Host aktiv ist.
- rwho** Dieser Befehl zeigt die Benutzer an, die an Hosts in einem lokalen Netz angemeldet sind. Dabei werden der Benutzername, der Hostname und das Datum und die Zeit der Anmeldung angezeigt.
- whois** Dieser Befehl identifiziert, zu wem eine Benutzer-ID oder ein Kurzname gehört. Er kann nur verwendet werden, wenn das lokale Netz mit dem

ARPA-Netz verbunden ist. Der Systemmanager kann darüber Auskunft geben, ob das System des Benutzers an dieses Netz angeschlossen ist.

Fernverbindung

Mit der elektronischen Post kann der Benutzer Textnachrichten an andere Benutzer des *lokalen* Netzes senden und von ihnen empfangen. Wenn das Datenverarbeitungssystem entsprechend konfiguriert ist und die passende elektronische Adresse bekannt ist, kann der Benutzer Nachrichten der elektronischen Post an andere Benutzer eines *fernen* Systems auf der ganzen Welt senden.

Das TCP/IP enthält die folgenden Befehle für die Fernverbindung:

- | | |
|-------------|---|
| mail | Dieser Befehl sendet und empfängt elektronische Briefe, Nachrichten und Kurzinformationen. |
| talk | Dieser Befehl ermöglicht dem Benutzer den interaktiven Dialog mit einem Benutzer auf einem fernen Host. |

Einführung in den Befehl telnet

Hinweis: Aus Gründen der Übersichtlichkeit bezieht sich **telnet** im folgenden auf die Befehle **telnet**, **tn** oder **tn3270**.

Mit dem Befehl **telnet** kann sich der Benutzer an einem fernen Host anmelden. Das wichtigste Merkmal des Befehls **telnet** ist, daß es sich dabei um einen *gesicherten* Befehl handelt. Ein gesicherter Befehl entspricht allen Sicherheitsstufen, die auf dem Computer konfiguriert sind. Im Gegensatz dazu handelt es sich bei dem Befehl **rlogin**, mit dem eine Fernanmeldung möglich ist, nicht um einen gesicherten Befehl. Bei Systemen, die besondere Sicherheitsvorkehrungen erfordern, sollte der Befehl **rlogin** nicht ausgeführt werden. Die Standards für gesicherte Befehle, Prozesse und Programme werden vom Department of Defense (Amerikanisches Verteidigungsministerium) festgelegt und verwaltet.

Für ein System können besondere Sicherheitsvorkehrungen erforderlich sein, um z. B. nicht berechnete Benutzer davon abzuhalten, auf Dateien zuzugreifen, kritische Daten zu entwenden, Dateien zu löschen oder Viren in das System zu bringen. Die Sicherheitsfunktionen des TCP/IP sind darauf ausgelegt, solche Vorkommnisse zu verhindern.

Ein Benutzer, der sich mit dem Befehl **telnet** an einem fernen Host anmelden möchte, muß über den Benutzernamen und das Kennwort eines für diesen Computer zugelassenen Benutzers verfügen. Diese Prozedur ist so ähnlich, als wenn sich der Benutzer am lokalen Host anmeldet. Verlieh die Anmeldung am fernen Host erfolgreich, ist die Datenstation des Benutzers so zu bedienen, als wäre sie direkt mit dem Host verbunden.

Der Befehl **telnet** implementiert das TELNET-Protokoll, mit dem eine Fernanmeldung an anderen Hosts möglich ist. Er verwendet das TCP/IP, um mit anderen Hosts im Netz zu kommunizieren. Der Befehl **tn** führt dieselbe Funktion aus wie der Befehl **telnet**.

Der Befehl **telnet** unterstützt die Option "Datenstationsvereinbarung". Unterstützt der ferne Host die Datenstationsvereinbarung, wird durch den Befehl **telnet** der lokale Datenstationstyp zum fernen Host gesendet. Falls der ferne Host den lokalen Datenstationstyp nicht akzeptiert, versucht der Befehl **telnet**, eine 3270-Datenstation oder eine Datenstation DEC VT100 zu emulieren. Gibt der Benutzer ein Terminal an, das emuliert werden soll, legt der Befehl **telnet** keinen Datenstationstyp fest. Gibt es keinen gemeinsamen Datenstationstyp zwischen lokalem und fernen Host, wechselt der Standardwert des lokalen Hosts auf **none**.

Der Befehl **telnet** unterstützt die folgenden Typen von 3270-Datenstationen: 3277-1, 3278-1, 3278-2, 3278-3, 3278-4 und 3278-5. Wird der Befehl **telnet** im 3270-Modus mit einem Farbbildschirm verwendet, werden standardmäßig die Farben und Felder wie bei einem Farbbildschirm IBM 3279 angezeigt. Es können andere Farben ausgewählt werden, indem eine der Tastaturbelegungsdateien für einen oben aufgelisteten Datenstationstyp editiert wird. Nach Beendigung der **telnet**-Sitzung werden die Farben so zurückgesetzt, wie sie vor der Sitzung waren.

Einführung in den Befehl rlogin

Der Befehl **rlogin** ist insofern dem Befehl **telnet** ähnlich, als daß beide einen lokalen Host mit einem fernen verbinden. Der einzige Unterschied ist, daß es sich bei dem Befehl **rlogin** um einen nicht gesicherten Befehl handelt, der, falls für das System besondere Sicherheitsvorkehrungen erforderlich sind, inaktiviert werden kann.

Der Befehl **rlogin** ist ein nicht gesicherter Befehl, da in der Datei **\$HOME/.rhosts**, deren Eigner der lokale Benutzer ist, und in der Datei **/etc/hosts.equiv**, deren Eigner der

Systemmanager ist (beide Dateien sind auf jedem lokalen Host vorhanden), Listen geführt werden von fernen Hosts, die auf den lokalen Host Zugriff haben. Wenn der Benutzer seine Datenstation verläßt, ohne sich abzumelden, wäre es daher möglich, daß eine unberechtigte Person die Namen und Kennwörter in diesen Listen untersucht oder schlimmstenfalls den Host in irgendeiner Weise beschädigt. Idealerweise müßten Benutzer auf einem fernen System nach der Eingabe des Befehls **rlogin** aufgefordert werden, ein Kennwort einzugeben. Diese empfehlenswerte Funktion kann aber umgangen werden.

Falls weder in der Datei **\$HOME/.rhosts** noch in der Datei **/etc/hosts.equiv** der Name eines fernen Hosts enthalten ist, der eine Anmeldung versucht, fordert der lokale Host zur Eingabe eines Kennwortes auf. In der fernen Kennwortdatei wird das Kennwort überprüft; der Anmeldedialog wird erneut angezeigt, wenn das Kennwort nicht korrekt war. Durch Drücken der Tasten ~. (Tilde, Punkt) im Anmeldedialog wird der Versuch der Fernanmeldung beendet.

Die Notwendigkeit, den Befehl **rlogin** einzugeben, kann auch aufgehoben werden. Der Befehl **rsh** ermöglicht die Ausführung von Befehlen auf einem fernen System; mit ihm kann aber die Kennwortanforderung nicht umgangen werden. Wenn ein Kennwort erforderlich ist, um auf den fernen Host zuzugreifen, ist auch zur Verwendung des Befehls **rsh** ein Kennwort erforderlich, da beide Befehle auf die Dateien **\$HOME/.rhosts** und **/etc/hosts.equiv** zugreifen.

Einführung in die Befehlsausführung auf einem fernen System

Die Befehle **rsh** und **rexec** können verwendet werden, um Befehle auf einem fernen Host auszuführen. Bei beiden Befehlen handelt es sich um nicht gesicherte Befehle, die möglicherweise nicht alle Sicherheitsstufen, die für den Computer konfiguriert sind, erfüllen. Deshalb können diese Befehle, falls besondere Sicherheitsvorkehrungen für das System erforderlich sind, inaktiviert werden.

Der Befehl **rsh**

Der Befehl **rsh** kann auf zwei Arten verwendet werden:

- Um einen einzelnen Befehl auf einem fernen Host auszuführen, wenn ein Befehlsname angegeben wurde.
- Um den Befehl **rlogin** auszuführen, wenn kein Befehlsname angegeben wurde.

Wurde der Befehl **rsh** eingegeben, sucht der lokale Host in der Datei **/etc/hosts.equiv** auf dem fernen Host nach der Anmeldeberechtigung. Ist diese Suche nicht erfolgreich, wird die Datei **\$HOME/.rhosts** durchsucht. Beide Dateien sind Listen von fernen Hosts, die Anmeldeberechtigung haben. Idealerweise müßten Benutzer auf einem fernen System nach der Eingabe des Befehls **rsh** aufgefordert werden, ein Kennwort einzugeben. Diese empfehlenswerte Funktion kann aber umgangen werden. Deshalb handelt es sich bei dem Befehl **rsh** um einen nicht gesicherten Befehl.

Die Befehle **rsh** und **rlogin** greifen auf dieselben Dateien zu. Deshalb erfordern beide Befehle dasselbe Kennwort zur vollständigen Fernanmeldung vom lokalen Host aus.

Der Befehl **rexec**

Bei der Eingabe des Befehls **rexec** durchsucht der lokale Host die Datei **\$HOME/.netrc** des fernen Hosts nach dem Benutzernamen und dem Kennwort. Werden sie gefunden, wird der Befehl, dessen Ausführung auf dem lokalen Host angefordert wurde, ausgeführt. Ansonsten sind zunächst ein Anmeldename und ein Kennwort erforderlich, bevor die Anforderung ausgeführt wird.

Empfohlene Dokumentation

Grundlegende Informationen

"Überblick über Netze" auf Seite 6-2

Zugehörige Informationen

Das Kapitel "TCP/IP Overview for System Management" im Handbuch *Communication Concepts and Procedures*.

Informationen über angemeldete Benutzer anzeigen

Der Befehl **finger** zeigt Informationen über Benutzer an, die momentan an einem Host angemeldet sind. Dieser Befehl kann verwendet werden, um:

- Informationen über alle Benutzer anzuzeigen, die an einem fernen Host angemeldet sind
- Eine kurze oder ausführliche Beschreibung der Aktivitäten eines einzelnen Benutzers an einem fernen Host anzuzeigen.

Voraussetzungen

Keine.

Informationen über alle an einem Host angemeldeten Benutzer anzeigen

Um Informationen über alle Benutzer anzuzeigen, die am Host `alcatraz` angemeldet sind, folgendes eingeben:

```
finger @alcatraz
```

Informationen, ähnlich den folgenden, werden angezeigt:

```
brown      console  Mar 15 13:19
smith     pts0    Mar 15 13:01
jones     tty0    Mar 15 13:01
```

Der Benutzer `brown` ist an der Konsole angemeldet, der Benutzer `smith` über eine Pseudofernschreiberleitung und Benutzer `jones` über eine nichtgrafikfähige Datenstation.

Informationen über einen einzelnen, an einem Host angemeldeten Benutzer anzeigen

Um Informationen über den Benutzer `brown` am Host `alcatraz` anzuzeigen, folgendes eingeben:

```
finger brown@alcatraz
```

Informationen, ähnlich den folgenden, werden angezeigt:

```
Anmeldename:  brown
Verzeichnis:  /home/brown      Shell:  /home/bin/xinit -L -n Startup
Am System seit May 8 07:13:49
Kein Plan.
```

Empfohlene Dokumentation

Zugehörige Informationen

Der Abschnitt zum Befehl **finger**.

Der Dämon **fingerd**.

Mit dem Befehl ftp Dateien kopieren

Mit dem Befehl **ftp** können Dateien von einem fernen Host kopiert werden. Dabei werden die Dateiattribute nicht beibehalten und die Unterverzeichnisse nicht kopiert. Ist eine dieser Bedingungen erforderlich, den Befehl **rcp** verwenden.

Voraussetzungen

1. Die Berechtigung zur Fernanmeldung muß in der Datei **\$HOME/.netrc** des fernen Hosts angegeben sein, falls die automatische Anmeldefunktion verwendet werden soll. Ansonsten muß ein Anmeldename und das entsprechende Kennwort für den fernen Host bekannt sein.
2. Soll eine Datei von einem fernen Host kopiert werden, muß eine Leseberechtigung für diese Datei vorliegen.

Hinweis: Lese- und Schreibberechtigungen für Dateien und Verzeichnisse auf einem fernen Host werden über den verwendeten Anmeldnamen festgelegt.

3. Soll eine Datei vom lokalen Host auf einen fernen Host kopiert werden, muß eine Schreibberechtigung für das Verzeichnis, in das die Dateien kopiert werden sollen, vorliegen. Ebenso muß, falls das Verzeichnis auf dem fernen Host eine Datei mit demselben Namen wie die zu kopierende Datei enthält, Schreibberechtigung vorliegen, um die Datei auf dem fernen Host anfügen zu können.

An einem fernen Host anmelden

1. Zum Verzeichnis wechseln, das die Datei enthält, die gesendet werden soll, oder zum Verzeichnis, in dem die übertragene Datei stehen soll.
2. Entweder direkt oder indirekt am fernen Host anmelden. Um sich direkt am fernen Host anzumelden, folgendes eingeben:

```
ftp Hostname
```

Liegt eine Berechtigung zum automatischen Anmelden vor, erscheint am lokalen Host eine Nachricht, ähnlich der folgenden:

```
Verbunden mit canopus.austin.century.com.  
220 canopus.austin.century.com FTP server (Version 4.1 Sat Nov 23  
12:52:09 CST 1991) ready.  
331 Password required for dee.  
230 User dee logged in.  
ftp>
```

Liegt die Berechtigung nicht vor, ähnelt die Nachricht des lokalen Hosts der folgenden. Den Anmeldnamen und das Kennwort eingeben, wenn das System dazu auffordert.

```
Verbunden mit canopus.austin.century.com.  
220 canopus.austin.century.com FTP server (Version 4.1 Sat Nov 23  
12:52:09 CST 1991) ready.  
Name (canopus:eric): dee  
331 Password required for dee.  
Password:  
230 User dee logged in.  
ftp>
```

3. Um sich indirekt am fernen Host anzumelden, folgendes eingeben:

```
ftp
```

Wenn die Eingabeaufforderung `ftp>` erscheint, folgendes eingeben:

open *Hostname*

Liegt eine Berechtigung zum automatischen Anmelden vor, erscheint am lokalen Host eine Nachricht, ähnlich der folgenden:

```
Verbunden mit canopus.austin.century.com.  
220 canopus.austin.century.com FTP server (Version 4.1 Sat Nov 23  
12:52:09 CST 1991) ready.  
331 Password required for dee.  
230 User dee logged in.  
ftp>
```

Liegt die Berechtigung nicht vor, ähnelt die Nachricht des lokalen Hosts der folgenden. Den Namen und das Kennwort eingeben, wenn das System dazu auffordert.

```
Verbunden mit canopus.austin.century.com.  
220 canopus.austin.century.com FTP server (Version 4.1 Sat Nov 23  
12:52:09 CST 1991) ready.  
Name (canopus:eric): dee  
331 Password required for dee.  
Password:  
230 User dee logged in.  
ftp>
```

Jetzt können Dateien zwischen zwei Hosts kopiert werden.

Eine Datei von einem fernen zu einem lokalen Host kopieren

Der Befehl **ftp** verwendet zum Kopieren von Dateien die ASCII-Standardübertragungsart.

1. Feststellen, ob sich die gewünschte Datei im aktuellen Verzeichnis befindet, indem der Unterbefehl **dir** eingegeben wird. (Der Unterbefehl **dir** für den Befehl **ftp** erfüllt dieselbe Funktion wie der Befehl **ls -l**.) Befindet sich die Datei dort nicht, mit dem Unterbefehl **cd** in das passende Verzeichnis wechseln.
2. Soll das binäre Abbild der lokalen Datei kopiert werden, folgendes eingeben:

```
binary
```

3. Um die Datei zum lokalen Host zu kopieren, folgendes eingeben:

```
get Dateiname
```

Die Datei wird in das Verzeichnis gestellt, von dem aus der Befehl **ftp** eingegeben wurde.

4. Um die Sitzung zu beenden, entweder die Tastenfolge Strg-D drücken oder **quit** eingeben.

Eine Datei von einem lokalen zu einem fernen Host kopieren

Der Befehl **ftp** verwendet zum Kopieren von Dateien die ASCII-Standardübertragungsart.

1. Soll die Datei nicht in das Benutzerverzeichnis **\$HOME** gestellt werden, den Unterbefehl **cd** verwenden, um in das gewünschte Verzeichnis zu wechseln.
2. Soll das binäre Abbild der lokalen Datei kopiert werden, folgendes eingeben:

```
binary
```

3. Um die Datei zum fernen Host zu kopieren, folgendes eingeben:

```
put Dateiname
```

Die Datei wird in das Arbeitsverzeichnis des fernen Hosts gestellt.

4. Um die Sitzung zu beenden, entweder die Tastenfolge Strg-D drücken oder **quit** eingeben.

Empfohlene Dokumentation

Zugehörige Informationen

Die Abschnitte zu den Befehlen **cd**, **ls**, **ftp** und **rcp**.

Der Dämon **ftpd**.

Von einem fernen System aus drucken

Gelegentlich kann es notwendig sein, eine Datei auf einem fernen Host zu drucken. Der Standort der Druckausgabe ist dabei abhängig davon, welche fernen Drucker dem fernen Host zur Verfügung stehen.

Hinweis: In diesem Abschnitt wird die einfachste Art der Druckausgabe von einem fernen Host aus erläutert. Weitere Informationen über das ferne Drucken befinden sich im Abschnitt zum Befehl **enq**.

Voraussetzungen

1. Das Anmelden an einem fernen System mit dem Befehl **rlogin** oder **telnet** muß möglich sein.
2. Die Leseberechtigung für die ferne Datei, die auf einem lokalen Drucker ausgedruckt werden soll, muß vorliegen.

Erforderliche Schritte

1. Am fernen System mit dem Befehl **rlogin** oder **telnet** anmelden.
2. Den passenden Warteschlangennamen und den Namen der fernen Einheit suchen. Der Warteschlangename beginnt gewöhnlich mit den Buchstaben **rp**, gefolgt von einem oder mehreren Numeralen. Der Name des fernen Druckers beginnt gewöhnlich mit den Buchstaben **drp**, gefolgt von einem oder mehreren Numeralen.
3. Den folgenden Befehl eingeben:

```
enq -P Warteschlangenname:Einheitenname Dateiname
```

Warteschlangenname ist der Name der Warteschlange (z. B. **rp1**) und *Einheitenname* ist der Name des Druckers (z. B. **drp1**), wie in der Datei **/usr/lib/lpd/qconfig** aufgeführt. Nicht den Doppelpunkt : zwischen *Warteschlangenname* und *Einheitenname* auslassen. *Dateiname* ist der Name der Datei, die ausgedruckt werden soll.

4. Die Verbindung zum fernen Host durch Drücken der Tastenfolge Strg-D beenden.

Empfohlene Dokumentation

Zugehörige Informationen

"An einem fernen System drucken" auf Seite 6-20.

Die Abschnitte zu den Befehlen **enq**, **rlogin** und **telnet**.

An einem fernen System drucken

Ist ein lokaler Drucker an den Host angeschlossen, bezieht sich dieser Abschnitt auf das Drucken an einem fernen Drucker. Ist kein lokaler Drucker angeschlossen, bezieht sich dieser Abschnitt auf das Drucken an einem nicht standardmäßigen fernen Drucker. Zum Ausführen dieser Aufgabe kann entweder der Befehl `enq` verwendet werden oder das System Management Interface Tool (SMIT).

Hinweis: In diesem Abschnitt wird die einfachste Art der Druckausgabe an einem fernen Host erläutert. Weitere Informationen über das ferne Drucken befinden sich im Abschnitt zum Befehl `enq`.

Voraussetzungen

1. Der Name des lokalen Hosts muß in der Datei `/etc/hosts.lpd` des fernen Hosts festgelegt sein.

Hinweis: Das Warteschlangensystem unterstützt keine Mehrbyte-Host-Namen.

2. Es muß möglich sein, den Warteschlangennamen und den Namen des fernen Druckers in der lokalen Datei `/usr/lib/lpd/qconfig` festzustellen.

Verwendung des Befehls `enq`

1. Den passenden Warteschlangennamen und den Namen der fernen Einheit suchen. Der Warteschlangename beginnt gewöhnlich mit den Buchstaben `rp`, gefolgt von einem oder mehreren Numeralen. Der Name des fernen Druckers beginnt gewöhnlich mit den Buchstaben `drp`, gefolgt von einem oder mehreren Numeralen.
2. Den folgenden Befehl eingeben:

```
enq -P Warteschlangename:Einheitename Dateiname
```

Warteschlangename ist der Name der Warteschlange (z. B. `rp1`) und *Einheitename* ist der Name des Druckers (z. B. `drp1`), wie in der Datei `/usr/lib/lpd/qconfig` aufgeführt. Nicht den Doppelpunkt : zwischen *Warteschlangename* und *Einheitename* auslassen. *Dateiname* ist der Name der Datei, die ausgedruckt werden soll.

Warteschlangename und *Einheitename* auslassen. *Dateiname* ist der Name der Datei, die ausgedruckt werden soll.

Verwendung des System Management Interface Tool (SMIT)

1. Den folgenden Befehl eingeben:

```
smit
```
2. Die Menüs `Druckjobs (Spooler)` und `Druckjob starten auswählen`.
3. Die Option zu druckende DATEI auswählen und den Namen der Datei eingeben, die gedruckt werden soll.
4. Die Option `Druckwarteschlange auswählen` und den Namen des fernen Druckers auswählen, an dem gedruckt werden soll. Jetzt kann an einem fernen Drucker gedruckt werden.

Empfohlene Dokumentation

Zugehörige Informationen

"Von einem fernen System aus drucken" auf Seite 6-19.

Der Abschnitt zum Befehl `enq`.

Dialog mit einem Benutzer auf einem fernen System

Mit dem Befehl **talk** ist es möglich, einen echten Dialog mit einem Benutzer auf einem fernen Host zu führen.

Voraussetzungen

1. Der Dämon **talkd** muß sowohl auf dem lokalen als auch auf dem fernen Host aktiv sein.
2. Der Benutzer auf dem fernen Host muß angemeldet sein.

Den Dialog mit dem Benutzer auf einem fernen System starten

Um mit dem Benutzer auf einem fernen Host zu kommunizieren, folgenden Befehl eingeben:

```
talk dale@host2
```

Eine Dialoganfrage beantworten

Möchte ein Benutzer auf einem fernen System einen Dialog starten, erscheint beim Empfänger eine Nachricht, ähnlich der folgenden:

```
Message from TalkDaemon@host1 at 15:16...
talk: connection requested by jane@host1.
talk: respond with: talk jane@host1
```

Um auf die Anfrage zu antworten, folgendes eingeben:

```
talk jane@host1
```

Die Benutzer sind jetzt in der Lage, einen interaktiven Dialog zu führen.

Eine Dialoganfrage beenden

Um den Dialog zu beenden, die Tastenfolge Strg-C drücken. Damit wird zur Eingabeaufforderung der Befehlszeile zurückgekehrt.

Empfohlene Dokumentation

Zugehörige Informationen

Der Abschnitt zum Befehl **talk**.

Der Dämon **talkd**.

Einen lokalen Host mit einem fernen Host verbinden

Die Befehle **telnet**, **tn** oder **tn3 270** ermöglichen, eine Verbindung zwischen einem lokalen und einem fernen Host unter Verwendung der TELNET-Schnittstelle herzustellen.

Voraussetzungen

Für den fernen Host muß eine gültige Benutzer-ID und ein Kennwort bestehen.

Erforderliche Schritte

1. Um sich am fernen Host anzumelden (in diesem Beispiel `host1`) und die Datenstation festzulegen, folgendes eingeben:

```
telnet host1
```

Informationen, ähnlich den folgenden, werden angezeigt:

```
Versuch . . .  
Verbunden mit host1  
Escape-Zeichen ist '^T'.
```

```
AIX telnet (host1)
```

```
IBM AIX Operating System  
Version 3.2  
(/dev/pts0)  
login:_
```

2. Ist ein Anmelden nicht möglich, kann die Verbindung durch Drücken der Tastenfolge Strg-T abgebrochen werden.

War die Anmeldung erfolgreich, können Befehle eingegeben werden. Um sich abzumelden und die Verbindung zu beenden, die Tastenfolge Strg-D drücken.

Empfohlene Dokumentation

Grundlegende Informationen

"Einführung in den Befehl **telnet**" auf Seite 6-12.

Zugehörige Informationen

Der Abschnitt zum Befehl **telnet**.

Der Dämon **telnetd**.

Mit dem Befehl tftp Dateien kopieren

Hinweis: Der Befehl `tftp` steht nicht zur Verfügung, wenn der Host auf einer hohen Sicherheitsstufe betrieben wird.

Voraussetzungen

1. Soll eine Datei von einem fernen Host kopiert werden, muß eine Leseberechtigung für das Verzeichnis, das die gewünschte Datei enthält, vorliegen.
2. Soll eine Datei auf einen fernen Host kopiert werden, muß eine Schreibberechtigung für das Verzeichnis, in das die Datei kopiert werden soll, vorliegen.

Eine Datei von einem fernen Host kopieren

1. Um die Verbindung zu einem fernen Host herzustellen, folgendes eingeben:

```
tftp host1
```

Bei diesem Beispiel ist `host1` der Name des Hosts, zu dem die Verbindung hergestellt werden soll.

2. Die Eingabeaufforderung `tftp>` wird angezeigt. Um festzustellen, ob eine Verbindung hergestellt wurde, folgendes eingeben:

```
status
```

Eine Nachricht, ähnlich wie die folgende, wird angezeigt:

```
Verbunden mit host1
Modus: netascii  Verbose: off  Trace: off
Remxt-interval: 5 Sekunden, Max-timeout: 25 Sekunden
tftp>
```

3. Nach der Eingabeaufforderung `tftp>` den Unterbefehl `get` eingeben, den Namen der Datei, die übertragen werden soll, und den Namen, den die Datei auf dem Zielsystem erhalten soll.

```
get /home/alice/update update
```

Das Verzeichnis `/home/alice` auf dem fernen Host muß anderen Hosts Lesezugriff gewährt haben. In diesem Beispiel wird die Datei `/home/alice/update` von `host1` zur Datei `update` im aktuellen Verzeichnis auf dem lokalen System übertragen.

4. Um die Sitzung zu beenden, folgendes eingeben:

```
quit
```

bzw.

die Tastenfolge Strg-D drücken.

Eine Datei zu einem fernen Host kopieren

1. Um die Verbindung zu einem fernen Host herzustellen, folgendes eingeben:

```
tftp host1
```

Bei diesem Beispiel ist `host1` der Name des Host, zu dem die Verbindung hergestellt werden soll.

2. Die Eingabeaufforderung `tftp>` wird angezeigt. Um festzustellen, ob eine Verbindung hergestellt wurde, folgendes eingeben:

```
status
```

Eine Nachricht, ähnlich wie die folgende, wird angezeigt:

```
Verbunden mit host1
Modus: netascii  Verbose: off  Trace: off
Remxt-interval: 5 Sekunden, Max-timeout: 25 Sekunden
tftp>
```

3. Nach der Eingabeaufforderung `tftp>` den Unterbefehl **put** eingeben, den Namen der Datei, die vom lokalen Host übertragen werden soll, und den Pfad- und Dateinamen für die Datei auf dem fernen Host:

```
put meinDat /home/alice/deinDat
```

Das Verzeichnis `/home/alice` auf dem fernen Host muß anderen Hosts Schreibzugriff gewähren. Die Datei `meinDat` aus dem aktuellen Arbeitsverzeichnis des Benutzers wird an `host1` übertragen. Der Pfadname muß angegeben werden, es sei denn, ein Standardwert wurde festgelegt. Die Datei `meinDat` erscheint auf dem fernen Host als `deinDat`.

4. Um die Sitzung zu beenden, folgendes eingeben:

```
quit
```

bzw.

die Tastenfolge Strg-D drücken.

Empfohlene Dokumentation

Zugehörige Informationen

Der Abschnitt zum Befehl **tftp**.

Der Dämon **tftpd**.

Kapitel 7. Post

Die Postfunktion ermöglicht das Senden, Empfangen und Speichern von Nachrichten.

Im ersten Teil dieses Kapitels wird die Postfunktion mit ihren Merkmalen erläutert, wobei folgende Bereiche behandelt werden:

- Post erstellen und senden
- Post adressieren
- Post senden
- Der Editor für elektronische Post
- Post empfangen und bearbeiten
- Der elektronische Briefkasten und der Postordner

Der zweite Teil dieses Kapitels enthält Prozeduren zur Verwendung der Postfunktion, wobei folgende Bereiche behandelt werden:

- Das Postprogramm starten
- Post empfangen und bearbeiten
- Post erstellen und senden
- Geheime Post senden und empfangen
- Das Postprogramm einstellen

Das Handbuch *Communication Concepts and Procedures* enthält Informationen, wie die Postfunktion verwaltet wird.

Überblick über die Postfunktion

Eine Nachricht der elektronischen Post kann aus Text bestehen, der mit einem Editor eingegeben wurde, oder aus einer ASCII-Datei. Zusätzlich zur eingegebenen Nachricht oder zur Datei können folgende Nachrichten gesendet werden:

Systemnachricht

Informiert den Benutzer, daß das System aktualisiert wurde. Eine Systemnachricht ähnelt einer Rundsendenachricht, wird aber nur an das lokale Netz gesendet.

Geheime Post Wird verwendet, um klassifizierte Informationen zu senden. Eine geheime Nachricht der elektronischen Post ist verschlüsselt. Der Empfänger muß ein Kennwort eingeben, um sie lesen zu können.

Urlaubsnachricht

Informiert den Benutzer, daß der Empfänger abwesend ist. Wenn das System während der Abwesenheit Post empfängt, wird eine Nachricht an den Absender gesendet, die besagt, daß der Empfänger abwesend ist. Während der Abwesenheit des Benutzers kann die Post auch weitergeleitet werden.

Empfängt der Benutzer Post mit dem Unterbefehl **mail**, stehen folgende Möglichkeiten zur Verfügung:

- Die Post im elektronischen Systembriefkasten lassen.
- Die Post lesen und löschen.
- Die Post weiterleiten.
- Der Post einen Kommentar hinzufügen.
- Die Post im persönlichen elektronischen Briefkasten (**mbox**) speichern.
- Die Post in einem selbsterstellten Ordner speichern.

Jedes System hat eine eindeutige Postadresse und einen eindeutigen Alias-Namen. Der Benutzer kann eine Datei für Alias-Namen oder eine Verteilerdatei erstellen und verwalten, die seine Post und Nachrichten der elektronischen Post steuert.

Hinweis: Es gibt einige Programme zum Senden und Empfangen von Post, u. a. Message Handler (MH), Bellmail, und das Programm **xtalk**. Welches Programm verwendet wird, hängt davon ab, welches auf dem System installiert und wie es konfiguriert ist.

Empfohlene Dokumentation

Grundlegende Informationen

"Dateien" auf Seite 3-3.

Zugehörige Informationen

Das Kapitel "Mail Overview for System Management" im Handbuch *Communication Concepts and Procedures*.

Das Kapitel "Message Handler (MH) Overview" im Handbuch *Communication Concepts and Procedures*.

Das Kapitel "Using the **xtalk** Command" im Handbuch *Communication Concepts and Procedures*.

Mit den Befehlen **mail** und **bellmail** wird das Postprogramm gestartet.

Post erstellen und senden

Mit dem Postprogramm kann eine Nachricht erstellt und an einen anderen Benutzer gesandt werden. Außerdem kann mit dem Postprogramm eine ASCII-Datei an einen anderen Benutzer gesendet werden. Die ASCII-Datei könnte beispielsweise ein Dokument sein, das mit einem bevorzugten Editor geschrieben wurde, oder eine Quellendatei für ein Programm.

Das Postprogramm verfügt über einen zeilenorientierten Editor zum Erstellen von Nachrichten. Mit diesem Editor kann der Benutzer eine einzelne Zeile seiner Nachricht eingeben, die Eingabetaste drücken und in eine neue Zeile weiteren Text eingeben. Wird Post mit dem Editor für elektronische Post erstellt, werden die Felder `date:` und `from:` automatisch vom System ausgefüllt. Der Benutzer hat die Möglichkeit, die Felder `subject:` und `cc:` auszufüllen. Diese Felder ähneln der Einleitung eines Standardgeschäftsbriefes.

Nachrichten und Dateien können an Benutzer im lokalen System, im eigenen Netz oder an Benutzer eines anderen, angeschlossenen Netzes gesendet werden. Der Empfänger muß nicht am System angemeldet sein, wenn die Informationen abgesendet werden. Post wird an die Adresse eines Benutzers gesendet. Über die Adresse, bestehend aus Benutzer-ID und Systemname, wird die Zustellung der Nachricht der elektronischen Post gesteuert.

Hinweis: Das Postprogramm versucht drei Tage lang, die Verbindung zu einem System, das heruntergefahren wurde, aufzunehmen. Danach wird eine Fehlernachricht an den Sender zurückgegeben und die Post muß erneut gesendet werden.

Post adressieren

Post wird an die Adresse eines Benutzers gesendet. Über die Adresse, bestehend aus Benutzer-ID und Systemname, wird die Zustellung der Nachricht der elektronischen Post gesteuert. Generell müssen zum Senden einer Nachricht an einen anderen Benutzer der Befehl `mail` und die Adresse wie folgt eingegeben werden:

`mail Benutzer@Adresse`

Das Format des Parameters *Adresse* ist jedoch abhängig vom Standort des Empfängers. Das Konzept ist so ähnlich, als wenn man einem Kollegen im Büro einen Brief zuschicken möchte: Arbeitet dieser Kollege (Ryan) in derselben kleinen Abteilung mit 6 bis 8 Leuten, könnte der Absender einfach den Namen auf eine Briefhülle schreiben und den Brief über das interne Postsystem abschicken. Falls der Kollege jedoch in einer anderen Abteilung arbeitet, sollte die Briefhülle mit weiteren Informationen versehen werden:

Ryan
Lohnbuchhaltung

Arbeitet der Kollege in einem anderen Werk, sind sogar noch weitere Informationen notwendig, damit der Brief ihn tatsächlich erreicht:

Ryan
Lohnbuchhaltung
Gaithersburg

Um Post elektronisch zu senden, ist eine ähnliche Art der Adressierung notwendig. Wird die Post an einen Benutzer im lokalen System gesendet, ist die Angabe der Adresse nicht erforderlich. Um beispielsweise an Jan im lokalen System Post zu senden, folgenden Befehl eingeben:

`mail jan`

Um an einen Benutzer im selben Netz Post zu senden, muß die volle Adresse des Systems bzw. Systemknotens eingegeben werden. Um beispielsweise an Kaye am Knoten Tybalt Post zu senden, folgenden Befehl eingeben:

```
mail kaye@tybalt
```

Um an einen Benutzer eines anderen verbundenen Netzes Post zu senden, muß die volle Systemadresse und die Netzadresse eingegeben werden. Um beispielsweise Post an Nick am Knoten Mars in einem anderen Netz zu senden, folgenden Befehl eingeben:

```
mail nick@mars.aus.dbm.com
```

Post an Benutzer des lokalen Systems senden

Um an einen Benutzer im lokalen System (an jemanden, dessen Anmelde-ID in der Datei `/etc/passwd` des Benutzers aufgelistet ist) Post zu senden, die Anmelde-ID als Adresse verwenden:

```
mail Anmelde-ID
```

Ist Ryan beispielsweise ein Benutzer des lokalen Systems, folgenden Befehl eingeben, um eine Nachricht zu erstellen und an ihn zu senden:

```
mail ryan
```

Mit diesem Befehl wird das Postprogramm aktiviert, kann eine Nachricht erstellt werden und wird die Nachricht an die lokale Benutzer-ID `ryan` gesendet. Wird die Nachricht erfolgreich zugestellt, erhält der Absender keine Hinweismeldung. Ist Ryan nicht Benutzer des lokalen Systems, gibt das Postprogramm eine Fehlermeldung zurück und stellt die nicht gesendete Nachricht in den elektronischen Systembriefkasten des Absenders zurück.

Post an Benutzer des Netzes senden

Um eine Nachricht über das lokale Netz an einen Benutzer eines anderen Netzes senden zu können, müssen die Anmelde-ID und der Name des anderen Systems bekannt sein. Folgendes Befehlsformat verwenden, um eine Nachricht zu senden:

```
mail Benutzer-ID@Systemname
```

Ist Ryan beispielsweise Benutzer des Systems Zeus, folgenden Befehl verwenden, um eine Nachricht zu erstellen und an ihn zu senden:

```
mail ryan@zeus
```

Mit diesem Befehl wird das Postprogramm aktiviert, kann eine Nachricht erstellt werden und wird die Nachricht an die Benutzer-ID `ryan` auf dem System `zeus` gesendet. Wird die Nachricht erfolgreich zugestellt, erhält der Absender keine Hinweismeldung. Ist die Postadresse fehlerhaft, erhält der Absender zwar keine Fehlermeldung, aber das Postprogramm stellt die nicht zugestellte Nachricht in den elektronischen Systembriefkasten des Benutzers mit einer Erklärung, warum sie nicht zugestellt werden konnte.

Den Namen eines Systems feststellen

Für das Postprogramm ist der Name des Systems über eine Konfigurationsdatei auf diesem System festgelegt. Üblicherweise ist der Name identisch mit dem Knotennamen des betreffenden Systems, er kann jedoch in der Konfigurationsdatei auch anders definiert sein.

Um den Namen des eigenen Systems zu ermitteln, den Befehl `uname -a` eingeben. Die Antwort ähnelt der folgenden:

aix procyon 2 3 0000520835

Procyon ist in diesem Beispiel der Knotenname.

Um den Namen eines anderen Systems zu ermitteln, ist es erforderlich, daß ein Benutzer auf diesem System den Befehl **uname -a** eingibt.

Zugriffsmöglichkeiten des Systems feststellen

Das lokale System des Benutzers muß Zugriff auf Informationen haben, wie andere Systeme im Netz definiert sind. Um festzustellen, ob das lokale System über diese Informationen verfügt, den Befehl **host** verwenden. Um beispielsweise festzustellen, ob das lokale System über Leitweginformationen über das System **zeus** verfügt, folgenden Befehl eingeben:

```
host zeus
```

Sieht die Systemantwort ähnlich aus wie die folgende:

```
zeus ist 192.9.200.4 (300,11,310,4)
```

verfügt das System über die nötigen Informationen und der Benutzer kann eine Nachricht an das System **zeus** senden. Die Adresse 192.9.200.4 wird vom System verwendet, um die Post weiterzuleiten.

Verfügt das System nicht über Informationen über das angeforderte System, erscheint folgende Nachricht als Antwort:

```
zeus: unbekannter Host
```

Wird eine Nachricht über einen unbekannt Host ausgegeben, könnte das folgende Ursachen haben:

- Der Systemname wurde fehlerhaft eingegeben (Rechtschreibung überprüfen).
- Der Systemname existiert im Netz, ist aber für das lokale System nicht definiert (den Verantwortlichen für die Einstellungen im Netz kontaktieren).
- Der Systemname existiert in einem anderen Netz und eine detailliertere Adressierung ist notwendig, um ihn zu definieren.
- Das angeforderte System ist nicht mit dem Netz verbunden.

Die Nachricht über einen unbekannt Host wird auch ausgegeben, wenn das Netz nicht in Betrieb ist und das lokale System bei der Netzadressenermittlung von einem fernen System abhängig ist.

Post an Benutzer in einem anderen Netz senden

Ist das Netz des Benutzers mit anderen Netzen verbunden, kann auch an die Benutzer der anderen Netze Post gesendet werden. Die Adreßparameter hängen davon ab, welche Adressen die Netze untereinander verwenden und wie sie miteinander verbunden sind.

Post senden unter Verwendung einer zentralen Namens- und Adressdatenbank

Ist im Netz eine zentrale Namensdatenbank vorhanden, sind keine zusätzlichen Informationen notwendig, um Post an Benutzer von verbundenen Netzen zu senden. Dasselbe Adreßformat wie für Benutzer des lokalen Netzes kann verwendet werden:

mail Benutzer-ID@Systemname

Diese Art der Adressierung funktioniert gut, wenn das Netz die Pflege einer zentralen Namensdatenbank ermöglicht.

Post senden über Domännennamen

Für weit ausgedehnte Netze oder nicht miteinander verbundene Netze an weit auseinanderliegenden Standorten ist eine zentrale Namensdatenbank nicht realisierbar. Um an einen Empfänger in einem solchen Netz Post zu senden, folgendes Befehlsformat verwenden:

`mail Benutzer-ID@Systemname.Domänenname`

Über den Parameter *Domänenname* wird, entsprechend dem lokalen Netz, das ferne Netz definiert, und zwar innerhalb einer definierten Struktur für eine größere Gruppe miteinander verbundener Netze.

Mit einem hinzugefügten Netznamen

Ein Domänenname kann einfach ein hinzugefügter Netzname sein. Ist beispielsweise ein lokales Netz mit dem Namen Olympus mit einem zweiten Netz mit dem Namen Valryanla verbunden, könnte der Benutzer Ryan folgenden Befehl eingeben, um eine Notiz an den Benutzer Kelly am System Odin im zweiten Netz zu senden:

`mail kelly@odin.valryanla`

Ebenso könnte Kelly dem Benutzer Ryan auf dem System Zeus mit dem folgenden Befehl antworten:

`mail ryan@zeus.olympus`

Mit dem Domänenpfad

Häufig ist der Domänenname nicht nur der Name eines anderen Netzes, sondern wird zum Pfad durch die logische Anordnung von Domänen im Netz. Dieser Pfad stellt allerdings nicht den tatsächlichen Leitweg dar, den die Nachricht nimmt, sondern nur die Position des Zielnetzes innerhalb der Struktur der miteinander verbundenen Netze.

Beispiel

Das größte und bekannteste Beispiel für diese Art der Verbindung ist ein Netz von Unternehmen, der amerikanischen Regierung und verschiedenen Bildungseinrichtungen, genannt ARPA-Netz. Auf höchster struktureller Ebene ist es in mehrere große Domänen aufgeteilt, unter anderem die folgenden:

- COM für Wirtschaftseinrichtungen
- EDU für Bildungseinrichtungen
- GOV für Regierungsbehörden
- ARPA für Miscellaneous Groups
- BITNET für die Verbindung zum BITNET-Netz
- CSNET für die Verbindung zum CSNET-Netz

Die folgende Abbildung zeigt einige Teile des ARPA-Netzes auf höchster Ebene, wobei einige gedachte Verzweigungen im Detail dargestellt sind, um das Konzept der Domännennamen zu verdeutlichen.

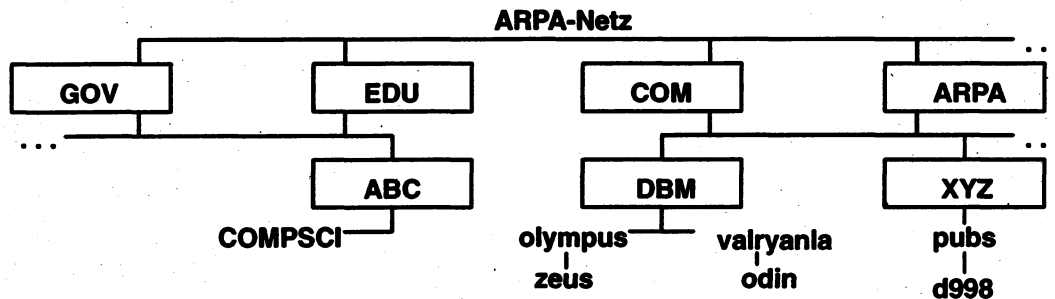


Abbildung 7. Allgemeine ARPA-Netz-Struktur mit Beispielverbindungen

In der Abbildung ist die Domäne `pubs` mit der größeren Domäne `XYZ` verbunden, aber nicht direkt mit `olympus`. Daher gibt der Benutzer Cath am System `d998` folgenden Befehl ein, um an den Benutzer Kelly am System `Odin` eine Notiz zu senden:

```
mail kelly@odin.valryanla.dbm
```

Ebenso könnte Kelly dem Benutzer Cath auf dem System `d998` mit dem folgenden Befehl antworten:

```
mail cath@d998.pubs.XYZ
```

Jede dieser Adressen gibt nur den Teil der Adresse an, der nötig ist, um die Zieladresse von der Domäne `COM` aus zu erreichen. Das Leitwegprogramm dieser Domäne erkennt die Domänen `DBM` und `XYZ`. Wenn allerdings von der Domäne `COMPSCI` aus eine Nachricht an den Benutzer `cath` gesendet werden soll, muß folgender Befehl eingegeben werden:

```
mail cath@d998.pubs.XYZ.COM
```

In diesem Befehl wird die volle Adresse des Benutzers `cath` in diesem Beispielnetz verwendet.

Wird häufig Post an Benutzer in anderen Netzen gesendet, kann durch die Erstellung von Alias-Namen, die die Adressen der Benutzer enthalten, Zeit gespart werden. Siehe dazu:

- "Das Postprogramm einstellen" auf Seite 7-18.
- "Alias-Namen oder Verteiler erstellen" auf Seite 7-50.
- "Definierte Alias-Namen oder Verteiler bestimmen" auf Seite 7-51.

Post senden über eine BNU- oder UUCP-Verbindung

Um eine Nachricht an einen Benutzer an einem anderen System zu senden, das über das Netz-Basisdienstprogramm BNU (Basic Networking Utilities) oder eine andere Version von UNIX-to-UNIX Copy Program (UUCP) mit dem eigenen System verbunden ist, muß folgendes bekannt sein:

- Die Anmelde-ID
- Der Name des anderen Systems
- Der physische Leitweg zu dem anderen System

Der Verantwortliche für die Verbindung von Systemen sollte die notwendigen Leitweginformationen zur Verfügung stellen können, um andere Systeme zu adressieren.

Adressierung mit einer UUCP-Verbindung auf dem eigenen System

Verfügt das lokale System über eine BNU- oder UUCP-Verbindung, die verwendet werden kann, um den fernen Standort zu erreichen, zum Senden einer Nachricht folgendes Format verwenden:

mail UUCP-Leitweg!Benutzer-ID

Die Variable *Benutzer-ID* ist die Anmelde-ID des Empfängers auf dem fernen System. Die Variable *UUCP-Leitweg* beschreibt den physischen Leitweg der Nachricht im UUCP-Netz. Falls das eigene System mit dem fernen System ohne ein zwischengeschaltetes UUCP-System verbunden ist, ist die Variable der Name des fernen Systems. Verfügt beispielsweise das lokale System über eine UUCP-Verbindung zu dem System Merlin und gibt es keine weiteren UUCP-Systeme zwischen dem lokalen System und Merlin, den folgenden Befehl verwenden, um an den Benutzer Ken auf diesem System eine Nachricht zu senden:

```
mail merlin!ken
```

Wenn die Nachricht erst durch eine oder mehrere zwischengeschaltete UUCP-Systeme geht, bevor sie das gewünschte ferne System erreicht, ist die Variable eine Liste der einzelnen zwischengeschalteten Systeme. Die Liste beginnt mit dem am nächsten gelegenen System und endet mit dem am weitesten gelegenen, die jeweils durch ein ! (Ausrufezeichen) voneinander getrennt sind. Wenn die Nachricht beispielsweise erst durch das System Arthur und dann durch Lancelot (in dieser Reihenfolge) geht, bevor sie das System Merlin erreicht, muß zum Senden folgender Befehl verwendet werden:

```
mail arthur!lancelot!merlin!ken
```

Adressierung mit einer UUCP-Verbindung auf einem anderen System

Angenommen, in einer Umgebung mit einem lokalen oder öffentlichen Netz verfügt eines der Systeme im Netz über eine BNU-Verbindung oder eine andere Art von UUCP-Verbindung zu einem fernen System, so kann über diese UUCP-Verbindung eine Nachricht an einen Benutzer an diesem fernen UUCP-System gesendet werden. Dabei folgendes Befehlsformat verwenden:

```
mail @SystemA:@SystemB.UUCP:Benutzer-ID@SystemC
```

Mit diesem Format wird die Post zunächst zu *SystemA* gesendet, dann zu *SystemB*, das sie über eine UUCP-Verbindung zu *SystemC* weiterleitet. Der Zusatz *.UUCP* zur Adresse von *SystemB* zeigt an, daß das UUCP-Postzustellprogramm an diesem System die Weiterleitung der Nachricht zu *SystemC* bearbeitet. Zu beachten ist, daß mit diesem Format keine Post an einen Benutzer eines der zwischengeschalteten Systeme gesendet wird, so daß vor der Angabe @ in der Domänenadresse keine Benutzer-ID eingegeben wird.

Beispiel einer UUCP-Verbindung im Netz

Die folgende Abbildung zeigt ein Beispielnetz, das für den Hauptanteil der Post die Adressierung über Domänen verwendet, aber über eine UUCP-Verbindung verfügt, die Post zu den Systemen *dept1* und *dept2* weiterleitet. Das System *dept2* ist mit einem anderen System verbunden, das System *dept3* mit einem lokalen Netz.

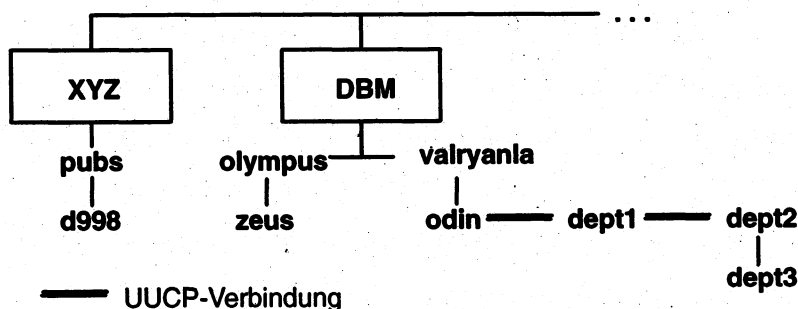


Abbildung 8. Beispiel einer UUCP-Verbindung im Netz

Beispiele für Befehlsformate

Die folgenden Befehlsformate zeigen Adressierungsmöglichkeiten für das Beispielnetz in der Abbildung "Beispiel einer UUCP-Verbindung im Netz".

Der Benutzer `kelly` am System `odin` muß folgende Befehle eingeben, um an Benutzer `fred` am System `dept1`, an `dick` am System `dept2` und an `bill` am System `dept3` Nachrichten zu senden:

```
mail @odin.UUCP:fred@dept1
mail @odin.UUCP:@dept1.UUCP:dick@dept2
mail @odin.UUCP:@dept1.UUCP:@dept2:bill@dept3
```

Mit dem ersten Befehl wird die Post zu `fred@dept1` gesendet und zwar über eine UUCP-Verbindung zwischen den Systemen `odin` und `dept1`. Mit dem zweiten Befehl wird die Post zu `dick@dept2` gesendet über die UUCP-Verbindung zum System `dept1` und dann über die zweite UUCP-Verbindung zum System `dept2`. Mit dem dritten Befehl wird die Post zu `bill@dept3` gesendet und zwar über die zwei UUCP-Verbindungen und dann über die lokale Netzverbindung zwischen den Systemen `dept2` und `dept3`.

Fred, Dick, und Bill antworten mit folgenden Befehlen:

```
mail @dept1.UUCP:kelly@odin
mail @dept2.UUCP:@dept1.UUCP:kelly@odin
mail @dept2.UUCP:@dept1.UUCP:kelly@odin
```

Mit dem dritten Befehl wird die Post von `bill@dept3` zu `kelly@odin` gesendet und zwar zunächst zum System `dept2`, dann über die UUCP-Verbindung zum System `dept1` und dann über die UUCP-Verbindung zum System `odin`.

Ebenso kann der Benutzer `cath` am System `d998` durch Eingabe der folgenden Befehle Post an dieselben Benutzer senden:

```
mail @odin.UUCP.valryanla.DBM:fred@dept1
mail @odin.UUCP.valryanla.DBM:@dept1.UUCP:dick@dept2
mail @odin.UUCP.valryanla.DBM:@dept1.UUCP:@dept2:bill@dept3
```

In diesem Beispiel wird mit dem ersten Befehl Post von `cath@d9 98.pubs.XYZ` zu `fred@dept1` gesendet, wobei die Post zunächst über das Domänennetz zum System `odin.valryanla.DBM` und dann über die UUCP-Verbindung zwischen den Systemen `odin` und `dept1` gesendet wird.

Fred, Dick, und Bill antworten mit folgenden Befehlen:

```
mail @dept1.UUCP:@odin:cath@d998.pubs.XYZ
mail @dept2.UUCP:@dept1.UUCP:@odin:cath@d998.pubs.XYZ
mail @dept2.UUCP:@dept1.UUCP:@odin:cath@d998.pubs.XYZ
```

Mit dem dritten Befehl wird die Post von `bill@dept3` zu `cath@d998.pubs.XYZ` gesendet, wobei sie zunächst den Weg über das lokale Netz zum System `dept2` nimmt, dann über die UUCP-Verbindungen zu den Systemen `dept1` und `odin` geht und schließlich über das Domänennetz vom System `odin` zum System `d998.pubs.XYZ` gelangt.

Wird häufig Post an Benutzer in anderen Netzen gesendet, kann durch die Erstellung von Alias-Namen, die die Adressen der Benutzer enthalten, Zeit gespart werden. Siehe dazu:

- "Das Postprogramm einstellen" auf Seite 7-18.
- "Alias-Namen oder Verteiler erstellen" auf Seite 7-50.
- "Definierte Alias-Namen oder Verteiler bestimmen" auf Seite 7-51.

Der Editor für elektronische Post

Das Postprogramm verfügt über einen zeilenorientierten Editor zum Erstellen von Nachrichten. Mit diesem Editor kann der Benutzer eine einzelne Zeile seiner Nachricht eingeben, die Eingabetaste drücken, und in eine neue Zeile weiteren Text eingeben. Wurde die Eingabetaste gedrückt, kann der Text in der Zeile nicht mehr geändert werden. Vor dem Drücken der Eingabetaste können jedoch mit der Rückschritt- und der Löschtaste noch Änderungen in der betreffenden Zeile vorgenommen werden. Außerdem kann mit den Unterbefehlen des Editors für elektronische Post ein Gesamtanzeigeditor aktiviert werden, mit dem die Nachricht geändert werden kann.

Einführung in den Editor für elektronische Post

Der Editor für elektronische Post wird von der Befehlszeile aus gestartet oder während die Post gelesen wird. Zum Starten von der Befehlszeile aus folgenden Befehl eingeben:

mail Benutzer@Adresse

Zum Starten, während Post gelesen wird, die Unterbefehle **R**, **r** oder **m** des elektronischen Briefkastens verwenden.

Hinweis: Mehr als ein Benutzer kann zur gleichen Zeit über die Befehlszeile angesprochen werden.

Um eine Nachricht zu senden, die mit dem Editor für elektronische Post eingegeben wurde, das Textendezeichen (normalerweise die Tastenfolge Strg-D) drücken, wenn der Cursor in der Nachricht am Anfang einer neuen Zeile steht. Der Editor kann auch gewechselt werden, um als zusätzliches Textendezeichen einen **.** (Punkt) verwenden zu können.

Der Editor für elektronische Post beinhaltet viele Steuerunterbefehle, mit denen noch andere Operationen an der Nachricht ausgeführt werden können. Jeder dieser Unterbefehle muß in einer neuen Zeile eingegeben werden und mit dem besonderen *Escape-Zeichen* **~** (Tilde) beginnen. Dieses Escape-Zeichen kann in jedes beliebige Zeichen geändert werden, indem der Eintrag **set escape** in der Datei **.mailrc** aufgenommen wird.

Hinweis: Um eine neue Zeile in der Nachricht mit einem Escape-Zeichen zu beginnen, zwei Escape-Zeichen eingeben. Zum Beispiel wird die Nachricht, die wie folgt eingegeben wird:

Dies ist eine Tilde (~), dies sind zwei (~~). Werden zwei
~~ gesendet, ist das Ergebnis nur eine.

folgendermaßen gesendet:

Dies ist eine Tilde (~), dies sind zwei (~~). Werden zwei
~ gesendet, ist das Ergebnis nur eine.

Um den Editor für elektronische Post zu verlassen, ohne die Nachricht zu senden, den Unterbefehl **~q** oder die Unterbrechungstastenfolge (normalerweise die Tastenfolge Alt-Pause oder Strg-C) verwenden. Wurde Text eingegeben, wird er mit dem Befehl **mail** in der Datei **dead.letter** gesichert.

Um eine Zusammenfassung der Unterbefehle des Editors aufzurufen, den Unterbefehl **~?** am Zeilenanfang eingeben.

Editor für elektronische Post: eine Nachricht anzeigen oder ändern

Um die Zeilen einer Nachricht anzuzeigen, den Unterbefehl `~p` verwenden. Der Editor zeigt den Inhalt der Nachricht an mit den zugehörigen Kopfdaten. Der Text wird vom unteren Rand der Anzeige her geblättert. Am Ende der Nachricht erscheint die Eingabeaufforderung des Editors für elektronische Post (`Continue`).

Ist die Nachricht länger als eine Anzeigeseite und wurde die Seitengröße für die Datenstation nicht eingestellt (siehe Befehl `stty`), wird der Text am Ende über den oberen Rand der Anzeige hinausgeschoben. Um den Inhalt von langen Nachrichten anzuzeigen, die Unterbefehle des Editors für elektronische Post verwenden, mit denen die Nachricht mit einem anderen Editor angezeigt wird.

Die Informationen einer Zeile können, wenn nach dem Drücken der Eingabetaste eine neue Zeile angezeigt wird, nicht mehr geändert werden. Der Inhalt der Nachricht kann jedoch vor dem Senden mit Hilfe eines anderen Editors geändert werden.

Eine der folgenden Tastenfolgen in einer neuen Zeile im Editor für elektronische Post eingeben, um einen der anderen Editoren zu aktivieren:

- `~e` Aktiviert den Editor `e` oder einen anderen Editor, der in der Datei `.mailrc` definiert wurde.
- `~v` Aktiviert den Editor `vi` oder einen anderen Editor, der in der Datei `.mailrc` definiert wurde.

Danach ist es möglich, den Text der aktuellen Nachricht zu editieren. Nach dem Verlassen dieses Editors wird in den Editor für elektronische Post zurückgekehrt.

Editor für elektronische Post: Kopffelder ändern oder hinzufügen

Die Überschrift einer Nachricht enthält Leitweginformationen und eine kurze Angabe des Betreffs. Es muß mindestens ein Empfänger für die Nachricht angegeben werden. Die übrigen Kopfdaten sind nicht unbedingt erforderlich. Im Kopf können u. a. folgende Informationen enthalten sein:

- `To:` Enthält die Adresse(n) der Empfänger.
- `Subject:` Enthält eine Kurzangabe des Themas der Nachricht.
- `Cc:` Enthält die Adresse(n) der Benutzer, die eine Kopie der Nachricht erhalten. Der Inhalt dieses Feldes ist Teil der Nachricht und wird an alle Empfänger gesendet.
- `Bcc:` Enthält die Adresse(n) der Benutzer, die eine *blinde* Kopie der Nachricht erhalten. Der Inhalt dieses Feldes ist *nicht* Teil der Nachricht, die an alle Empfänger gesendet wird.

Mit den Unterbefehlen des Editors für elektronische Post die Informationen in diesen Feldern ändern:

- `~h` Zum Editieren der Kopfdaten.
- `~t` Zum Hinzufügen von Empfängern in der Liste `To:`.
- `~s` Zum Einstellen oder Zurücksetzen des Feldes `Subject:`.
- `~c` Zum Hinzufügen von Empfängern in der Liste `Cc:`.
- `~b` Zum Hinzufügen von Empfängern in der Liste `Bcc:`.

Das Postprogramm kann so angepaßt werden, daß es für diese Felder automatisch Informationen anfordert. Dazu müssen in der Datei `.mailrc` Einträge angelegt sein.

Post empfangen und bearbeiten

Wird Post an die Adresse eines Benutzers gesendet, wird sie in einem eigens für Post angelegten Benutzerverzeichnis gespeichert. Dieses Benutzerverzeichnis enthält für jeden Benutzer des lokalen Systems eine Datei. In diesem Verzeichnis bleibt die Post so lange, bis der Benutzer etwas damit tut.

Über die Befehls-Shell wird der Benutzer informiert, daß Post eingetroffen ist. Der Hinweis darüber wird vor der nächsten Eingabeaufforderung angezeigt, vorausgesetzt, die Umgebungsvariable **MAIL** ist eingestellt. Und vorausgesetzt, daß das unter **MAILCHECK** angegebene Zeitintervall seit der letzten Prüfung der Shell nach Post abgelaufen ist. Die Hinweismeldung ist der Wert der Umgebungsvariablen **MAILMSG**. Je nach dem, welche Shell verwendet wird (Bourne-, K- oder C-Shell), sieht die Hinweismeldung etwa aus wie folgt:

```
YOU HAVE NEW MAIL
```

Post bearbeiten

Liegt neue Post vor und gibt der Benutzer den Befehl **mail** ein, wird eine Liste mit allen ungeöffneten Poststücken angezeigt. Von der Eingabeaufforderung des elektronischen Briefkastens aus (dem Et-Zeichen &) können Unterbefehle des Briefkastens eingegeben werden, um den Inhalt zu verwalten.

Hinweis: Bei vielen Unterbefehlen des elektronischen Briefkastens ist es möglich, Gruppen von Nachrichten anzugeben.

Mit den Unterbefehlen stehen folgende Bearbeitungsoptionen zur Verfügung:

- Eine Nachricht weiterleiten
- Nachrichten anzeigen
- Eine Nachricht löschen
- Eine Nachricht in einer Datei sichern
- Eine Liste von Bearbeitungsoptionen anzeigen
- Eine Nachricht ignorieren
- Eine Nachricht anhängen
- Eine Nachricht beantworten
- Eine Nachricht im elektronischen Systembriefkasten lassen
- Eine Nachricht im persönlichen elektronischen Briefkasten (**mbox**) speichern
- Eine Nachricht in einem persönlich erstellten Ordner speichern

Post weiterleiten

Ist der Benutzer nicht unter seiner normalen Netzadresse zu erreichen, möchte er möglicherweise seine Post an eine andere Netzadresse gesendet haben. Das Senden der eingehenden Post an eine andere Adresse (bzw. andere Adressen) wird "weiterleiten" genannt.

Die neue Adresse kann eine beliebige gültige Adresse im eigenen Netz oder einem verbundenen Netz sein. Es kann beispielsweise die Adresse eines Kollegen sein, der während der Abwesenheit die Post bearbeitet. Wird die Post weitergeleitet, erhält der Benutzer keine Kopie der eingehenden Post in seinem elektronischen Briefkasten. Die gesamte Post wird direkt an die angegebene(n) Adresse(n) weitergeleitet.

Informationen über die Prozedur des Weiterleitens der Post enthält der Abschnitt "Post weiterleiten" auf Seite 7-45.

Der elektronische Briefkasten und die Postordner

Im Postprogramm wird die Post auf unterschiedliche Weise gespeichert.

Der elektronische Systembriefkasten

Der elektronische Systembriefkasten ist wie ein Postfach. Das Postamt stellt Briefe zu, die an den Besitzer des Postfachs adressiert sind. Der elektronische Systembriefkasten ist eine Datei, die einem bestimmten Benutzer zugeordnet ist. Diese Datei wird erstellt, wenn für eine Benutzer-ID Post eintrifft. Sie wird wieder gelöscht, wenn alle Nachrichten entfernt wurden. Für jede Benutzer-ID existiert ein separater elektronischer Systembriefkasten, der in der Datei `/etc/passwd` definiert ist. Das Postprogramm speichert alle elektronischen Systembriefkästen im Verzeichnis `/var/spool/mail`. Jeder elektronische Systembriefkasten ist nach der zugeordneten Benutzer-ID benannt. Für die Benutzer-ID `karen` beispielsweise lautet der elektronische Systembriefkasten `/var/spool/mail/karen`.

Geht für eine Benutzer-ID Post ein, stellt das Postprogramm sie in den betreffenden elektronischen Systembriefkasten. Ist der Benutzer angemeldet, schickt das Postprogramm eine Nachricht an die Datenstation. Ist der Benutzer nicht angemeldet, schickt das Postprogramm bei der Anmeldung eine Nachricht an die Datenstation. Die Standardnachricht sieht etwa aus wie folgt:

YOU HAVE NEW MAIL

Mit dem Befehl `mail` kann die Nachricht gelesen und aus dem elektronischen Systembriefkasten gelöscht werden. Nicht den elektronischen Systembriefkasten zum Speichern der Nachrichten verwenden, sondern den persönlichen elektronischen Briefkasten (`$HOME/mbox`) und Ordner.

Der persönliche elektronische Briefkasten

Der persönliche elektronische Briefkasten ähnelt einem Posteingangskorb in einem Büro. Die Post wird nach dem Empfangen in einen Posteingangskorb gelegt, bevor sie in einer Datei abgespeichert wird. Der persönliche elektronische Briefkasten ist der Arbeitsspeicherplatz für Post, die noch bearbeitet werden muß.

Im Postprogramm ist der persönliche elektronische Briefkasten eine Datei, die einem bestimmten Benutzer zugeordnet ist. Wird Post aus dem elektronischen Systembriefkasten gespeichert, erstellt das Postprogramm eine Datei mit dem Namen `$HOME/mbox` (wobei `$HOME` das Anmeldeverzeichnis des Benutzers ist). Die Datei `mbox` besteht nur, wenn eine Nachricht darin enthalten ist. Nachrichten, die zwar gelesen wurden, aber nicht gelöscht, in eine Datei geschrieben oder in einem Ordner gespeichert wurden, werden in der Datei `mbox` gespeichert, bis sie verschoben oder gelöscht werden.

Die Datei `dead.letter`

Kann das Zustellungssystem eine Nachricht nicht zustellen, stellt es sie in die Datei `dead.letter` im Verzeichnis `$HOME`. Besteht die Datei `dead.letter` noch nicht, wird sie vom Zustellungssystem erstellt. Das Zustellungssystem zeigt eine Nachricht an, daß die Post nicht zugestellt werden konnte. Mögliche Gründe für einen Fehler bei der Zustellung:

Benutzer unbekannt

Die in der Adresse der Nachricht angegebene Benutzer-ID ist keine für das angegebene System definierte Benutzer-ID oder kein definierter Alias-Name.

Host unbekannt

Der den Host oder die Domäne betreffende Teil der Adresse in der Nachricht ist fehlerhaft. Es kann sich dabei um einen Syntaxfehler oder einen falschen Host- oder Domännennamen handeln.

Das System verwendet die Datei **dead.letter** außerdem zum Sichern unvollständiger Nachrichten; zum Beispiel, wenn der Editor für elektronische Post mit der Unterbrechungstastenfolge (Strg-C) verlassen wird, ohne die Nachricht zu senden. Der vorherige Inhalt der Datei **dead.letter** wird durch die unvollständige Nachricht ersetzt. Mit dem Unterbefehl **~d** wird der Inhalt der Datei **dead.letter** abgerufen.

Achtung: Die Datei **dead.letter** nicht verwenden, um die Nachricht zu speichern. Der Inhalt der Datei wird überschrieben.

Ordner

In Ordnern können Nachrichten übersichtlich gespeichert werden. Dabei können beliebig viele Ordner erstellt werden. Sie sollten Namen erhalten, die auf den Inhalt der dort abgelegten Nachrichten hinweisen, Aktenordnern in einem Büro vergleichbar. Mit dem Postprogramm können von folgenden Bereichen aus Nachrichten in einen Ordner gestellt werden:

- elektronischer Systembriefkasten
- persönlicher elektronischer Briefkasten
- Datei **dead.letter**
- ein anderer Ordner

Jeder Ordner ist eine Textdatei und wird in dem in der Datei **.mailrc** angegebenen Verzeichnis abgelegt. Dieses Verzeichnis muß zunächst erstellt werden, bevor Ordner zum Speichern von Nachrichten verwendet werden können. Sobald das Verzeichnis existiert, erstellt das Postprogramm dort die notwendigen Ordner. Weitere Informationen befinden sich im Abschnitt "Ordner im elektronischen Briefkasten verwenden" auf Seite 7-43.

Unterbefehle des elektronischen Briefkastens im Postprogramm

Wenn das Postprogramm einen elektronischen Briefkasten verarbeitet, zeigt es eine Eingabeaufforderung des Briefkastens an und wartet auf eine Eingabe. Die Eingabeaufforderung des elektronischen Briefkastens ist ein Et-Zeichen (&) am Anfang einer neuen Zeile. An der Eingabeaufforderung kann ein beliebiger Unterbefehl des elektronischen Briefkastens eingegeben werden. Bei vielen Unterbefehlen des Briefkastens können ein oder mehrere Parameter angegeben werden:

- Nachrichtenliste
- Datei- oder Ordnername.

Eine Nachrichtenliste angeben

Viele Unterbefehle des Briefkastens wirken sich auf eine Gruppe von Nachrichten aus. Zur näheren Angabe der Nachrichten die in der Liste des Briefkasteninhalts angezeigten Daten verwenden, z. B. Nachrichtennummer oder Absender. Befehle, mit denen die Angabe einer Gruppe von Nachrichten möglich ist, verwenden den Parameter *Nachrichtenliste*. Wird kein Wert für den Parameter *Nachrichtenliste* angegeben, wird der Standardwert, die aktuelle Nachricht, angenommen.

Folgende Unterbefehle des elektronischen Briefkastens verwenden den Parameter *Nachrichtenliste*: **d** (löschen), **e** (editieren), **v** (ausführlich), **h** (Kopfzeilen), **n** (nächste), **s** (sichern), **t** (eingeben) und **w** (schreiben).

Das Format für den Unterbefehl **t** (ausgewählte Nachrichten eingeben) lautet beispielsweise **t Nachrichtenliste**. Die Parameter *Nachrichtenliste* können sein:

- Eine oder mehrere Nachrichtennummern, durch ein Leerzeichen getrennt:

```
t 1 2 4 7
```

- Ein Bereich von Nachrichtennummern, wobei die erste und die letzte angegeben wird, durch einen Bindestrich getrennt.

```
t 2-5
```

- Eine oder mehrere Adressen, durch ein Leerzeichen getrennt, um den Befehl auf Nachrichten anzuwenden, die von diesen Adressen empfangen wurden:

```
t amy geo@zeus
```

Hinweis: Im Adreßfeld einer Nachricht kann die Adresse in Groß- oder Kleinbuchstaben enthalten sein. Die beim Unterbefehl eingegebene Adresse muß damit nicht exakt übereinstimmen. Beipielsweise werden bei der Anfrage für die Adresse amy folgende Übereinstimmungen gefunden:

- amy
- AmY
- amy@zeus
- hamy

- Eine Zeichenfolge, der ein Schrägstrich vorangeht, um Übereinstimmungen mit dem Feld **Subject**: der Nachrichten zu suchen:

```
t /pol
```

Hinweis: Die Zeichenfolge wendet den Befehl auf alle Nachrichten an, deren Feld **Subject**: die Buchstaben pol (in Groß- oder Kleinbuchstaben) enthält. Die eingegebenen Zeichen für das Vergleichsmuster müssen nicht exakt mit dem Feld **Subject**: übereinstimmen. Das Postprogramm gleicht die Zeichen ab, die an einer beliebigen Stelle im Feld **Subject**: vorkommen. Beispielsweise werden bei der Anfrage für pol folgende Übereinstimmungen gefunden:

- Polizeiwagen überfallen
- Politik im Wandel
- Pole schmelzen

- Besondere Platzhalterzeichen:

- Der Stern (*) gleicht alle Nachrichten ab.
- Das Winkelzeichen (^) gleicht die erste Nachricht ab.
- Das Dollarzeichen (\$) gleicht die letzte Nachricht ab.

Datei- und Ordnernamen angeben

Bei vielen Unterbefehlen des elektronischen Briefkastens können Datei- oder Ordnernamen angegeben werden. Diese Unterbefehle verwenden den Parameter *Dateiname*. Folgende Unterbefehle des elektronischen Briefkastens verwenden den Parameter *Dateiname*: **c** (kopieren), **file** (Datei), **s** (sichern) und **w** (schreiben). Zum Beispiel ist das Format für den Unterbefehl **file**:

```
file Dateiname
```


Bei diesem Format kann der Parameter *Dateiname* einer der folgenden sein:

- Der Pfadname des neuen elektronischen Briefkastens in Bezug auf das aktuelle Verzeichnis. Ist das aktuelle Verzeichnis beispielsweise das Benutzerverzeichnis, kann mit diesem Unterbefehl zum persönlichen elektronischen Briefkasten gewechselt werden:

```
file mbox
```

Das Programm wechselt zu diesem elektronischen Briefkasten und zeigt eine Liste mit dessen Inhalt an.

- Der absolute Pfadname des neuen elektronischen Briefkastens. Für die Benutzer-ID *george* beispielsweise den folgenden Unterbefehl verwenden, um zum elektronischen Systembriefkasten zu wechseln:

```
file /usr/mail/george
```

- Eine Kurzform eines Ordnersnamens im eigenen Verzeichnis. Ist beispielsweise das eigene Ordnerverzeichnis als *letters* definiert, den folgenden Unterbefehl verwenden, um zum Ordner *reports* zu wechseln:

```
file +reports
```

Das Pluszeichen (+) ist dabei die Kurzform des vollständigen Pfadnamens für das Ordnerverzeichnis. Dieser Unterbefehl führt dieselbe Funktion aus, als wenn er folgendermaßen eingegeben wäre:

```
file /home/george/letters/reports
```

Hinweis: Weitere Informationen über die Unterbefehle des Postprogramms enthält der Abschnitt "Verwendung von Unterbefehlen zum Senden von Post" auf Seite 7-25.

Das Postprogramm einstellen

Befehle und Optionen in den Dateien `.mailrc` und `/usr/share/lib/Mail.rc` betreffen jede Postsitzung. Diese Dateien können den persönlichen Bedürfnissen des Benutzers angepaßt werden.

Postoptionen einstellen

Den Befehl `pg` verwenden, um die Datei `/usr/share/lib/Mail.rc` oder die Datei `.mailrc` anzuzeigen. Der Inhalt der Datei `/usr/share/lib/Mail.rc` definiert die Konfiguration des Postprogramms. Die Systemkonfiguration für das Postprogramm ändern, indem eine Datei `$.HOME/.mailrc` erstellt wird. Bei Ausführung des Befehls `mail` überschreiben die in der Datei `.mailrc` enthaltenen Unterbefehle die entsprechenden Unterbefehle in der Datei `/usr/share/lib/Mail.rc`. Die Optionen in der Datei `.mailrc` können angepaßt werden und sind dann bei jeder Postsitzung gültig.

Beim Lesen der Post den Befehl `set` ohne Argument eingeben, so daß alle aktivierten Optionen der Datei `.mailrc` aufgelistet werden. Dieser Liste ist auch zu entnehmen, ob ein Ordnerverzeichnis ausgewählt wurde und ob eine Protokolldatei definiert wurde, um abgehende Nachrichten aufzuzeichnen.

Die folgende Liste enthält einige der Kenndaten einer Postsitzung, die angepaßt werden können:

Eingabeaufforderungen für den Betrefftext einer Nachricht

Bei Eingabe des Befehls `mail` fordert das Programm möglicherweise dazu auf, das Feld `Subject`: auszufüllen. Erscheint diese Eingabeaufforderung, kann der Benutzer eine Kurzangabe über das Thema der Nachricht eingeben. Dieser Betrefftext erscheint, wenn der Empfänger die Nachricht liest, am Anfang des Textes.

Eingabeaufforderungen, welche Benutzer eine Kopie erhalten sollen

Die Datei `.mailrc` kann so angepaßt werden, daß das Postprogramm beim Senden einer Nachricht dazu auffordert, die Namen der Benutzer einzugeben, die eine Kopie der Nachricht erhalten sollen.

Alias-Namen oder Verteiler

Sendet der Benutzer Post innerhalb eines großen Netzes oder sendet er häufig dieselbe Nachricht an eine größere Gruppe von Empfängern, wird das Eingeben langer Adressen für jeden Empfänger mühsam. Dieser Prozeß kann durch das Erstellen eines Alias-Namens oder eines Verteilers in der Datei `.mailrc` vereinfacht werden:

Alias-Name Ein vom Benutzer definierter Name, der anstelle der Adresse eines Einzelbenutzers verwendet werden kann.

Verteiler Ein vom Benutzer definierter Name, der anstelle der Adresse einer Gruppe von Benutzern verwendet werden kann.

Anzahl der beim Lesen angezeigten Zeilen

Die Anzahl der Zeilen des Nachrichtenkopfes oder -textes, die pro Anzeige erscheinen, kann geändert werden.

Daten, die in Nachrichten angezeigt werden

Nachrichtenköpfe, wie zum Beispiel das automatisch eingesetzte Feld `message-id` können ausgeschaltet werden.

Ein Ordnerverzeichnis zum Speichern von Nachrichten

Zum Speichern von Nachrichten kann ein Sonderverzeichnis erstellt werden. Beim Speichern von Nachrichten und Ansehen von Ordnern kann dann der Kurzbefehl + (Pluszeichen) verwendet werden, um dieses Verzeichnis zu kennzeichnen.

Eine Protokolldatei zum Aufzeichnen abgehender Nachrichten

Das Postprogramm kann so angepaßt werden, daß alle abgehenden Nachrichten in einer Datei oder einem Unterverzeichnis des Benutzerverzeichnisses aufgezeichnet werden.

Editoren zum Eingeben von Nachrichten

Neben dem Editor für elektronische Post können zwei weitere Editoren für Nachrichten ausgewählt werden (**e3** oder **vi**).

Den Editor für elektronische Post verlassen

Der Befehl zum Verlassen des Editors für elektronische Post kann geändert werden.

Speichern von Nachrichten beim Postprogramm

Die Speicherstelle für Nachrichten, die bereits gelesen wurden, kann geändert werden.

Optionen aktivieren und inaktivieren

Die vier Unterbefehle des elektronischen Briefkastens, mit denen am häufigsten die Kenndaten einer Postsitzung verändert werden, lauten:

set	Aktiviert die Postoptionen.
unset	Inaktiviert die Postoptionen.
alias	Verkürzt die Adressierung der Post.
ignore	Unterdrückt die Nachrichtenkopffelder.

Diese Optionen können entweder innerhalb des elektronischen Briefkastens eingestellt werden oder durch entsprechende Einträge in der Datei **.mailrc**. Optionen können entweder binär oder mit Werten versehen sein. Binäre Optionen sind entweder **set** (aktiviert) oder **unset** (inaktiviert), während ansonsten ein bestimmter Wert eingestellt wird.

Um Postbefehle auszuführen, die in einer Datei gespeichert sind, den Befehl **source** verwenden. Beim Lesen von Post kann dieser Befehl an der Eingabeaufforderung des elektronischen Briefkastens eingegeben werden:

```
& source Pfadname
```

wobei *Pfadname* den Pfad und die Datei angibt, in der die Postbefehle enthalten sind. Die Befehle in dieser Datei überschreiben für die Dauer der aktuellen Sitzung die vorherigen Einstellungen ähnlicher Befehle. Die Kenndaten der aktuellen Postsitzung können außerdem durch die Eingabe von Befehlen an der Eingabeaufforderung des elektronischen Briefkastens geändert werden.

Die Nachrichtenanzeige im Postprogramm ändern

Der Benutzer kann einige Optionen in seiner Datei **.mailrc** einstellen, um die Anzeige von Informationen in unterschiedlichen Situationen zu steuern.

In Anzeigen blättern

Durch Änderungen in der Datei **.mailrc** wird die Fähigkeit, durch Listen des elektronischen Briefkastens oder durch tatsächlichen Nachrichtentext zu blättern, gesteuert.

In der Nachrichtenliste

Jede Nachricht im elektronischen Briefkasten verfügt über einen einzeiligen Kopf in der Nachrichtenliste. Sind mehr als 24 Nachrichten vorhanden, verschwindet der erste Kopf aus der Liste am oberen Anzeigenrand. Der Befehl **set screen** steuert, wieviele Zeilen einer Liste gleichzeitig angezeigt werden. Besteht zum Beispiel folgender Eintrag in der Datei **.mailrc**:

```
set screen=20
```

werden nach Eingabe des Befehls **h 20** Nachrichtenkopfzeilen angezeigt. Dann muß der Benutzer erst die Eingabetaste drücken, bevor weitere 20 Kopfzeilen angezeigt werden.

In einer langen Nachricht

Wird eine Nachricht mit mehr als 24 Zeilen aufgerufen, verschwinden die ersten Zeilen der Nachricht am oberen Anzeigenrand. Um durch lange Nachrichten zu blättern, kann das Programm **pg** innerhalb des Postprogramms verwendet werden. Voraussetzung dafür ist, daß der Befehl **set crt** in der Datei **.mailrc** enthalten ist. Der Befehl **set crt** steuert, wieviele Zeilen eine Nachricht lang sein muß, bevor das Programm **pg** gestartet wird.

Wird zum Beispiel der Befehl **t** verwendet, um eine lange Nachricht zu lesen, wird nur eine Anzeige (eine Seite) angezeigt. Am Ende der Seite folgt eine Eingabeaufforderung in Form eines Doppelpunktes, die anzeigt, daß noch mehr Text vorhanden ist. Die Eingabetaste drücken, um die nächste Seite der Nachricht anzuzeigen. Nach der letzten Seite der Nachricht folgt eine Eingabeaufforderung ähnlich der folgenden:

EOF:

An dieser Eingabeaufforderung kann der Benutzer jeden gültigen **pg**-Unterbefehl eingeben. Vorherige Seiten können aufgerufen werden oder die Nachricht kann nach Zeichenfolgen durchsucht werden. Die Nachricht kann allerdings auch verlassen werden, woraufhin die Eingabeaufforderung des elektronischen Briefkastens (**&**) erscheint.

Der Befehl **set crt** wird wie folgt in der Datei **.mailrc** eingegeben:

```
set crt=Zeilen
```

Zum Beispiel gibt:

```
set crt=20
```

an, daß eine Nachricht 20 Zeilen lang sein muß, bevor das Programm **pg** gestartet wird. Das Programm **pg** wird hier also gestartet, wenn der Benutzer eine Nachricht mit mehr als 20 Zeilen lesen will.

Am Anfang der Nachricht

Mit dem Befehl **top** ist es möglich, einen Teil einer Nachricht zu überfliegen, und nicht die ganze Nachricht lesen zu müssen. Der Benutzer kann steuern, wieviele Zeilen einer Nachricht angezeigt werden, indem er die Option **toplines** wie folgt einstellt:

```
set topline=Zeilen
```

In diesem Befehl bezieht sich die Variable *Zeilen* auf die Anzahl der Zeilen, vom Anfang der Nachricht einschließlich aller Kopffelder an, die nach Eingabe des Befehls **top** angezeigt werden.

Steht z. B. bei Benutzer Amy folgende Zeile in der Datei `.mailrc`:

```
set topline=10
```

erscheint, wenn der Benutzer Amy den Befehl `mail` ausführt, um neue Nachrichten zu lesen, folgende Anzeige:

Mail Mit ? kann Hilfetext abgerufen werden.

```
"/usr/mail/amy": 2 Nachrichten 2 neu>
```

```
N 1 george Wed Jan 6 9:47 11/257 "Abteilungsmeeting"
```

```
N 2 mark Wed Jan 6 12:59 17/44 5 "Projektplanung"
```

```
&
```

Verwendet der Benutzer Amy den Befehl `top`, um durch die Nachrichten zu blättern, wird nur eine Teilnachricht angezeigt:

```
& top 1
```

```
Nachricht 1:
```

```
From george Wed Jan 6 9:47 CST 1988
```

```
Received: by zeus
```

```
id AA00549; Wed, 6 Jan 88 9:4 7:46 CST
```

```
Date: Wed, 6 Jan 88 9:47:46 CST
```

```
From: george@zeus
```

```
Message-Id: <8709111757.AA00178>
```

```
To: amy@zeus
```

```
Subject: Abteilungsmeeting
```

```
Bitte am Freitag um 13.30 h das Abteilungsmeeting einplanen.
```

```
Ort: Besprechungsraum.
```

```
&
```

Die Nachricht wird nur teilweise angezeigt, da in der Option `toplines 10` angegeben wurde. Nur Zeile 1 (das Feld `Received:`) bis Zeile 10 (die zweite Zeile des Nachrichtentextes) werden angezeigt. Die oberste Zeile, `From george Wed Jan 6 9:47 CST 1988`, wird immer angezeigt, zählt aber nicht für die Option `toplines`.

Art der angezeigten Informationen steuern

Durch Änderungen in der Datei `.mailrc` kann gesteuert werden, welche Informationen in einer Nachricht angezeigt werden.

In den Kopffeldern

Jede Nachricht enthält am Anfang mehrere Kopffelder. Diese Kopffelder werden beim Lesen einer Nachricht angezeigt. Mit dem Befehl `ignore` kann das Anzeigen der Kopffelder beim Lesen unterdrückt werden. Das Format für den Befehl `ignore` ist folgendes:

```
ignore [Feldliste]
```

Die Variable `Feldliste` kann aus einem oder mehreren Feldnamen bestehen, die beim Anzeigen der Nachricht *ignoriert* werden sollen. Steht z. B. bei Benutzer Amy folgende Zeile in der Datei `.mailrc`:

```
ignore date from to
```

und die Datei `/usr/share/lib/Mail.rc` enthält die Zeile:

```
ignore received message-id
```

sieht das Ergebnis nach Eingabe des Befehls `t` wie folgt aus:

& t 1

Message 1:

From george Wed Jan 6 9:47 CST 1988

Subject: Abteilungsmeeting

Bitte am Freitag um 13.30 h das Abteilungsmeeting einplanen.

Ort: Besprechungsraum.

Thema: Projektplanungsprogramm

&

Die Felder `Received:`, `Date:`, `From:`, `Message-Id:`, und `To:` werden nicht angezeigt. Um diese Felder anzuzeigen, die Befehle **T** oder **P** (in Großbuchstaben) verwenden oder den Befehl **top**.

Hinweis: In diesem Beispiel wird die Zeile `From` angezeigt. Dabei handelt es sich nicht um das Feld `From:`, das für die Variable *Feldliste* des Befehls **ignore** angegeben wurde.

Der Befehl **ignore** kann außerdem ohne Argument in der Befehlszeile **mail** eingegeben werden, um eine Liste der ignorierten Kopffelder aufzurufen.

In der Namensanzeige

Die *Namensanzeige* enthält den Namen des Postprogramms und die Nachrichtennummer, die bei Eingabe des Befehls **mail** angezeigt werden. Wird die Option **quiet** in der Datei `.mailrc` wie folgt eingestellt, erscheint die Namensanzeige nicht:

```
set quiet
```

Eine weitere Option, mit der die Namensanzeige **mail** unterdrückt werden kann, ist:

```
set noheader
```

Steht dieser Befehl in der Datei `.mailrc`, wird die Nachrichtenliste des elektronischen Briefkastens nicht angezeigt. Wird das Postprogramm gestartet, erscheint nur die Eingabeaufforderung des elektronischen Briefkastens (&). Eine Liste der Nachrichten kann durch Eingabe des Befehls **h** aufgerufen werden.

Den Löschbefehl mit dem Druckbefehl kombinieren

Nach dem Lesen kann die Nachricht mit dem Unterbefehl **d** gelöscht werden. Die nächste Nachricht kann mit dem Unterbefehl **p** aufgerufen werden. Durch die Eingabe der folgenden Zeile in der Datei `.mailrc` können diese Befehle kombiniert werden:

```
set autoprnt
```

Steht der Befehl **set autoprnt** in der Datei `.mailrc`, wird mit dem Unterbefehl **d** die aktuelle Nachricht gelöscht und die nächste angezeigt.

Post senden

Voraussetzungen

- Das Postprogramm muß auf dem System installiert sein.
- Der Name und die Adresse des Empfängers muß bekannt sein.

Erforderliche Schritte

1. Den Befehl **mail** in der Befehlszeile eingeben, gefolgt von dem Namen und der Adresse des/der Empfänger der Nachricht. Zum Beispiel:

```
mail jan@brown
```

Das System gibt folgende Antwort aus:

```
Subject:
```

2. Den Betrefftext der Nachricht eingeben. Zum Beispiel:

```
Subject: Abteilungsmeeting
```

Danach die Eingabetaste drücken. Nun kann der Hauptteil des Textes eingegeben werden.

3. Die Nachricht eingeben.

```
Heute findet ein kurzes Abteilungsmeeting in meinem Büro statt.  
Bitte den frühen Nachmittag dafür freihalten.
```

Hinweis: Der Editor für elektronische Post verfügt über einen automatischen Zeilenumbruch. Es muß nicht an jedem Zeilenende die Eingabetaste gedrückt werden.

4. Ist die Eingabe der Nachricht beendet, den Cursor durch Drücken der Eingabetaste zu einer neuen Zeile bewegen. Entweder . (Punkt) oder das Textendezeichen (Strg-D bei den meisten Datenstationen) eingeben. Das System zeigt das Feld Cc: ("Kopie zur Kenntnisnahme") an:

```
Cc:
```

5. Die Namen und die Adressen der Benutzer eingeben, die eine Kopie der Nachricht erhalten sollen, und die Eingabetaste drücken. Zum Beispiel:

```
Cc: karen@hobo cliff@cross
```

Hinweis: Sollen keine Kopien gesendet werden, nur die Eingabetaste drücken.

6. Nach dem Ausfüllen des Feldes Cc: und dem Drücken der Eingabetaste wird die Nachricht gesendet. Die Eingabeaufforderung wird angezeigt.

Hinweis: Wird ein Name oder eine Adresse eingegeben, die dem System nicht bekannt sind oder die nicht als Alias-Name oder in einem Verteiler definiert sind, gibt das System die betreffende Benutzer-ID, gefolgt von einer Fehlermeldung, aus:

```
Benutzer-ID]... User unknown
```

Empfohlene Dokumentation

Grundlegende Informationen

"Post erstellen und senden" auf Seite 7-3.

Zugehörige Informationen

"Texteditor zur Eingabe von Nachrichten wechseln" auf Seite 7-54.

Verwendung von Unterbefehlen zum Senden von Post

Voraussetzungen

- Das Postprogramm muß auf dem System installiert sein.
- Der Name und die Adresse des Empfängers müssen bekannt sein.

Erforderliche Schritte

Im folgenden wird die Verwendung von Unterbefehlen des Postprogramms `Senden von Post` erläutert.

Eine Datei in eine Nachricht einfügen

Um eine Datei in eine aktuelle Nachricht einzufügen, am Anfang einer neuen Zeile folgenden Befehl eingeben:

```
~r Plan
```

Dabei ist `Plan` der Name der Datei, die eingefügt werden soll. In diesem Beispiel werden die Informationen der Datei `Plan` am aktuellen Ende der Nachricht eingefügt.

Eine bestimmte Nachricht in eine Nachricht einfügen

Um eine bestimmte Nachricht in eine Nachricht einzufügen, am Anfang einer neuen Zeile folgenden Befehl eingeben:

```
~m 2
```

Dabei ist `2` die Nummer der Nachricht, die eingefügt werden soll. Dieser Befehl hängt die angegebene Nachricht an das Ende der aktuellen Nachricht an. Die eingefügte Nachricht ist um ein Tabulatorzeichen vom normalen linken Rand der Nachricht eingerückt. Dieser Unterbefehl ist nur gültig, wenn der Editor für elektronische Post von der Eingabeaufforderung des elektronischen Briefkastens aus aufgerufen wurde, und zwar mit dem Befehl `mail`, `reply` oder `Reply`.

Außerdem kann eine Nachricht mit folgendem Befehl eingefügt werden:

```
~f Nachrichtenliste
```

Dieser Befehl hängt die angegebene(n) Nachricht(en) an das Ende der aktuellen Nachricht an, aber *ohne sie einzurücken*. Diesen Unterbefehl ebenfalls verwenden, um Nachrichten anzuhängen, die zu breit sind, um mit dem Unterbefehl `~m` angehängt werden zu können.

Hinweise:

1. Um die Unterbefehle `~m` und `~f` verwenden zu können, muß der Editor vom elektronischen Briefkasten aus aufgerufen worden sein (entweder mit dem Unterbefehl `r`, `R` oder `m` des elektronischen Briefkastens).
2. In dem vorangegangenen Format bezieht sich der Parameter *Nachrichtenliste* auf eine Liste ganzer Zahlen, d. h. der gültigen Nachrichtennummern im elektronischen Briefkasten oder im Ordner, der vom Postprogramm bearbeitet wird. Es können also auch Bereiche von Zahlen eingegeben werden. Zum Beispiel:

```
~m 1
```

Hängt Nachricht 1 an das Ende der aktuellen Nachricht an, um ein Tabulatorzeichen vom linken Rand eingerückt.

```
~m 1 3
```

Hängt Nachricht 1 und dann Nachricht 3 an das Ende der aktuellen Nachricht an, um ein Tabulatorzeichen vom linken Rand eingerückt.

~f 1-4

Hängt die Nachrichten 1, 2, 3 und 4 an das Ende der aktuellen Nachricht an. Diese Nachrichten sind mit dem linken Rand ausgerichtet (nicht eingerückt).

Den Inhalt des Betreffeldes ändern

Um den Inhalt des Betreffeldes zu ändern, am Anfang einer neuen Zeile folgenden Befehl eingeben:

~s Urlaub

In diesem Beispiel wird das Betreffeld in Urlaub geändert.

Den Editor verlassen

Um den Editor zu verlassen, am Anfang einer neuen Zeile folgenden Befehl eingeben:

~q

Die Nachricht wird nicht gesendet. Sie wird in der Datei **dead.letter** im Benutzerverzeichnis gespeichert, es sei denn, es wurde kein Text eingegeben. Die Eingabeaufforderung wird angezeigt.

Den Editor mit einer Unterbrechungstaste verlassen

Um den Editor mit einer Unterbrechungstaste zu verlassen, die Tastenfolge Strg-C oder Alt-Pause drücken. Folgende Nachricht wird angezeigt:

(Unterbrechung -- Um Nachricht abubrechen, Taste erneut drücken)

Die Unterbrechungstaste erneut drücken.

(Letzte Unterbrechung -- Nachricht in der Datei **dead.letter** gespeichert)

Die Nachricht wird nicht gesendet. Sie wird in der Datei **dead.letter** im Benutzerverzeichnis gespeichert, es sei denn, es wurde kein Text eingegeben. Die Eingabeaufforderung wird angezeigt.

Hinweis: Wird nach der ersten Zeile Text die Tastenfolge Strg-C gedrückt, bevor die Eingabetaste gedrückt war, wird die Nachricht nicht in der Datei **dead.letter** gespeichert.

Den Inhalt der Datei **dead.letter** in die aktuelle Nachricht einfügen

Kann das Postprogramm eine Nachricht nicht zustellen, stellt es sie in die Datei **dead.letter** im Benutzerverzeichnis. Die Datei **dead.letter** kann auch Teile eines Briefes enthalten, die gespeichert wurden, als der Editor für elektronische Post über den Befehl ~q verlassen wurde. Um den Inhalt der Datei **dead.letter** in die aktuelle Nachricht einzufügen, folgenden Befehl eingeben:

~d

Dieser Befehl hängt den Inhalt der Datei **dead.letter** an das Ende der aktuellen Nachricht an. An der Eingabeaufforderung weiteren Text eingeben oder die Nachricht senden.

Einen alternativen Editor starten und den Nachrichtentext als Eingabedatei verwenden

Um einen alternativen Editor zu starten und den Nachrichtentext als Eingabedatei zu verwenden, am Anfang einer neuen Zeile folgenden Befehl eingeben:

~e

Hinweis: Der alternative Editor kann in der Datei **.mailrc** definiert werden mit:

```
set EDITOR=Pfadname
```

Hiermit wird der Editor definiert, der bei Eingabe des Unterbefehls `~e` aktiviert wird. Der Wert von *Pfadname* muß der vollständige Pfadname zum gewünschten Editor sein. Zum Beispiel bewirkt die Definition `set EDITOR=/usr/bin/vi`, daß der Editor `vi` bei Eingabe des Unterbefehls `~e` verwendet wird.

Empfohlene Dokumentation

Grundlegende Informationen

"Post erstellen und senden" auf Seite 7-3.

Zugehörige Informationen

"Texteditor zur Eingabe von Nachrichten wechseln" auf Seite 7-54.

Kopffelder in einer Nachricht der elektronischen Post ändern und hinzufügen

Voraussetzungen

1. Das Postprogramm muß auf dem System installiert sein.
2. Den Editor für elektronische Post starten und beginnen, eine Nachricht zu editieren.

Erforderliche Schritte

Unter Verwendung der Unterbefehle des Editors für elektronische Post können, wie im folgenden beschrieben, die Kopffelder in einer Nachricht geändert werden.

Kopffelder editieren

Um Informationen in mehr als einem Kopffeld zu ändern oder hinzuzufügen, den Unterbefehl `~h` verwenden. Wird dieser Unterbefehl in einer neuen Zeile eingegeben, zeigt das System jedes der vier Kopffelder nacheinander an. Folgende Optionen stehen dann zur Verfügung:

- Den Inhalt jedes Feldes anzeigen.
- Informationen aus dem Feld löschen (mit der Rückschrittaste).
- Information in diesem Feld hinzufügen.

Durch Drücken der Eingabetaste werden die Änderungen in diesem Feld gespeichert und das nächste Feld mit seinem Inhalt wird angezeigt. Wird beim letzten Feld (`Bcc:`) die Eingabetaste gedrückt, wird zum Editor zurückgekehrt.

Beispiel

1. In der Nachricht am Anfang einer neuen Zeile `~h` eingeben.

Das System zeigt den Inhalt des Feldes `To:` an und setzt den Cursor an das Ende dieses Feldes:

```
To: mark@austin_
```

Falls notwendig, das Feld bearbeiten. Danach die Eingabetaste drücken.

2. Das System zeigt den Inhalt des Feldes `Subject:` an:

```
Subject: Segelausflug
```

Hinweis: Wurde dieses Feld geändert, steht der Cursor möglicherweise nicht am Ende des Feldes.

Um den Tippfehler im Betreffeld zu korrigieren, die Tasten `Cursor` nach rechts und `Cursor` nach links verwenden, um den Cursor unter das `u` in `Segelausflug` zu stellen. Den Betrefftext korrigieren (`Segelausflug`). Die Eingabetaste drücken.

3. Das System zeigt den Inhalt des Feldes `Cc:` an:

```
Cc: mel@gtwn_
```

Um einen anderen Benutzer zu der Kopieempfängerliste hinzuzufügen, sicherstellen, daß der Cursor am Ende der Liste steht, ein Leerzeichen und dann die neue Adresse eingeben:

```
Cc: mel@gtwn geo@austin
```

Die Kopieempfängerliste wird hierdurch auf zwei Benutzer erweitert. Ist die Kopieempfängerliste vollständig, die Eingabetaste drücken.

4. Das System zeigt den Inhalt des Feldes `Bcc:` an. Eventuelle Änderungen eingeben und die Eingabetaste drücken.
5. Das System zeigt die Eingabeaufforderung an und kehrt zum Editor für elektronische Post an das Ende der Nachricht zurück.

Die Zielliste (To:) erweitern

Den Unterbefehl `~t` verwenden, um eine oder mehrere Adressen der Liste `To:` hinzuzufügen. Die Liste `To:` enthält zum Beispiel für eine Nachricht folgende Adresse:

```
To: mark@austin
```

Um diese Liste zu erweitern, den Unterbefehl `~t subcommand`, wie folgt eingeben:

```
~t geo@austin mel@gtwn
```

Damit wird die Liste geändert in:

```
To: mark@austin geo@austin mel@gtwn
```

Hinweis: Der Unterbefehl `~t` kann nicht verwendet werden, um den Inhalt der Liste `To:` zu ändern oder zu löschen. Dazu den Unterbefehl `~h` verwenden, wie im Abschnitt "Kopffelder editieren" auf Seite 7-28 beschrieben.

Das Betreffeld (Subject:) einstellen oder zurücksetzen

Den Unterbefehl `~s` verwenden, um das Feld `Subject:` mit einem bestimmten Begriff oder Satz zu belegen. Mit diesem Unterbefehl wird der vorherige Inhalt des Feldes `Subject:` (falls vorhanden) ersetzt. Das Feld `Subject:` enthält zum Beispiel folgenden Begriff:

```
Subject: Urlaub
```

Zum Ändern des Feldes folgendes eingeben:

```
~s Segelausflug
```

Dadurch wird das Feld `Subject:` geändert in:

```
Subject: Segelausflug
```

Mit diesem Unterbefehl kann nichts an das Feld `Subject:` angehängt werden. Dazu den Unterbefehl `~h` verwenden, wie im Abschnitt "Kopffelder editieren" auf Seite 7-28 beschrieben.

Den Kopienverteiler (Cc:) erweitern

Den Unterbefehl `~c` verwenden, um eine oder mehrere Adressen der Liste `Cc:` hinzuzufügen. Die Liste `Cc:` enthält zum Beispiel für eine Nachricht folgende Adressen:

```
Cc: mark@austin amy
```

Um diese Liste zu erweitern, folgendes eingeben:

```
~c geo@austin mel@gtwn
```

Dadurch wird die Liste `Cc:` geändert in:

```
Cc: mark@austin amy geo@austin mel@gtwn
```

Der Unterbefehl `~c` kann nicht verwendet werden, um den Inhalt der Liste `Cc:` zu ändern oder zu löschen. Dazu den Unterbefehl `~h` verwenden, wie im Abschnitt "Kopffelder editieren" auf Seite 7-28 beschrieben.

Die Liste Bcc: erweitern

Den Unterbefehl **-b** verwenden, um eine oder mehrere Adressen der Liste **Bcc:** hinzuzufügen. Die Liste **Bcc:** enthält für eine Nachricht zum Beispiel die folgende Adresse:

```
Bcc: mark@austin
```

Um diese Liste zu erweitern, folgendes eingeben:

```
~b geo@austin mel@gtwn
```

Dadurch wird die Liste **Bcc:** geändert in:

```
Bcc: mark@austin geo@austin mel@gtwn
```

Der Unterbefehl **-b** kann nicht verwendet werden, um den Inhalt der Liste **Bcc:** zu ändern oder zu löschen. Dazu den Unterbefehl **-h** verwenden, wie im Abschnitt "Kopffelder editieren" auf Seite 7-28 beschrieben.

Empfohlene Dokumentation

Grundlegende Informationen

"Der Editor für elektronische Post" auf Seite 7-10.

Zugehörige Informationen

"Texteditor zur Eingabe von Nachrichten wechseln" auf Seite 7-54.

Rechtschreibprüfung beim Editor für elektronische Post

Voraussetzungen

1. Das Postprogramm muß auf dem System installiert sein.
2. Die Programme zur Textformatierung müssen auf dem System installiert sein.

Erforderliche Schritte

Um den Befehl **spell** verwenden zu können, der die Nachricht auf Rechtschreibfehler überprüft:

1. Die Nachricht in eine temporäre Datei schreiben. Um z. B. die Nachricht in die Datei `checkit` zu schreiben, folgendes eingeben:

```
~w checkit
```

2. Den Befehl **spell** ausführen und dabei die temporäre Datei als Eingabedatei verwenden. Folgendes eingeben:

```
~! spell checkit
```

Der Befehl **spell** erstellt eine Liste mit Wörtern, die nicht in der Liste der bekannten Wörter enthalten sind, gefolgt von einem ! (Ausrufezeichen), das anzeigt, daß zum Postprogramm zurückgekehrt wurde.

3. Die Liste der Wörter prüfen. Entscheiden, ob zum Einbau von Korrekturen ein Editor benötigt wird.
4. Die temporäre Datei mit folgendem Befehl löschen:

```
~! rm checkit
```

Empfohlene Dokumentation

Grundlegende Informationen

"Der Editor für elektronische Post" auf Seite 7-10.

Zugehörige Informationen

"Texteditor zur Eingabe von Nachrichten wechseln" auf Seite 7-54.

Eine Nachricht im Editor für elektronische Post neu formatieren

Voraussetzungen

1. Das Postprogramm muß auf dem System installiert sein.
2. Der Befehl **fmt** muß auf dem System installiert sein.

Erforderliche Schritte

Nach der Eingabe und vor dem Senden kann die Nachricht neu formatiert werden, um das Erscheinungsbild zu verbessern. Dazu wird das Shell-Programm **fmt** verwendet. Um die Nachricht neu zu formatieren, folgenden Unterbefehl in eine neue Zeile eingeben:

```
~| fmt
```

Dieser Unterbefehl verwendet den Befehl **fmt**, um das Erscheinungsbild der Nachricht zu ändern, indem die Informationen für jeden Absatz mit definierten Rändern neu angeordnet werden (die Absätze müssen jeweils durch eine Leerzeile getrennt sein).

Achtung: Den Befehl **fmt** nicht verwenden, wenn die Nachricht eingebettete Nachrichten enthält oder vorformatierte Daten aus externen Dateien. Der Befehl **fmt** formatiert die Kopffelder der eingebetteten Nachrichten neu und ändert möglicherweise das Format der vorformatierten Daten. In diesem Fall den Unterbefehl **-e** oder **-v** verwenden, um einen Gesamtanzeigeditor aufzurufen und die Nachricht neu zu formatieren.

Empfohlene Dokumentation

Grundlegende Informationen

"Der Editor für elektronische Post" auf Seite 7-10.

Zugehörige Informationen

"Texteditor zur Eingabe von Nachrichten wechseln" auf Seite 7-54.

Post empfangen und lesen

Voraussetzungen

1. Das Postprogramm muß auf dem System installiert sein.
2. Im elektronischen Systembriefkasten muß Post sein.

Erforderliche Schritte

Mit den folgenden Schritten kann der elektronische Briefkasten überprüft und die Post angezeigt werden.

Den elektronischen Briefkasten überprüfen

In der Befehlszeile folgendes eingeben:

```
mail
```

Ist im elektronischen Briefkasten keine Post, gibt das System folgende Nachricht aus:

```
Keine Post für Benutzername
```

Ist im elektronischen Briefkasten Post, zeigt das System eine Liste der Nachrichten im elektronischen Systembriefkasten an.

Post anzeigen

Folgende Unterbefehle verwenden, um die Post anzeigen zu lassen:

Die nächste Nachricht im elektronischen Briefkasten ansehen:

An der Eingabeaufforderung des elektronischen Briefkastens folgendes eingeben:

```
next
```

Der Text der nächsten Nachricht wird angezeigt. Sie wird damit zur aktuellen Nachricht. Auch durch Drücken der Eingabetaste kann die nächste Nachricht aufgerufen werden.

Den Inhalt des persönlichen elektronischen Briefkastens ansehen:

An der Eingabeaufforderung der Befehlszeile folgendes eingeben:

```
mail -f
```

Eine Liste der Nachrichten im persönlichen elektronischen Briefkasten **\$HOME/mbox** wird angezeigt.

Den Inhalt eines bestimmten Postordners ansehen:

An der Eingabeaufforderung der Befehlszeile folgendes eingeben:

```
mail -f +dept
```

Bei diesem Beispiel wird eine Liste der Nachrichten im Ordner `dept` angezeigt.

Den Text einer bestimmten Nachricht anzeigen:

Den Unterbefehl `t` verwenden:

```
t 3
```

Der Text der Nachricht 3 wird angezeigt.

Mehrere Nachrichten im elektronischen Briefkasten ansehen:

An der Eingabeaufforderung des elektronischen Briefkastens folgendes eingeben:

```
type 2 3 4
```

bzw.

```
type 2-4
```

Bei diesem Beispiel wird der Text der Nachrichten 2, 3 und 4 angezeigt. Sowohl der Unterbefehl `type` als auch `print` zeigt den Text der Nachrichten an. Soll mehr als eine Nachricht gleichzeitig angezeigt werden, sicherstellen, daß die Anweisung `set crt` in der Datei `$HOME/.mailrc` enthalten ist. Dieser Unterbefehl kann auch an der Eingabeaufforderung des elektronischen Briefkastens eingegeben werden. Wird dieser Unterbefehl nicht verwendet und liegt dennoch mehr als eine Anzeige mit Nachrichten vor, werden die angezeigten Nachrichten nach oben aus der Anzeige verschoben, bevor der Benutzer sie lesen kann.

Die aktuelle Nachricht löschen und die nächste anzeigen:

An der Eingabeaufforderung des elektronischen Briefkastens folgendes eingeben:

```
dp
```

Die aktuelle Nachricht wird gelöscht und die nächste wird angezeigt. Eine bestimmte Nachricht kann auch gelöscht werden, indem `delete` eingegeben wird, gefolgt von der Nachrichtennummer.

Hilfetext abrufen:

An der Eingabeaufforderung des elektronischen Briefkastens folgendes eingeben:

```
?
```

Der Unterbefehl `?` zeigt eine Zusammenfassung der häufigsten Unterbefehle des elektronischen Briefkastens an. Eine Liste aller Unterbefehle des elektronischen Briefkastens (ohne Erläuterungen) kann auch durch Eingabe des Unterbefehls `!` an der Eingabeaufforderung des elektronischen Briefkastens aufgerufen werden.

Den elektronischen Briefkasten verlassen und zum Betriebssystem zurückkehren:

Um den elektronischen Briefkasten zu verlassen und zum Betriebssystem zurückzukehren, können die folgenden zwei Unterbefehle verwendet werden:

- Den Unterbefehl `q` an der Eingabeaufforderung des elektronischen Briefkastens eingeben. Beim Verlassen des elektronischen Briefkastens werden alle Nachrichten, die zum Löschen markiert sind, aus dem Briefkasten entfernt und können nicht wiederhergestellt werden. Das Postprogramm speichert die Nachrichten, die bereits gelesen, aber weder gelöscht noch gespeichert wurden, im persönlichen elektronischen Briefkasten (`mbox`). Wurden keine Nachrichten der elektronischen Post gelesen, bleiben sie bis zur Bearbeitung im elektronischen Systembriefkasten.
- Den Unterbefehl `x` an der Eingabeaufforderung des elektronischen Briefkastens eingeben, um den Briefkasten zu verlassen und zum Betriebssystem zurückzukehren, *ohne den ursprünglichen Inhalt des elektronischen Briefkastens zu ändern*. Das Programm ignoriert Anforderungen zum Löschen von Nachrichten.

Empfohlene Dokumentation

Grundlegende Informationen

"Post empfangen und bearbeiten" auf Seite 7-13.

"Unterbefehle des elektronischen Briefkastens im Postprogramm" auf Seite 7-15.

Post beantworten

Voraussetzungen

1. Das Postprogramm muß auf dem System installiert sein.
2. Im elektronischen Systembriefkasten muß Post sein.

Erforderliche Schritte

Die folgenden Schritte ermöglichen das Beantworten von Post.

Eine Nachricht der elektronischen Post beantworten

An der Eingabeaufforderung des elektronischen Briefkastens (&) folgendes eingeben:

r

Der Unterbefehl r (reply = antworten) erstellt eine neue Nachricht, die an den Absender der ausgewählten Nachricht adressiert ist und für alle Benutzer in der Kopieempfängerliste (falls vorhanden) kopiert wird. Im Feld `Subject`: der neuen Nachricht wird auf die ausgewählte Nachricht verwiesen. Der Standardwert des Unterbefehls r ist die aktuelle Nachricht. Dieser Standardwert kann überschrieben werden, indem die Nachrichtennummer nach dem r eingegeben wird.

Nur dem Absender der Nachricht antworten

An der Eingabeaufforderung des elektronischen Briefkastens den Unterbefehl R (Reply = Antworten) eingeben:

R

Damit wird eine Antwort an den Absender der Nachricht erstellt. Der Standardwert des Unterbefehls R ist die aktuelle Nachricht. Dieser Standardwert kann überschrieben werden, indem die Nachrichtennummer nach dem R eingegeben wird. Zum Beispiel startet:

R 4

eine Antwort auf Nachricht 4. Die Systemantwort sieht ähnlich aus wie die folgende Nachricht:

To: karen@thor
Subject: Re: Abteilungsmeeting

Daraufhin kann eine Antwort eingegeben werden:

Ich werde teilnehmen.

Ist die Eingabe beendet, entweder einen . (Punkt) eingeben oder die Tastenfolge Strg-D drücken, um die Nachricht zu senden. Ist die Antwort gesendet, wird zur Eingabeaufforderung des elektronischen Briefkastens zurückgekehrt.

Neue Nachrichten erstellen innerhalb des elektronischen Briefkastens

Den Unterbefehl m eingeben. Um z. B. eine Nachricht an nita@beatrice zu senden, an der Eingabeaufforderung des elektronischen Briefkastens folgendes eingeben:

m nita@beatrice

Der Parameter *Adresse* kann jede gültige Benutzeradresse sein. Dieser Befehl startet den Editor für elektronische Post und ermöglicht das Erstellen einer neuen Nachricht.

Empfohlene Dokumentation

Grundlegende Informationen

"Post erstellen und senden" auf Seite 7-3.

"Post empfangen und bearbeiten" auf Seite 7-13.

Den Inhalt eines elektronischen Briefkastens anzeigen

Voraussetzungen

1. Das Postprogramm muß auf dem System installiert sein.
2. Im elektronischen Systembriefkasten muß Post sein.

Erforderliche Schritte

Mit den folgenden Schritten kann der Inhalt des elektronischen Briefkastens angezeigt werden.

Den Inhalt des elektronischen Briefkastens anzeigen

An der Eingabeaufforderung des elektronischen Briefkastens (&) den Unterbefehl **h** eingeben:

& h

Etwa 20 Nachrichten können gleichzeitig angezeigt werden. Die tatsächliche Anzahl hängt vom Typ der verwendeten Datenstation ab und vom Befehl **set screen**.

Bei mehr als 20 Nachrichten im elektronischen Briefkasten

An der Eingabeaufforderung des elektronischen Briefkastens (&) folgendes eingeben:

& h 21

Die Nachricht 21 und nachfolgende, bis zu Nachricht 40 (wenn so viele Nachrichten im Briefkasten vorhanden sind), werden angezeigt. So lange den Unterbefehl **h** eingeben, bis alle Nachrichten angezeigt wurden.

Zur Gruppe der ersten 20 Nachrichten zurückkehren

An der Eingabeaufforderung des elektronischen Briefkastens (&) folgendes eingeben:

& h 1

Die Gruppe der ersten 20 Nachrichten wird angezeigt.

Empfohlene Dokumentation

Grundlegende Informationen

"Post empfangen und bearbeiten" auf Seite 7-13.

"Unterbefehle des elektronischen Briefkastens im Postprogramm" auf Seite 7-15.

Zugehörige Informationen

"Die Nachrichtenanzeige im Postprogramm ändern" auf Seite 7-19.

Informationen zu ausgewählten Nachrichten anzeigen

Voraussetzungen

1. Das Postprogramm muß auf dem System installiert sein.
2. Im elektronischen Systembriefkasten muß Post sein.

Erforderliche Schritte

Mit den folgenden Schritten können Informationen zu ausgewählten Nachrichten angezeigt werden.

Die Kopffelder von bestimmten Nachrichten anzeigen

Um die Kopffelder der Nachrichten 1, 4 und 7 anzuzeigen, an der Eingabeaufforderung des elektronischen Briefkastens (&) folgendes eingeben:

f 1 4 7

Die Kopffelder der Nachrichten 1, 4 und 7 werden angezeigt.

Die Kopffelder von einem Bereich von Nachrichten anzeigen

Um die Kopffelder der Nachrichten 1 bis 10 anzuzeigen, an der Eingabeaufforderung des elektronischen Briefkastens (&) folgendes eingeben:

f 1-10

Die Kopffelder der Nachrichten 1 bis 10 werden angezeigt.

Nachrichten eines bestimmten Absenders anzeigen

An der Eingabeaufforderung des elektronischen Briefkastens (&) folgendes eingeben:

& f ron

Nachrichten vom Benutzer ron (falls vorhanden) werden angezeigt. Die für die Adresse eingegebenen Zeichen müssen nicht exakt mit der gesuchten Adresse übereinstimmen. Daher stimmt die Anforderung nach der Adresse ron (in Groß- oder Kleinbuchstaben) mit allen folgenden Adressen überein:

- RoN
- ron@topdog
- hron
- rOn.

Nachricht mit einem bestimmten Betrefftext anzeigen

An der Eingabeaufforderung des elektronischen Briefkastens (&) folgendes eingeben:

& f /treff

Nachrichten (falls vorhanden), deren Feld Subject: die Buchstaben treff enthält, werden angezeigt. Die eingegebenen Zeichen für das Suchmuster müssen nicht exakt mit dem Feld Subject: übereinstimmen. Sie müssen nur in Groß- oder Kleinbuchstaben im Feld Subject: enthalten sein. Daher stimmt die Anforderung nach dem Betrefftext treff mit allen folgenden Betrefftexten überein:

- Treffen am Donnerstag
- Entscheidungen treffen
- Treffpunkt Kantine.

Empfohlene Dokumentation

Grundlegende Informationen

"Post empfangen und bearbeiten" auf Seite 7-13.

"Unterbefehle des elektronischen Briefkastens im Postprogramm" auf Seite 7-15.

Eine Nachricht im elektronischen Briefkasten editieren

Voraussetzungen

1. Das Postprogramm muß auf dem System installiert sein.
2. Der Befehl **mail** muß abgesetzt sein, um Post im elektronischen Systembriefkasten, in einem anderen elektronischen Briefkasten oder in einem Ordner zu lesen.

Erforderliche Schritte

Um einer Nachricht im elektronischen Briefkasten Daten hinzuzufügen, den Unterbefehl **e** oder **v** eingeben, gefolgt von der Nachrichtennummer. Um z. B. der Nachricht 13 mit Hilfe des Editors **e** (oder dem jeweils in der Datei **.mailrc** definierten Editor) eine Notiz hinzuzufügen, folgendes eingeben:

e 13

Wird keine Nachrichtennummer angegeben, aktiviert der Befehl den Editor unter Verwendung der aktuellen Nachricht. Beim Verlassen des Editors wird zur Eingabeaufforderung des elektronischen Briefkastens zurückgekehrt, um weitere Nachrichten im Briefkasten zu verarbeiten.

Empfohlene Dokumentation

Grundlegende Informationen

"Post empfangen und bearbeiten" auf Seite 7-13.

"Unterbefehle des elektronischen Briefkastens im Postprogramm" auf Seite 7-15.

Zugehörige Informationen

"Texteditor zur Eingabe von Nachrichten wechseln" auf Seite 7-54.

Post speichern

Voraussetzungen

Das Postprogramm muß auf dem System installiert sein.

Erforderliche Schritte

Post aus dem elektronischen Systembriefkasten, dem persönlichen elektronischen Briefkasten, einer Datei oder einem Ordner kann in jeder anderen Datei oder jedem anderen Ordner gespeichert werden. Der Inhalt einer Nachricht kann zu einer Datei oder einem Ordner hinzugefügt werden, indem der Befehl **s** oder **w** verwendet wird. Beide Befehle fügen Daten an das Ende einer vorhandenen Datei an oder erstellen eine neue Datei. Die in der Datei schon vorhandenen Daten werden nicht zerstört.

Mit dem Unterbefehl **s** eine Nachricht mit Kopf speichern

Um eine Nachricht mit Kopffeldern in einer Datei oder einem Ordner zu speichern, den Unterbefehl **s** verwenden. Das Format für den Unterbefehl **s** ist folgendes:

s *Nachrichtenliste* *Dateiname*

Um beispielsweise die Nachrichten 1, 2, 3 und 4 mit ihren Kopffeldern in der Datei *notes* im aktuellen Verzeichnis zu speichern, folgendes eingeben:

```
s 1-4 notes
```

Das Postprogramm gibt folgende Nachricht aus:

```
"notes" [Angefügt] 62/1610
```

Um die gesamte Nachricht 6 im vorhandenen Ordner *admin* im Ordnerverzeichnis zu speichern, folgendes eingeben:

```
s 6 +admin
```

Wurde das Ordnerverzeichnis als */home/george/letters* in der Datei *.mailrc* definiert, gibt das System folgende Nachricht aus:

```
"/home/george/letters/admin" [Neue Datei] 14/321
```

Mit dem Unterbefehl **w** eine Nachricht ohne Kopf speichern

Um eine Nachricht ohne Kopffelder in einer Datei oder einem Ordner zu speichern, den Unterbefehl **w** verwenden. Das Format für den Unterbefehl **w** ist folgendes:

w *Nachrichtenliste* *Dateiname*

Um nur den Text der Nachricht 6 in der Datei *pass* im aktuellen Verzeichnis zu speichern, folgendes eingeben:

```
w 6 pass
```

Besteht die Datei *pass* noch nicht, gibt das System folgende Nachricht aus:

```
"pass" [Neue Datei] 12/30
```

Empfohlene Dokumentation

Grundlegende Informationen

"Post empfangen und bearbeiten" auf Seite 7-13.

Zugehörige Informationen

"Unterbefehle des elektronischen Briefkastens im Postprogramm" auf Seite 7-15.

"Das Postprogramm anpassen" auf Seite 7-48.

Ordner im elektronischen Briefkasten verwenden

Voraussetzungen

1. Das Postprogramm muß auf dem System installiert sein.
2. Um die Kurzform + (Pluszeichen) zum Umschalten zwischen Ordnern verwenden zu können, muß ein Ordnerverzeichnis erstellt werden und der Eintrag `set folder` in der Datei `.mailrc` angelegt sein.

Erforderliche Schritte

Den Befehl **mail** verwenden, um den elektronischen Systembriefkasten (Standardwert), den persönlichen elektronischen Briefkasten oder einen definierten Ordner anzusehen (siehe "Post empfangen und lesen" auf Seite 7-33). Vom elektronischen Briefkasten aus stehen folgende Optionen zur Verfügung:

- Den Namen des aktuellen elektronischen Briefkastens oder Ordners suchen.
- Zu einem anderen elektronischen Briefkasten oder Ordner wechseln.

Den Namen des aktuellen elektronischen Briefkastens suchen

Um den Namen des aktuellen Briefkastens zu suchen, den Unterbefehl **folder** ohne Parameter eingeben:

```
folder
```

Ist der aktuelle Briefkasten `/home/lance/mbox`, wird folgendes angezeigt:

```
/home/lance/mbox: 2 Nachrichten 1 gesichert
```

Diese Nachricht zeigt an, welches der aktuelle elektronische Briefkasten ist, daß er zwei Nachrichten enthält, von denen eine beim Verlassen des elektronischen Briefkastens gelöscht werden wird. Auch wenn nach Eingabe des Befehls **mail** der Name des aktuellen elektronischen Briefkastens angezeigt wird, kann er möglicherweise während der Arbeit wieder in Vergessenheit geraten.

Zu einem anderen elektronischen Briefkasten wechseln

Hinweis: Jede Nachricht, die zum Löschen markiert ist, wird beim Verlassen des jeweiligen elektronischen Briefkastens gelöscht. Gelöschte Nachrichten können nicht wiederhergestellt werden.

Ist das Postprogramm einmal mit einem elektronischen Briefkasten gestartet, den Unterbefehl **file** oder **folder** verwenden, um zu einem anderen elektronischen Briefkasten zu wechseln. Um beispielsweise von der Datei `mbox` zum Ordner `project` zu wechseln, folgendes eingeben:

```
folder +project
```

Wurden alle Nachrichten in der Datei `mbox` gelöscht, zeigt das Postprogramm folgende Nachricht an:

```
/home/dee/mbox gelöscht  
+project: 2 Nachrichten 2 neu
```

Darunter folgt eine Liste der Nachrichten im Ordner `project`.

Empfohlene Dokumentation

Grundlegende Informationen

"Post empfangen und bearbeiten" auf Seite 7-13.

"Unterbefehle des elektronischen Briefkastens im Postprogramm" auf Seite 7-15.

Zugehörige Informationen

"Das Postprogramm anpassen" auf Seite 7-48.

Post weiterleiten

Voraussetzungen

Das Postprogramm muß auf dem System installiert sein.

Erforderliche Schritte

Um Post an einen anderen Benutzer weiterzuleiten, folgende Schritte ausführen:

1. Den Befehl **cd** ohne Parameter eingeben, um sicherzustellen, daß man sich im Benutzerverzeichnis befindet. Für die Benutzer-ID beispielsweise folgendes eingeben:

```
$ cd
$ pwd
```

Das System gibt folgende Antwort aus:

```
/home/mary
```

2. Vom Benutzerverzeichnis aus die Datei **.forward** erstellen. Die Datei **.forward** enthält die Netzadresse oder -adressen der Benutzer, die die weitergeleitete Netzpost empfangen sollen. Es dürfen nur gültige Adressen enthalten sein. Handelt es sich dabei um eine Leerdatei (ohne Inhalt), wird die Post nicht weitergeleitet, sondern im elektronischen Briefkasten gespeichert. Mit dem Befehl **cat** kann die Datei **.forward** wie folgt erstellt werden:

```
$ cat > .forward
mark
joe@sandwich
[END OF FILE]
$
```

Hinweis: [END OF FILE] stellt das Dateiendezeichen dar (an den meisten Datenstationen die Tastenfolge Strg-D). Es muß gesondert in einer Zeile stehen.

Jetzt enthält die Datei **.forward** die Adressen von Benutzern, an die Post weitergeleitet werden soll. Die Post wird an **mark** auf dem lokalen System und an **joe** auf dem System **sandwich** weitergeleitet.

3. Um das Weiterleiten der Post zu stoppen, unter Verwendung des Befehls **rm** die Datei **.forward** aus dem Benutzerverzeichnis entfernen:

```
$ rm .forward
```

Die Datei **.forward** wird mit dem Befehl **rm** gelöscht. Eingehende Post wird nicht länger an andere Benutzer weitergeleitet.

Empfohlene Dokumentation

Grundlegende Informationen

"Post erstellen und senden" auf Seite 7-3.

"Post empfangen und bearbeiten" auf Seite 7-13.

Zugehörige Informationen

"Verwendung von Unterbefehlen zum Senden von Post" auf Seite 7-25.

"Urlaubsnachricht senden" auf Seite 7-46.

Urlaubsnachricht senden

Voraussetzungen

1. Das Postprogramm muß auf dem System installiert sein.
2. Eine Datei **.forward** muß im Anmeldeverzeichnis **\$HOME** bestehen und die Benutzer-ID und Adresse enthalten.

Erforderliche Schritte

1. Um die Urlaubsnachricht zu initialisieren, im Anmeldeverzeichnis **\$HOME** folgendes eingeben:

```
vacation -I
```

Dadurch werden die Dateien **.vacation.dir** und **.vacation.pag** erstellt.

2. Die folgende Zeile in der Datei **.forward** für die Benutzer-ID **carl** hinzufügen:

```
carl, "|vacation carl"
```

3. Damit jede Person, die Post an den Benutzer sendet, eine Nachricht erhält, die Datei **\$HOME/.vacation.msg** erstellen und eine entsprechende Nachricht hinzufügen. Es folgt ein Beispiel für diese Datei:

```
From: carl@odin.austin (Carl Jones)  
Subject: Urlaub
```

```
Ich bin bis Oktober in urlaub. In dringenden Fällen  
Jim Terry <terry@zeus.valhalla> kontaktieren.  
--carl
```

Diese Nachricht erhält jeder, der an den Benutzer Post sendet, wobei die Urlaubsnachricht nur einmal pro Woche ausgegeben wird, unabhängig davon, wie oft der Absender Nachrichten an den Benutzer sendet. Wird die Post an jemanden weitergeleitet, wird die Nachricht des Absenders an die Person gesendet, die in der Datei **.forward** definiert ist.

4. Um die Urlaubsnachricht abubrechen, die Dateien **.forward**, **.vacation.dir**, **.vacation.pag** und **.vacation.msg** aus dem Anmeldeverzeichnis **\$HOME** entfernen:

```
rm .forward .vacation.dir .vacation.pag .vacation.msg
```

Empfohlene Dokumentation

Grundlegende Informationen

- "Post erstellen und senden" auf Seite 7-3.
- "Post empfangen und bearbeiten" auf Seite 7-13.

Zugehörige Informationen

- "Verwendung von Unterbefehlen zum Senden von Post" auf Seite 7-25.
- "Post weiterleiten" auf Seite 7-45.

Geheime Post senden und empfangen

Voraussetzungen

1. Das Postprogramm muß auf dem System installiert sein.
2. Ein Kennwort muß über den Befehl **enroll** definiert sein.

Erforderliche Schritte

Geheime Post senden

Um geheime Post an den Benutzer Barbara zu senden, folgendes eingeben:

```
xsend barbara
```

Mit dem Postprogramm den Text der Nachricht eingeben. Ist die Eingabe der Nachricht beendet, die Tastenfolge Strg-D drücken oder einen Punkt (.) eingeben, um den Editor für elektronische Post zu verlassen und die Nachricht zu senden. Durch den Befehl **xsend** wird die Nachricht vor dem Senden verschlüsselt.

Geheime Post empfangen

Folgendes eingeben:

```
xget
```

Der Benutzer wird zur Eingabe des Kennwortes aufgefordert, das vorher über den Befehl **enroll** definiert worden ist. Nach Eingabe des Kennworts wird die Eingabeaufforderung für den Befehl **xget** angezeigt (&) und eine Liste der geheimen Post. Das Postprogramm wird verwendet, um die gesamte geheime Post anzuzeigen.

Empfohlene Dokumentation

Grundlegende Informationen

"Post erstellen und senden" auf Seite 7-3.

"Post empfangen und bearbeiten" auf Seite 7-13.

Zugehörige Informationen

"Unterbefehle des elektronischen Briefkastens im Postprogramm" auf Seite 7-15.

"Post empfangen und lesen" auf Seite 7-33.

Das Postprogramm anpassen

Voraussetzungen

Das Postprogramm muß auf dem System installiert sein.

Erforderliche Schritte

Mit den folgenden Schritten kann das Postprogramm angepaßt werden.

Die aktivierten Postoptionen ansehen

An der Eingabeaufforderung des elektronischen Briefkastens (&) folgendes eingeben:

```
set
```

Eine Nachricht, ähnlich wie die folgende, wird angezeigt:

```
ask
metoo
toplines 10
&
```

Bei diesem Beispiel sind zwei binäre Optionen aktiviert: **ask** und **metoo**. Der Eintrag **askcc** ist in der Liste nicht enthalten, d. h., die Option **askcc** ist nicht aktiviert. Der Option **toplines** wurde der Wert 10 zugeordnet.

Die Kopffelder Date, From und To unterdrücken

Um zu verhindern, daß die Kopffelder **date**, **from** und **to** angezeigt werden, wenn eine Nachricht mit dem Unterbefehl **type** oder **print** zum Lesen aufgerufen wird, in der Datei **\$HOME/.mailrc** folgendes eingeben:

```
ignore date from to
```

Wird eine Nachricht über den Unterbefehl **type** oder **print** angezeigt, werden die Kopffelder **date**, **from** und **to** nicht angezeigt. Sollen diese Felder dennoch angezeigt werden, ohne jedoch die Anweisung **ignore** aus der Datei **\$HOME/.mailrc** zu löschen, den Unterbefehl **Type**, **Print** oder **top** verwenden, um die Nachricht aufzurufen.

Eine Liste der momentan ignorierten Kopffelder abrufen

An der Eingabeaufforderung des elektronischen Briefkastens (&) folgendes eingeben:

```
ignore
```

Eine Liste der momentan ignorierten Kopffelder wird angezeigt. Zum Beispiel:

```
ask askcc
```

Die Namensanzeige unterdrücken

Die Namensanzeige ist die oberste Zeile in der Liste der Nachrichten, die den Namen des Postprogramms zeigt. Es ähnelt der folgenden Zeile:

```
Mail [5.2 UCB] [Workstation 3.1] Mit ? kann Hilfetext abgerufen werden.
```

Um zu verhindern, daß die Namensanzeige beim Starten des Postprogramms erscheint, in der Datei **\$HOME/.mailrc** folgendes eingeben:

```
set quiet
```


Die Anzahl der angezeigten Zeilen in der Nachrichtenliste ändern

Um die Anzahl der Zeilen zu ändern, die von der Nachrichtenliste gleichzeitig angezeigt werden, in der Datei **\$HOME/.mailrc** folgendes eingeben:

```
set screen=20
```

Bei diesem Beispiel wird das System 20 Nachrichtenkopfzeilen gleichzeitig anzeigen. Entweder den Unterbefehl **headers** oder **z** verwenden, um weitere Gruppen von Kopfzeilen anzusehen. Dieser Unterbefehl kann auch an der Eingabeaufforderung des elektronischen Briefkastens eingegeben werden.

Gesendete Nachrichten aufzeichnen

Die folgenden Anweisungen in der Datei **\$HOME/.mailrc** eingeben:

```
set record=letters/mailout  
set folder=letters
```

Neue Ordner können während einer Postsitzung erstellt werden; das Verzeichnis, in dem sie gespeichert werden, muß allerdings vor der Definition neuer Ordner bereits bestehen. Bei diesem Beispiel muß der Pfad **\$HOME/letters/mailout** erstellt sein, bevor Nachrichten gespeichert werden können.

Kopien der gesendeten Nachrichten lesen

Folgendes an der Eingabeaufforderung eingeben:

```
mail -f +mailout
```

Bei diesem Beispiel enthält die Datei `mailout` Kopien der Nachrichten, die an andere Benutzer gesendet wurden.

Empfohlene Dokumentation

Grundlegende Informationen

"Elektronische Briefkästen und Postordner verwenden" auf Seite 7-14.

"Das Postprogramm anpassen" auf Seite 7-18.

Alias-Namen oder Verteiler erstellen

Voraussetzungen

- Das Postprogramm muß auf dem System installiert sein.
- Die Namen und Adressen der Benutzer, die in den Alias-Namen oder im Verteiler aufgenommen werden sollen, müssen bekannt sein.

Erforderliche Schritte

Mit den folgenden Schritten kann ein Alias-Name oder ein Verteiler erstellt werden.

Einen Alias-Namen erstellen

Die folgende Anweisung in der Datei **\$HOME/.mailrc** eingeben:

```
alias kath kathleen@ gtw
```

Bei diesem Beispiel wurde der Alias-Name **kath** für den Benutzer **kathleen** in **gtw** erstellt. Nachdem diese Zeile in der Datei **\$HOME/.mailrc** hinzugefügt wurde, kann mit folgender Eingabe an der Eingabeaufforderung der Befehlszeile eine Nachricht an Kathleen gesendet werden:

```
mail kath
```

Unter Verwendung dieses Alias-Namens kann nun Post an Kathleen gesendet werden.

Einen Verteiler für die Abteilung erstellen

Die folgende Anweisung in der Datei **\$HOME/.mailrc** eingeben:

```
abt dee@merlin anne@anchor jerry@zeus bill carl
```

Nachdem diese Zeile in der Datei **\$HOME/.mailrc** hinzugefügt wurde, kann mit folgender Eingabe an der Eingabeaufforderung der Befehlszeile eine Nachricht an die Abteilung gesendet werden:

```
mail abt
```

Die nun erstellte Nachricht wird an den Benutzer **dee** auf dem System **merlin**, an **anne** auf dem System **anchor**, an **jerry** auf dem System **zeus** und an **bill** und **carl** auf dem lokalen System gesendet.

Die erstellten Alias-Namen und Verteiler auflisten

An der Eingabeaufforderung des elektronischen Briefkastens (&) folgendes eingeben:

```
alias
```

Eine Liste der Alias-Namen und Verteiler wird angezeigt.

Empfohlene Dokumentation

Grundlegende Informationen

"Post erstellen und senden" auf Seite 7-3.

"Das Postprogramm anpassen" auf Seite 7-18.

Zugehörige Informationen

"Definierte Alias-Namen und Verteiler bestimmen" auf Seite 7-51.

Definierte Alias-Namen und Verteiler bestimmen

Voraussetzungen

Das Postprogramm muß auf dem System installiert sein.

Erforderliche Schritte

Um eine Liste der Alias-Namen aufzurufen, die für die aktuelle Postsitzung definiert wurden, an der Eingabeaufforderung des elektronischen Briefkastens folgendes eingeben:

a

Alle Alias-Namen und Verteiler werden aufgelistet, einer pro Zeile.

Empfohlene Dokumentation

Grundlegende Informationen

"Post erstellen und senden" auf Seite 7-3.

"Das Postprogramm anpassen" auf Seite 7-18.

Zugehörige Informationen

"Alias-Namen oder Verteiler erstellen" auf Seite 7-50.

Bedienerführung für Betrefffeld ändern

Voraussetzungen

Das Postprogramm muß auf dem System installiert sein.

Erforderliche Schritte

Bedienerführung für Betrefffeld aktivieren

Um die Bedienerführung für Betrefffeld zu aktivieren, in der Datei `.mailrc` die Option `ask` folgendermaßen editieren:

```
set ask
```

Die Bedienerführung für Betrefffeld ist aktiviert.

Bedienerführung für Betrefffeld inaktivieren

Um die Bedienerführung für Betrefffeld zu inaktivieren, in der Datei `.mailrc` die Option `ask` folgendermaßen editieren:

```
unset ask
```

Die Bedienerführung für Betrefffeld ist inaktiviert.

Empfohlene Dokumentation

Grundlegende Informationen

"Post erstellen und senden" auf Seite 7-3.

"Das Postprogramm anpassen" auf Seite 7-18.

Zugehörige Informationen

"Kopffelder in einer Nachricht der elektronischen Post ändern und hinzufügen" auf Seite 7-28.

Bedienerführung für Kopienfeld ändern

Voraussetzungen

Das Postprogramm muß auf dem System installiert sein.

Erforderliche Schritte

Bedienerführung für Kopienfeld (Cc:) aktivieren

Um die Bedienerführung für Kopienfeld (Cc:) zu aktivieren, in der Datei `.mailrc` die Option `askcc` folgendermaßen editieren:

```
set askcc
```

Die Bedienerführung für Kopienfeld ist aktiviert.

Bedienerführung für Kopienfeld (Cc:) inaktivieren

Um die Bedienerführung für Kopienfeld (Cc:) zu inaktivieren, in der Datei `.mailrc` die Option `askcc` folgendermaßen editieren:

```
unset askcc
```

Die Bedienerführung für Kopienfeld ist inaktiviert.

Empfohlene Dokumentation

Grundlegende Informationen

"Post erstellen und senden" auf Seite 7-3.

"Das Postprogramm anpassen" auf Seite 7-18.

Zugehörige Informationen

"Kopffelder in einer Nachricht der elektronischen Post ändern und hinzufügen" auf Seite 7-28.

Texteditor zur Eingabe von Nachrichten wechseln

Voraussetzungen

Das Postprogramm muß auf dem System installiert sein.

Erforderliche Schritte

Zum Editor **e** wechseln

Um vom Postprogramm aus zum Editor **e** zu wechseln, folgendes eingeben:

`~e`

Diese Zeichenfolge aktiviert den Editor **e** oder einen anderen in der Datei **.mailrc** definierten Editor. Die Nachricht der elektronischen Post mit diesem Editor bearbeiten.

Zum Editor **vi** wechseln

Um vom Postprogramm aus zum Editor **vi** zu wechseln, folgendes eingeben:

`~v`

Diese Zeichenfolge aktiviert den Editor **vi** oder einen anderen in der Datei **.mailrc** definierten Editor. Die Nachricht der elektronischen Post mit diesem Editor bearbeiten.

Einen Editor definieren

Um den Editor zu definieren, der zur Eingabe von Nachrichten der elektronischen Post verwendet werden soll, in der Datei **.mailrc** folgendes eingeben:

set EDITOR=*Pfadname*

Dieser Befehl in der Datei **.mailrc** definiert den Editor, der mit der Zeichenfolge `~e` aktiviert wird. Der Wert von *Pfadname* muß der vollständige Pfadname zum gewünschten Editor sein.

Außerdem kann folgender Befehl verwendet werden:

set VISUAL=*Pfadname*

Dieser Befehl in der Datei **.mailrc** definiert den Editor, der mit der Zeichenfolge `~v` aktiviert wird. Der Wert von *Pfadname* muß der vollständige Pfadname zum gewünschten Editor sein. Standardwert: **/usr/bin/vi**.

Empfohlene Dokumentation

Grundlegende Informationen

"Post erstellen und senden" auf Seite 7-3.

"Das Postprogramm anpassen" auf Seite 7-18.

Zugehörige Informationen

"Post senden" auf Seite 7-23.

Das Kapitel "Überblick über Editoren" im Handbuch *Editierkonzepte und -prozeduren*.

Die Abschnitte zu den Befehlen **e**, **ed** und **vi**.

Kapitel 8. Online-Informationen und Hilfetext

Dieses Kapitel ist eine Einführung zu InfoExplorer und den neuen Funktionen und es beschreibt, wie die Anwendung in einer Grafikschnittstelle bzw. ASCII-Schnittstelle verwendet wird. Es erläutert weiterhin die erforderlichen Schritte, wie mit Befehlen Online-Informationen gesucht werden können.

Die folgenden Abschnitte enthalten Informationen zu InfoExplorer:

- Ein neuer Zugriff auf Dokumentation
- Neue Funktionen bei InfoExplorer Version 1.2.0
- Verfügbare Hilfetextennungen bei InfoExplorer
- Erste Schritte (InfoExplorer-ASCII-Schnittstelle)
- Erste Schritte (InfoExplorer-Grafikschnittstelle)

Wie mit Befehlen auf Online-Informationen zugegriffen werden kann, wird in folgenden Abschnitten beschrieben:

- Informationen zu Befehlen anzeigen
- Handbuchseiten anzeigen

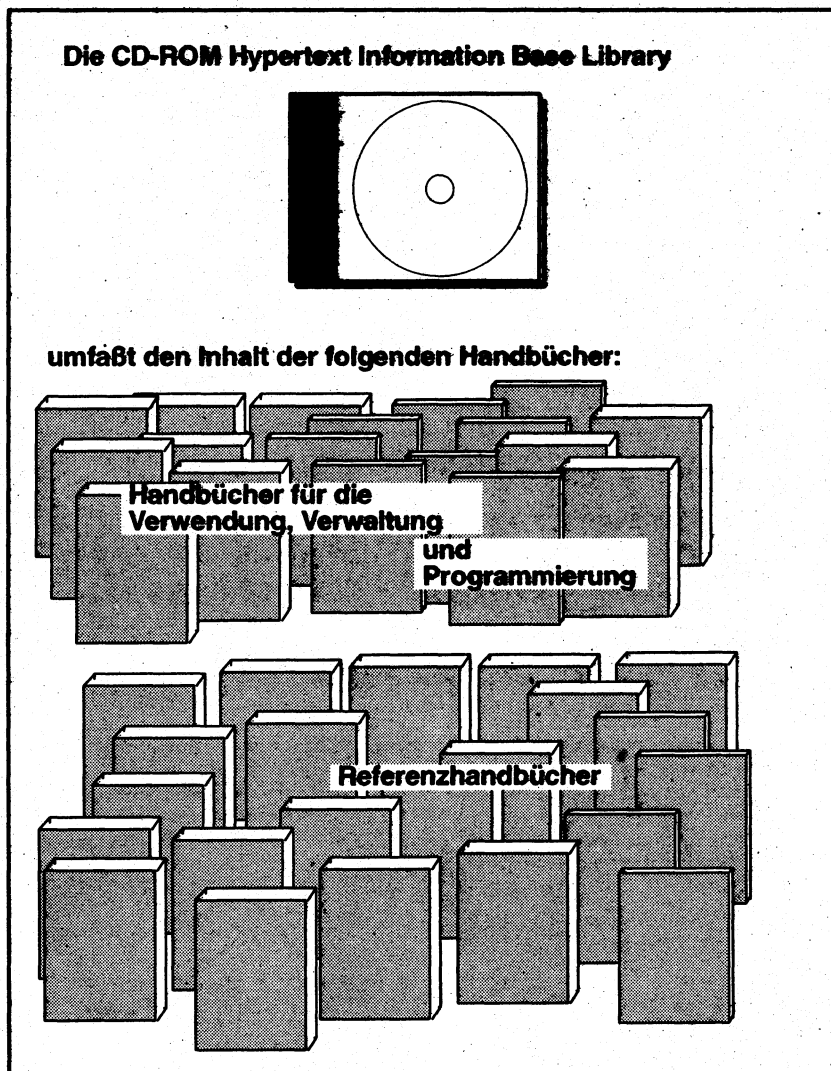
Weitere Informationen zu InfoExplorer enthält das *InfoExplorer Benutzerhandbuch*.

Informationen über "Änderungen in letzter Minute" enthält die Informationsdatei. Auf die Informationsdatei kann von InfoExplorer oder vom Benutzersystem aus zugegriffen werden. Eine Liste aller verfügbaren Informationsdateien wird nach folgender Eingabe angezeigt:

```
find /usr/lpp -name "**README*" -print
```

Ein neuer Zugriff auf Dokumentation

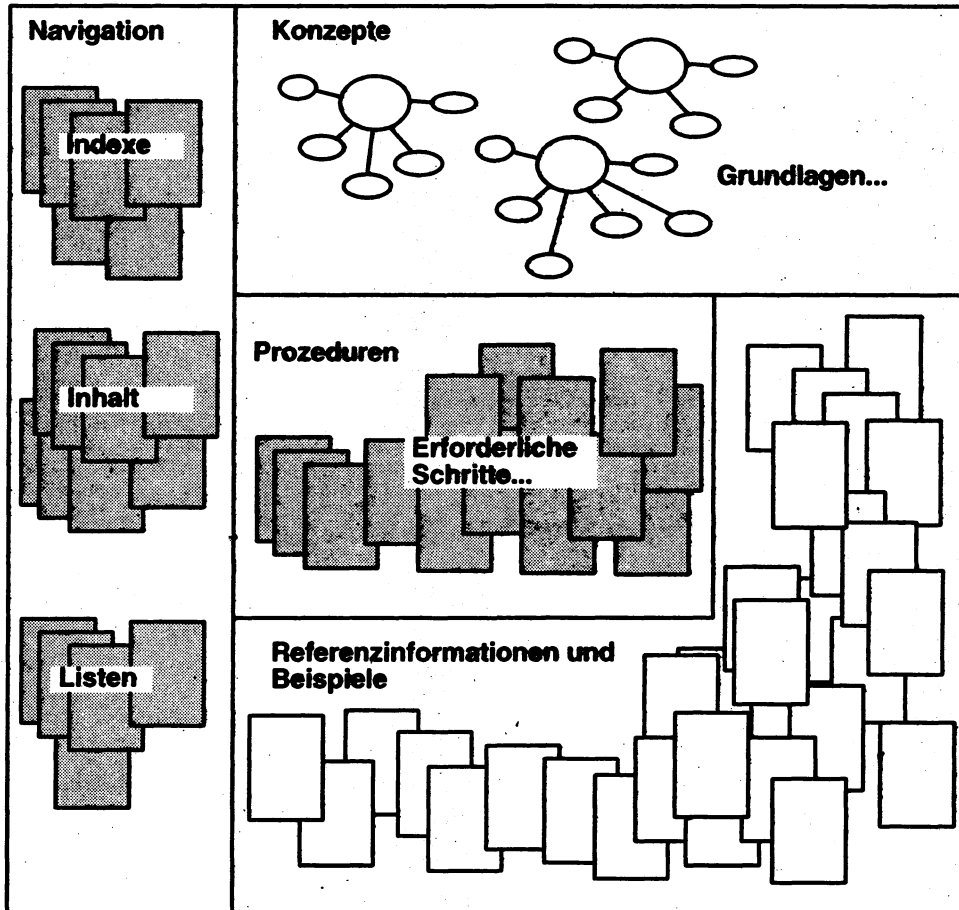
InfoExplorer und die Online-Hypertextdatenbank vereinfachen den Zugriff auf die Systemdokumentation. Die Informationen können auf CD-ROM oder Festplatte vorliegen. Wie in der Abbildung dargestellt wird zur Arbeit mit dem System keine umfangreiche Bibliothek mit Handbüchern benötigt.



In den InfoExplorer-Hypertext-Informationen, die mit dem Betriebssystem geliefert wurden, ist der Text enthalten, der sich in den Softwarehandbüchern zur Verwendung, Verwaltung und Programmierung und den Software-Referenzhandbüchern befindet. Genaue Angaben darüber, welche Informationen sich auf der CD-ROM-Version befinden und welche Informationen automatisch mit dem Betriebssystem und den Lizenzprogrammen geliefert werden, sind dem *Dokumentationsüberblick* zu entnehmen.

Informationen in der Datenbank

Die folgende Abbildung veranschaulicht die in InfoExplorer verfügbaren Informationsarten.



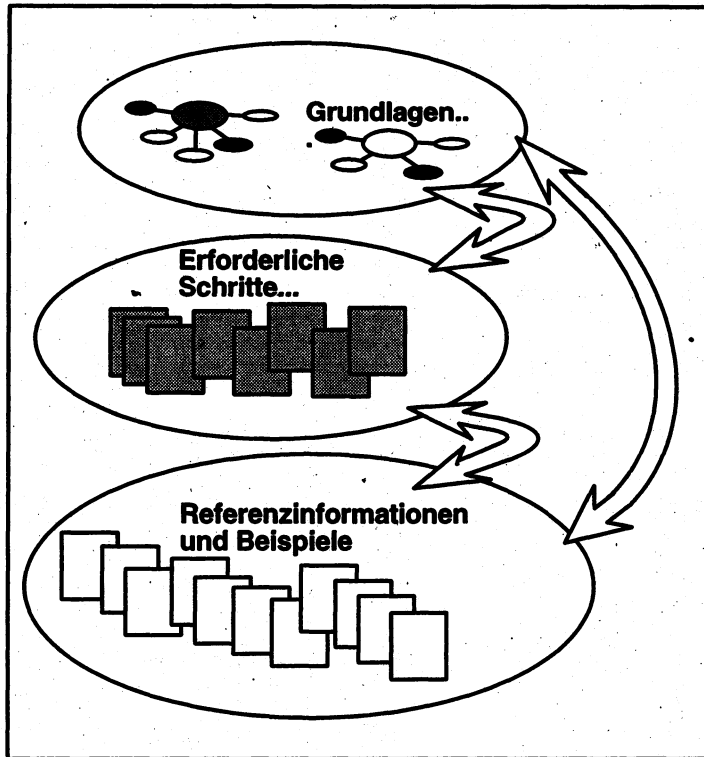
Navigationsinformationen: Stellen mehrere verschiedene Pfade für den Zugriff auf Informationen zur Verfügung.

Konzeptuelle Informationen: Bilden einen Bezugsrahmen für die Entscheidungsfindung und die Entwicklung von individuellen, spezifischen Prozeduren. Konzeptuelle Informationen enthalten Verweise auf andere verwandte Konzepte.

Ablaufbezogene Informationen: Erläutern dem Benutzer nur die erforderliche Vorgehensweise, enthalten jedoch keine Erklärungen bestimmter Sachverhalte.

Referenzinformationen: Beschreiben einzelne Befehle, Aufrufe, Unterroutinen, Dateiformate, Funktionen etc. Enthalten außerdem zahlreiche Verwendungsbeispiele.

Wie in der folgenden Abbildung dargestellt kann der Benutzer über Hypertextverbindungen einfach von einer Informationsart zur nächsten gelangen.



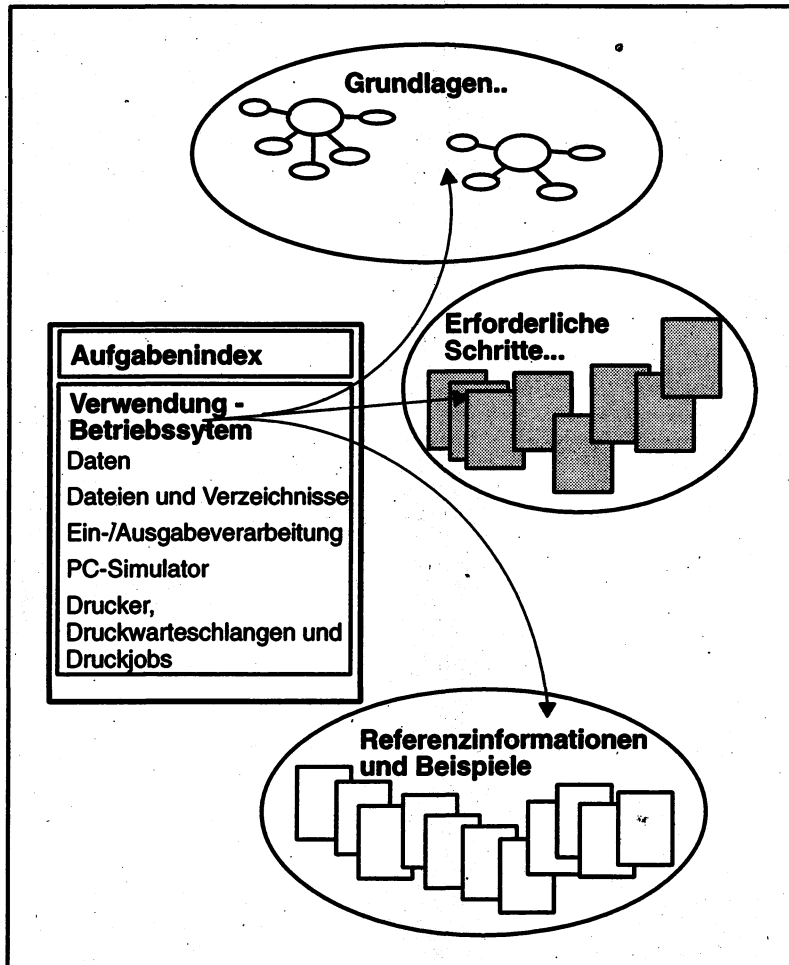
• Methode für das Abrufen von Informationen festlegen

Eine Vielzahl von Abruf- und Navigationsmethoden ermöglicht es neuen oder erfahrenen Benutzern, Informationen auf bequeme Art abzurufen.

Aufgabe	Abrufmethode
Die auszuführende Aufgabe ist bekannt, nicht jedoch die Vorgehensweise.	Aufgabenindex
Informationen zum verwendeten Befehl sollen angezeigt werden, z. B. zum Sendebefehl für elektronische Post.	Befehle
Alle Informationen zu Verwendung der Datenfernverarbeitung sollen angezeigt werden.	Bücher
Alle Informationen zu einem bestimmten Thema sollen angezeigt werden.	Suchen
Zuvor angezeigte Informationen sollen wieder angezeigt werden.	Protokoll
Eine bestimmte Artikelgruppe soll für Übungszwecke angezeigt werden.	Lesezeichen
Nur die aktualisierten Informationen sollen angezeigt werden.	Notizen

Aufgabenindex

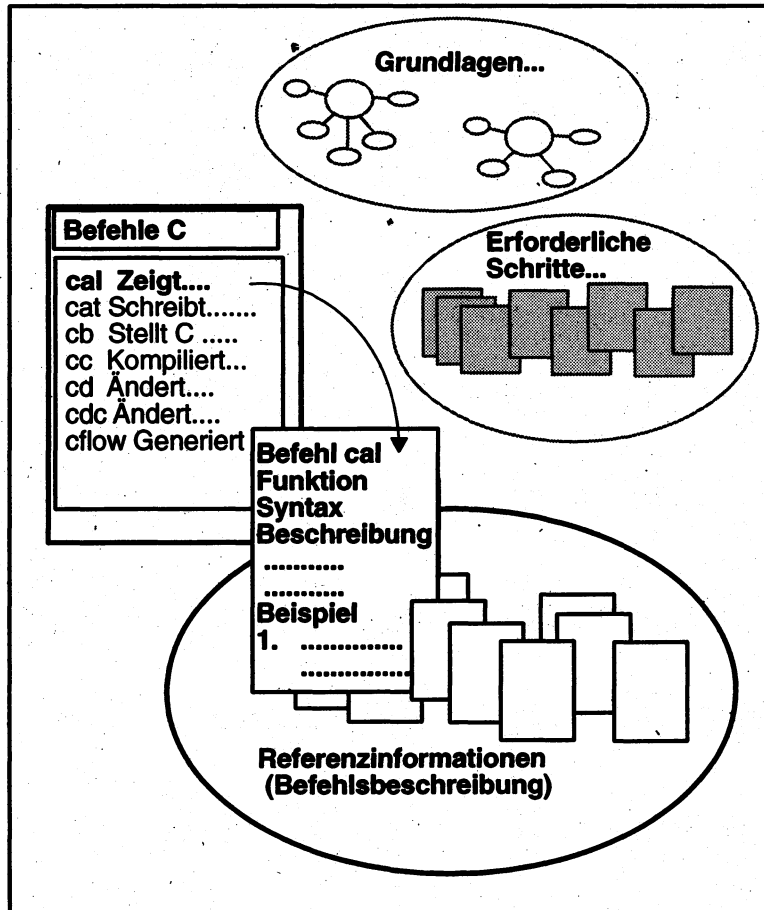
Der Aufgabenindex ist das primäre Navigationsdokument. Wie die Abbildung zeigt, kann über ihn das entsprechende Beispiel oder die Prozedur in der Datenbank aufgerufen werden. Über den Schulungsteil des Aufgabenindex können die konzeptuellen Informationen aufgerufen werden.



Sobald die Informationen angezeigt werden, können über Hypertextverbindungen andere themenbezogene Artikel aufgerufen werden.

Befehle

Der Befehlsindex ist ein Navigationsdokument, das sinnvoll eingesetzt werden kann, wenn der Benutzer mit den Systembefehlen bereits vertraut ist. Er enthält eine alphabetische Liste und Befehlsgruppen, die nach ihrer Funktion zusammengefaßt sind. Die folgende Abbildung zeigt, wie über den Befehlsindex der gewünschte Befehl aufgerufen werden kann.



Wörter suchen

Mit den Suchfunktionen kann der Benutzer eigene Navigationsdokumente erstellen. Fast alle Wörter in der Datenbank sind indexiert.

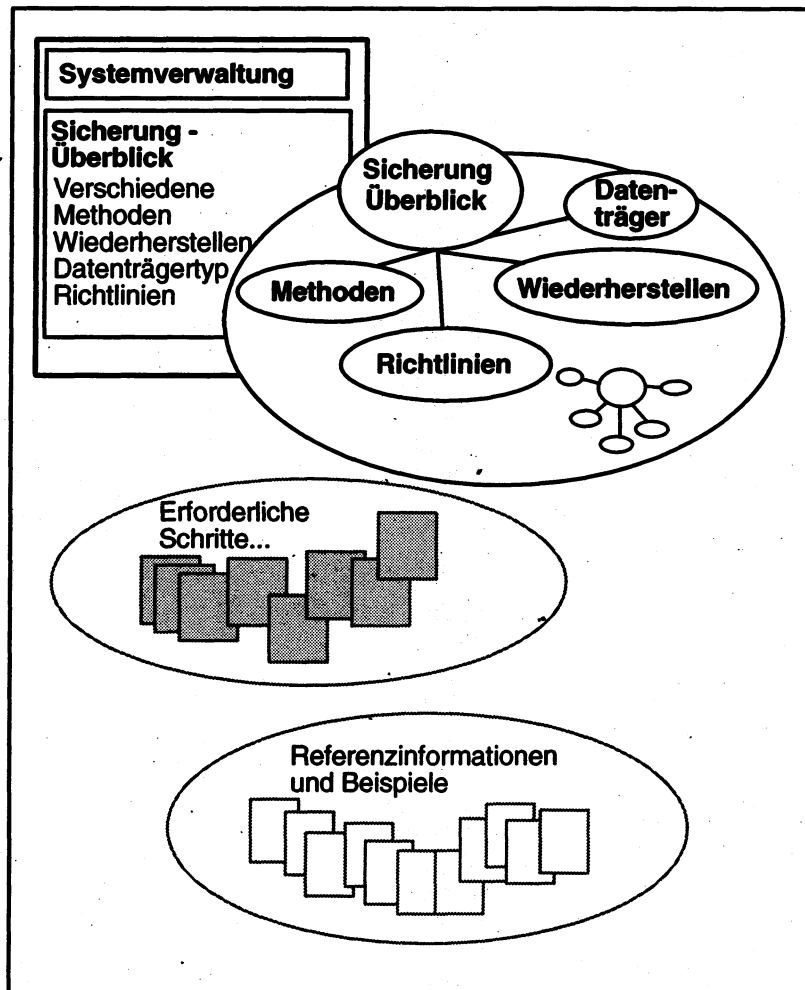
Der Benutzer hat folgende Optionen zur Auswahl:

- Datenbank ganz oder teilweise durchsuchen:
 - Titel
 - Überschriften
 - Gesamter Text in allen Artikeln
 - Beispiele
- Komplexe Suchabfragen mit Qualifikationsmerkmalen "und", "oder" und "aber nicht" erstellen.
- Komplexe Suchabfragen und -ergebnisse sichern.

- Die Suche nach Wunsch begrenzen. Die Suche kann beispielsweise auf das Protokoll, das Hauptnavigationfenster, den aktuellen Artikel oder eine bzw. mehrere Datenbanken begrenzt werden.

Bücher

Wie die Abbildung zeigt, können die Dokumentationen mit Hilfe der Bücherliste so aufgerufen werden, daß ihr Aufbau dem eines Buches ähnelt. Mit der Bücherliste können die Artikel wie Abschnitte in einem Buch gelesen werden.



Lesezeichen

Für häufig aufgerufene Artikel können Lesezeichen erstellt werden. Geladene Lesezeichen können als Navigationsdokument verwendet werden.

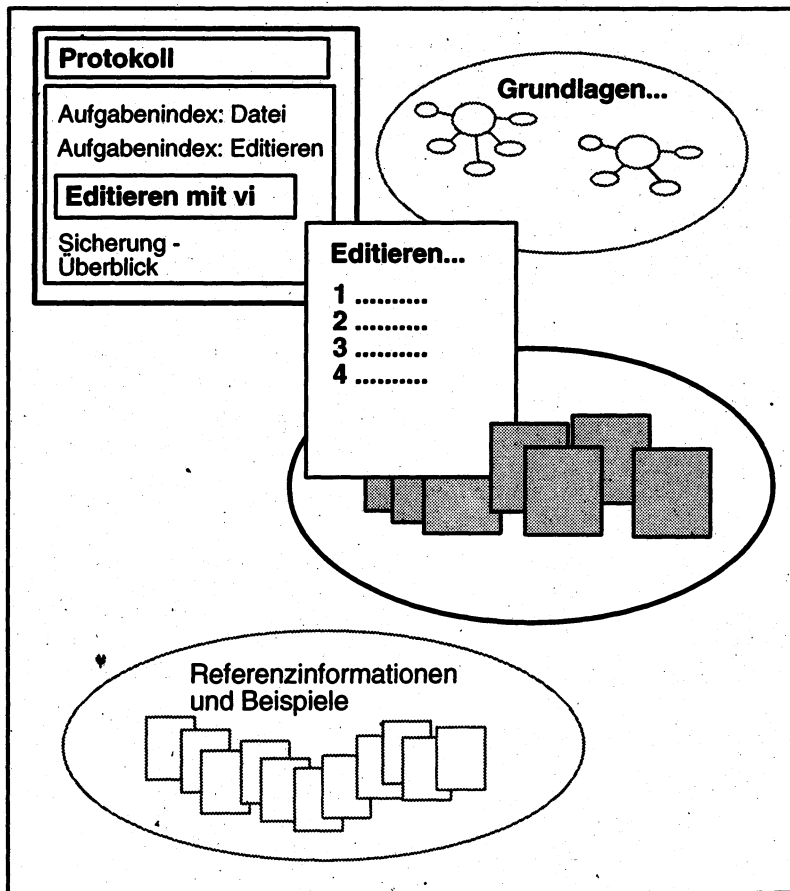
Lesezeichen können als persönliche Markierung verwendet werden, wenn der Lesevorgang unterbrochen wird.

Lesezeichen können in Lernprogrammen zum Hervorheben bestimmter Informationen verwendet werden, die für lokale Prozeduren relevant sind.

Lesezeichen können editiert und gesichert werden.

Protokoll

InfoExplorer zeichnet die Hypertextverbindungen auf, die aufgerufen wurden, und erstellt eine Protokolldatei. Wie die Abbildung zeigt, ist jeder Eintrag in der Protokolldatei eine Hypertextverbindung. Die Größe der Protokolldatei kann vom Benutzer festgelegt werden.



Notizen

Der Benutzer kann "Randnotizen" in Artikeln erstellen, um lokale Prozeduren hervorzuheben oder Informationen hinzuzufügen. Geladene Notizen können als Navigationsdokument verwendet werden.

Notiz können privat oder öffentlich sein.

Notizen können in Lernprogrammen oder zum Hervorheben lokaler Prozeduren verwendet werden.

Notiz können editiert und gesichert werden.

InfoExplorer-Schnittstellen

InfoExplorer enthält einige tausend Seiten mit Informationen zu Software und Hardware. Diese Informationen können in einer ASCII-Schnittstelle und einer Grafikschnittstelle angezeigt werden. Die Darstellung der Informationen und der Zugriff ist jedoch abhängig von der verwendeten Schnittstelle. In beiden Umgebungen besteht Zugriff auf effektive Abrufmethoden, die das Auffinden der gewünschten Informationen in einer Datenbank ermöglichen. Zu der Grafikschnittstelle enthält "InfoExplorer Grafikschnittstelle" im *InfoExplorer Benutzerhandbuch* weitere Informationen, zu der ASCII-Schnittstelle enthält "InfoExplorer-ASCII-Schnittstelle" weitere Informationen.

Neue Funktionen in InfoExplorer Version 1.2.0

InfoExplorer Version 1.2.0 stellt eine Reihe von neuen und geänderten Funktionen zur Verfügung. Dazu zählen erweiterte Optionen für Anpassung, Festlegen von Standardwerten, Suche und Drucken. Der Benutzer wird bei seiner Arbeit durch neue Fenster, Verbindungsarten und detaillierte Hilfetexte unterstützt. In den folgenden Abschnitten werden die erweiterten Funktionen von InfoExplorer Version 1.2.0 kurz beschrieben:

- Navigationsfunktionen
- Standardwerteanpassung
- Unterstützung mehrerer Bibliotheken
- Druckoptionen
- InfoExplorer-Grafikschnittstelle

Navigationsfunktionen

Bei InfoExplorer Version 1.2.0 wurden Änderungen an folgenden Navigationsfunktionen vorgenommen:

Aufgabenindex

Der "Aufgabenindex" ersetzt die "Aufgabenliste". Über diesen neuen Index können Informationen zu zahlreichen Themen und Aufgaben angefordert werden. Die Themen sind nach Zielgruppe und Betreff geordnet. Im Menü auf der höchsten Ebene (Eingangsebene) ist der Index nach den wichtigsten Aufgaben und Zielgruppen geordnet:

- Verwendung
- Verwaltung
- Programmierung
- Fehlerbehebung

Die nächste Indexebene ist nach Hauptthemen geordnet, zum Beispiel "Datenfernverarbeitung", "Einheiten" und "Benutzerschnittstelle". Auf der dritten und/oder vierten Ebene werden die entsprechenden Artikel angezeigt.

Deutsche Version

Über die neue Navigationsfunktion "Deutsche Version" können spezifische Informationen zur deutschen Version aufgerufen werden. Im englischen InfoExplorer ist an dieser Stelle die Funktion "Programming Reference" aktiv, die Referenzartikel zu Unterroutinen, Funktionen, Dateien etc. für jeden Programmiersachverhalt enthält.

Verbindungen (Links)

Die folgenden neuen Optionen stehen für InfoExplorer-Verbindungen zur Verfügung:

- InfoExplorer Version 1.2.0 umfaßt zwei neue Arten von Verbindungen:

Prozeßverbindungen

Prozeßverbindungen führen einen Prozeß aus. Bei der Auswahl einer Prozeßverbindung wird der Prozeß in einem anderen Fenster ausgeführt. InfoExplorer wird ausgesetzt, bis dieser Prozeß abgeschlossen ist.

Dateiverbindungen

Dateiverbindungen zeigen Dateien an. Dateiverbindungen stellen insbesondere eine Verbindung zu README-Dateien her, die mit dem Betriebssystem geliefert werden.

- Beim Anzeigen von InfoExplorer-Artikeln hat der Benutzer die Möglichkeit, Verbindungen anzuzeigen oder zu verdecken. Bei der Grafikschnittstelle steht oben im Lesefenster ein neuer Knopf zur Verfügung, mit dem zwischen den beiden Optionen umgeschaltet werden kann. Bei der ASCII-Schnittstelle steht diese Option in der Menüleiste zur Verfügung.

Hilfe

InfoExplorer bietet nun kontextspezifischere Hilfetexte mit detaillierten Beschreibungen der Funktionen, die in den einzelnen Fenstern oder Anzeigen zur Verfügung stehen. Über die Hilfefunktion kann weiterhin eine umfangreiche Liste mit Artikeln zur Verwendung von InfoExplorer angefordert werden.

Suchen

In InfoExplorer Version 1.2.0 wurden mehrere Änderungen an der Suchfunktion vorgenommen:

- InfoExplorer-Hilfetexte enthalten Hinweise zur effektiveren Verwendung der Suchfunktionen. Weiterhin erläutern die Hilfetexte, wie eine InfoExplorer-Sitzung bei einem angegebenen Navigationseingangsartikel gestartet werden kann. Es kann beispielsweise der folgende Befehl eingegeben werden:

```
info -h Speicher
```

Dieser Befehl startet eine InfoExplorer-Sitzung und zeigt einen Navigationsartikel an, der eine Liste mit Speicherverwaltungsaufgaben enthält. Über Hypertextverbindungen können Artikel im Handbuch aufgerufen werden, in denen diese Aufgaben beschrieben werden. (Ist InfoExplorer bei der Eingabe des Befehls **info** aktiv, wird der entsprechende Artikel angezeigt, es wird jedoch keine neue Sitzung gestartet.) Dieser Artikel kann auch mit der InfoExplorer-Suchfunktion angezeigt werden.

- Beim Ausführen einer komplexen Suche können Datenbanken ausgeschlossen werden. Mit dieser Funktion haben Benutzer, die über keine technischen Kenntnisse verfügen, zum Beispiel die Möglichkeit, Artikel mit Programmierinformationen von der Suche auszuschließen. Diese Einschränkung kann mit dem InfoExplorer-Standardwerteditor als Standardwert festgelegt werden.
- Wird bei einer Suche nur eine Übereinstimmung gefunden, übergeht InfoExplorer die Eingabeaufforderung und ruft den Artikel mit der Übereinstimmung direkt auf. Werden bei einer Suche nur in einer Datenbank Übereinstimmungen gefunden, zeigt InfoExplorer die Liste der Artikel mit Übereinstimmungen an.
- Alle Suchläufe werden jetzt über das Navigationsfenster ausgeführt. Die Suche nach Artikeltitle kann über das Fenster "Komplexe Suche" oder "Einfache Suche" gestartet werden.

Zitate und Verweise

Mit der neuen Zitat- und Verweisfunktion hat der Benutzer die Möglichkeit, die Beziehung zwischen Artikeln in der Datenbank anzuzeigen. Mit der Zitatfunktion werden alle Artikel angezeigt, die auf den aktuellen Artikel verweisen bzw. diesen zitieren. Mit der Verweisfunktion werden alle Artikel angezeigt, zu denen der aktuelle Artikel Verbindungen enthält. Artikel, auf die verwiesen wird bzw. die zitiert werden, werden als Hypertextverbindungen angezeigt.

Glossar

Mit der neuen Glossarfunktion können Definitionen aufgerufen werden. Mit der Glossarfunktion kann ein bestimmter Begriff oder Ausdruck in einem Artikel hervorgehoben werden. InfoExplorer durchsucht die Datenbank, in der sich das Glossar befindet. Alle

Begriffe und Ausdrücke, die den ausgewählten Text enthalten, werden mit den dazugehörigen Definitionen in einem Glossarfenster oder einer -anzeige ausgegeben.

Pfad

InfoExplorer Version 1.2.0 umfaßt folgende Änderungen der Pfadfunktion:

- Die Pfadfunktion steht jetzt auch in der InfoExplorer-ASCII-Schnittstelle zur Verfügung.
- In der Grafikschnittstelle steht die Pfadfunktion jetzt in Navigations- und Lesefenstern zur Verfügung.
- Der Pfad wird auf einer anderen Ebene zurückgesetzt. Zuvor wurde der Pfad von InfoExplorer bei jedem Zugriff auf das Navigationsfenster zurückgesetzt; in der neuen Version wird der Pfad nur zurückgesetzt, wenn eine andere Navigationsart ausgewählt wird.
- Bei der Pfadfunktion wird jetzt aufgezeichnet, an welcher Position ein Artikel aufgerufen und verlassen wird.

Protokoll

Mit der Protokollfunktion können verwendete Verbindungen und durchgeführte Aufgaben in einer InfoExplorer-Sitzung aufgezeichnet und wiederaufgerufen werden. Diese Funktion wurde um die Protokollierung von Übersichten, Zitaten, Verweisen, Glossarsuchen, einfachen und komplexen Suchen und Datei- und Prozeßverbindungen erweitert.

Notizen

Die folgenden Funktionen gelten für Notizen in InfoExplorer Version 1.2.0:

- Mit Hilfe von Schablonen können Notizen in einem Standardformat erstellt oder Standardinformationen, wie Name oder Datum, in bestimmte Gruppen von Notizen eingegeben werden.
- In InfoExplorer können nun auch Notizen erstellt werden, ohne daß zuvor Artikeltext ausgewählt werden muß. Wird Text ausgewählt, enthält die erstellte Notiz diesen Text. Wurde Text in eine Notiz kopiert, kann der Inhalt der Notiz editiert oder ergänzt werden, bzw. es kann auf den Inhalt der Notiz verwiesen werden.
- Der Befehl **mergenote**, mit dem mehrere Notizdateien in einer Notizdatei zusammengefaßt werden, ersetzt die Befehle **pknote** und **unpknote**.

Standardwerteanpassung

Mit dem neuen Standardwerteditor können eine Anzahl von Optionen festgelegt werden, zum Beispiel die Datenbanken für eine Suche, das Navigationsdokument, das beim Start von InfoExplorer angezeigt wird, und die Standarddrucker. Mit dem Standardwerteditor können auch vordefinierte Dateien mit Lesezeichenlisten, Protokollisten und Notizlisten geladen werden.

Unterstützung mehrerer Bibliotheken

In InfoExplorer kann jetzt zwischen mehreren Datenbanken einschließlich vom Benutzer entwickelter Datenbanken gewählt werden. Alternative Datenbanken müssen mit dem Programm InfoCrafter vorbereitet werden. Zum Arbeiten mit dieser Funktion ist eine erweiterte Lizenz für InfoExplorer erforderlich.

Druckoptionen

Mit dem Standardwerteditor können Standarddrucker oder Standarddruckerpfade für verschiedene Arten der Druckausgabe angegeben werden. Es kann ein Drucker für

Normalschrift, ein Drucker für Schönschrift (Informationen zu Schriftart und Darstellung bleiben erhalten) sowie ein PostScript-Drucker für Bildmaterial angegeben werden.

InfoExplorer-Grafikschnittstelle

Mehrere Änderungen in InfoExplorer gelten ausschließlich für Funktionen der Grafikschnittstelle.

- Über das neue Dienstprogrammfenster können Übereinstimmungen, Übersichten, Zitate und Listen angezeigt werden, ohne daß der Kontext des Navigationsdokuments aus der Anzeige gelöscht wird.
- Das Symbol "Fenster Halten" im Lesefenster wurde durch einen Knopf ersetzt, der zwischen **Halten** und **Freigeben** umgeschaltet werden kann, und die Aktion angibt, die bei Auswahl des Knopfes ausgeführt wird.
- Bildmaterial kann jetzt auf zwei Arten angezeigt werden: in einem Lesefenster mit Artikeltext oder, wie zuvor, in einem separaten Bildfenster.
- Über den Profileditor stehen weitere Schriftgrößen für die Textanzeige zur Auswahl.
- Abgeschnittener Text wird durch einen Pfeil am rechten Fensterrand gekennzeichnet.
- Es stehen jetzt auch hoch- und tiefgestellte Zeichen zur Verfügung.

Hilfetextkennungen in InfoExplorer

Hilfetextkennungen sind in vielen Artikeln in der Bibliothek enthalten, um dem Benutzer schnell Informationen zu bestimmten Aufgaben zur Verfügung zu stellen. Über die Hilfetextkennungen wird der Benutzer direkt zu einem Befehlsartikel oder einer Inhaltstliste geführt, die die zu einer bestimmten Aufgabe verfügbaren Informationen zeigen. Um diese Kennungen zu suchen, kann entweder beim Starten von InfoExplorer die Option **-h** des Befehls **info** verwendet werden oder einem Begriff, das die gesuchte Aufgabe kennzeichnet, kann das Präfix **hlp** hinzugefügt werden. Um beispielsweise Informationen zum Drucken zu suchen, in der Befehlszeile folgendes eingeben:

```
info -h print
```

Mit diesem Befehl wird InfoExplorer gestartet und jeder Artikel angezeigt, der die Zeichenfolge **hlpprint** enthält. Vom InfoExplorer-Menü "Suche" aus können über die Zeichenfolge **hlpprint** die gleichen Artikel aufgerufen werden. .

Auch Platzhalterzeichen können bei der Suche nach Hilfetextkennungen nützlich sein, zum Beispiel:

```
info -h print*
```

Über diese Suche wird jeder Artikel angezeigt, der Hilfetextkennungen zu **print**, **prints** oder **printing** enthält.

Die folgende Liste enthält Beispiele der in der Bibliothek verfügbaren Hilfetextkennungen:

- Betriebssystemaufgaben, wie:

```
file  
directory  
print  
SMIT  
start  
stop  
security  
shell
```

- Datenfernverarbeitungsaufgaben, wie:

```
em78  
ate  
bnu  
hcon  
mail
```

- Programmierungsaufgaben, wie:

```
streams  
xdr  
sockets  
lvm  
iohan
```

Erste Schritte (InfoExplorer-ASCII-Schnittstelle)

InfoExplorer ist ein effizientes Hilfsprogramm zum Abrufen von Texten, das Navigationshilfen zum Auffinden und Verwalten von Informationen verwendet. Für den ersten Einstieg ist die Kenntnis der wichtigsten Navigationshilfen ausreichend. Mit Hilfe der in den folgenden Abschnitten dargestellten grundlegenden Schritte kann der Benutzer mit der InfoExplorer-Datenbank arbeiten, in der weitere Informationen enthalten sind.

In diesem Artikel werden folgende Schritte erläutert:

- InfoExplorer starten
- Anzeigarten erkennen
- Hypertextverbindung auswählen
- Text in einer Anzeige durchblättern
- Menüoption auswählen
- Zu einer vorherigen Position zurückkehren
- Hilfe zu einer bestimmten Anzeige aufrufen
- Informationen suchen
- Liste der Hilfetexte aufrufen
- InfoExplorer verlassen

InfoExplorer starten

Zum Starten von InfoExplorer folgenden Befehl bei der Systemeingabeaufforderung eingeben und die Eingabetaste drücken:

```
info
```

Wurde das Programm bisher noch nicht gestartet, InfoExplorer jetzt mit diesem Befehl starten.

Der Artikel "Einführung in die InfoExplorer-ASCII-Schnittstelle" wird in der Leseanzeige ausgegeben. Der Artikel enthält eine Einführung in das Arbeiten mit Anzeigen und in Verfahren für den Zugriff auf die Hilfefunktion, das Copyright und die Warenzeichen bei InfoExplorer.

[Info](#) [Hilfe](#) [Anzeige](#) [Protokoll](#) [Lesezeichen](#) [Notizen](#) [Suchen](#) [Ende](#).

Einführung in die InfoExplorer-ASCII-Schnittstelle

Info-Dokument

Einführung in die InfoExplorer-ASCII-Schnittstelle

Bei InfoExplorer kann auf Hardware- und Softwareinformationen zum IBM RISC System/6000 zugegriffen werden, die in einer Online-Datenbank gespeichert sind. Die Datenbank enthält Artikel, die über Hypertextverbindungen verknüpft sind. Eine Hypertextverbindung stellt einen Pfad zu einer anderen Textstelle dar.

Hypertextverbindungen werden bei InfoExplorer als unterstrichene Textstellen angezeigt. Die aktuelle Verbindung wird in Umkehranzeige dargestellt. Über die folgenden Verbindungen können weitere Informationen zu InfoExplorer aufgerufen werden. Mit der Tabulatortaste kann zwischen den einzelnen Verbindungen gewechselt werden. Wird die gewünschte Verbindung hervorgehoben, die Eingabetaste drücken:

[Grundlegende Informationen zu InfoExplorer-Anzeigen](#)

[InfoExplorer Hilfefunktion](#)

[Antwortbrief](#)

[Copyright und Warenzeichen](#)

Anzeigearten erkennen

In InfoExplorer wird zwischen zwei grundlegenden Anzeigearten unterschieden: der *Navigationsanzeige* und der *Leseanzeige*.

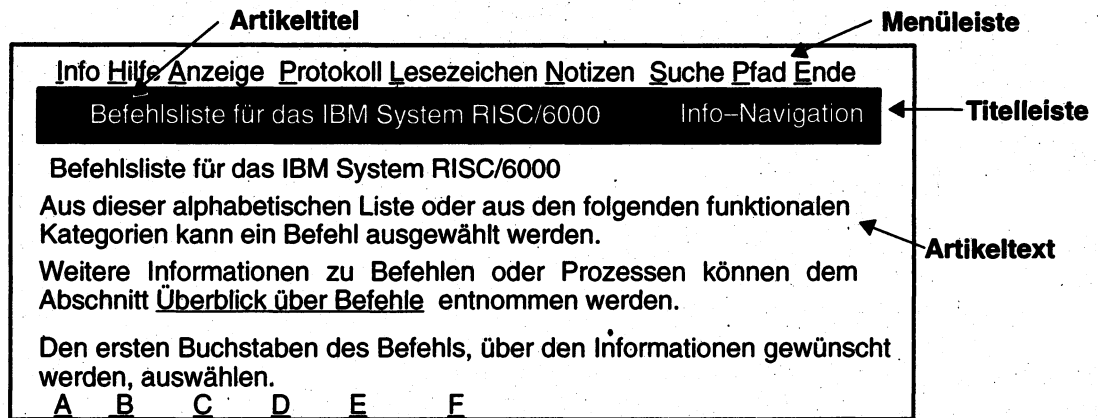
Navigationsanzeige

Die Navigationsanzeige ist der Ausgangspunkt zum Auffinden der Dokumentation zum Betriebssystem und zu anderen Programmen, die auf der Datenstation verfügbar sind. Sie enthält Informationen, die den Benutzer beim Auffinden der gewünschten Dokumentation unterstützen. Die Informationen sind nach Themen und Aufgaben, nach Buch oder alphabetisch nach Befehlen sortiert. Die primären Navigationsoptionen werden in der Navigationsanzeige ausgegeben:

- **Aufgabenindex** – zeigt Informationen nach Aufgaben geordnet an. Dies ist der Standardnavigationsartikel.
- **Befehle** – zeigt eine alphabetische Liste der verfügbaren Befehle an.
- **Bücher** – zeigt die Artikel in der Reihenfolge der Bücher an.
- **Deutsche Version** – zeigt für die deutsche Version spezifische Informationen an. (In der englischen Version ist dieser Knopf mit "Programming Reference" belegt und zeigt Listen mit Programmierungsfunktionen an, die aber in der deutschen Version nicht verfügbar sind.)

Hinweis: Die Navigationsinformationen in diesem Abschnitt gelten für das InfoExplorer, das mit dem Betriebssystem geliefert wurde. Die Navigationsinformationen gelten möglicherweise nicht für andere Bibliotheken.

In der ersten Zeile der Anzeige befindet sich die Menüleiste, in der zweiten Zeile die Titelleiste in Umkehranzeige. Der Titel der Anzeige wird auf der rechten Seite der Titelleiste angegeben. Diese Titelleiste trennt außerdem die Menüoptionen vom Artikeltext.

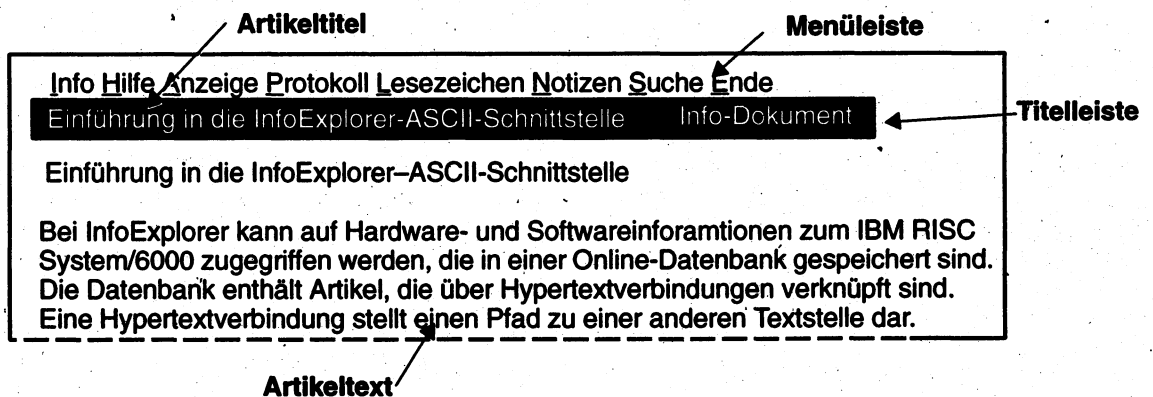


Wenn mit InfoExplorer in der ASCII-Schnittstelle gearbeitet wird, steht jeweils nur eine Navigationsanzeige zur Verfügung. Jedesmal, wenn eine Verbindung zu einem anderen Navigationsartikel ausgewählt wird, werden die Informationen aktualisiert und geben den Inhalt des ausgewählten Artikels wieder. Wird zum Beispiel der Aufgabenindex angezeigt, und anschließend die Befehlsliste aufgerufen, wird der Aufgabenindex in der Navigationsanzeige durch eine Liste mit Befehlen ersetzt.

Leseanzeige

Die Leseanzeige enthält ablaufbezogene, konzeptuelle und Referenzinformationen. Ablaufbezogene Informationen beschreiben das Ausführen einer Aufgabe. Konzeptuelle Informationen behandeln verschiedene Themen. Die Referenzinformationen sind Beschreibungen der einzelnen Befehle, Unterroutinen etc.

Wie bei der Navigationsanzeige befinden sich in der ersten Zeile die Menüoptionen, die zweite Zeile ist die Titelleiste in Umkehranzeige. Der Titel der Anzeige, Info-Dokument, wird rechts in der Titelleiste angezeigt. Diese Titelleiste trennt außerdem die Menüoptionen vom Artikeltext.



Wenn mit InfoExplorer in der ASCII-Schnittstelle gearbeitet wird, steht jeweils nur eine Leseanzeige zur Verfügung. Jedesmal, wenn eine Verbindung zu einem anderen Artikel ausgewählt wird, wird der Inhalt der Anzeige durch den Text des ausgewählten Artikels ersetzt. Wird beispielsweise der Artikel "InfoExplorer (ASCII-Schnittstelle) starten" angezeigt und wird anschließend der Artikel zum Befehl **Info** aufgerufen, ersetzt dieser Artikel den Artikel "InfoExplorer (ASCII-Schnittstelle) starten" in der Leseanzeige.

Zwischen Anzeigen umschalten

Soll die Navigationsanzeige über die Leseanzeige aufgerufen werden (oder über die Navigationsanzeige zu einer aktuellen Leseanzeige zurückgekehrt werden), die Steuertaste (Strg) gedrückt halten und die Taste W drücken (Strg-W). Mit der Tastenfolge Strg-W wird zwischen der Navigationsanzeige und Leseanzeige umgeschaltet. Zwischen Anzeigen kann auch mit Hilfe der Menüoption in der Menüleiste umgeschaltet werden. Dies wird an späterer Stelle in diesem Kapitel (siehe Seite 8-19) erläutert.

Zur Übung einige Male zwischen der Navigationsanzeige und der Leseanzeige umschalten und dabei die Unterschiede der beiden Anzeigearten beachten.

Außerdem stehen Anzeigen für *Suche*, *Glossar*, *Bildmaterial*, *Dateien* und *Zusatzfunktionen* zur Verfügung. Weitere Informationen zu den verschiedenen Anzeigearten sind im Abschnitt "Hilfetext zu einer Anzeige aufrufen" auf Seite 8-20 zu finden.

Hypertextverbindung auswählen

Unter einer Hypertextverbindung versteht man eine Verbindung zwischen zwei Informationseinheiten. Diese Verbindungen werden als unterstrichener Text dargestellt. Wird der Cursor auf eine Verbindung gesetzt, wird diese in Umkehranzeige dargestellt. Durch Drücken der Eingabetaste werden die Zielinformationen bzw. die zugehörigen Informationseinheiten angezeigt.

In der Anzeige "Einführung in die InfoExplorer-ASCII-Schnittstelle" werden fünf Hypertextverbindungen angezeigt. Sie werden als Liste mit fünf unterstrichenen Einträgen dargestellt.

<p><u>Info</u> <u>Hilfe</u> <u>Anzeige</u> <u>Protokoll</u> <u>Lesezeichen</u> <u>Notizen</u> <u>Suche</u> <u>Ende</u></p> <p>Einführung in die InfoExplorer-ASCII-Schnittstelle Info-Dokument</p> <p>Hypertextverbindungen werden bei InfoExplorer als unterstrichene Textstellen angezeigt. Die aktuelle Verbindung wird in Umkehranzeige dargestellt. Über die folgenden Verbindungen können weitere Informationen zu InfoExplorer aufgerufen werden. Mit der Tabulatortaste kann zwischen den einzelnen Verbindungen gewechselt werden. Wird die gewünschte Verbindung hervorgehoben, die Eingabetaste drücken:</p> <p><u>Grundlegende Informationen zu InfoExplorer-Anzeigen</u> <u>InfoExplorer Hilfefunktion</u> <u>Neue InfoExplorer-Funktionen</u> <u>Antwortbrief</u> <u>Copyright und Warenzeichen</u></p>
--

Hypertextverbindung

An einer ASCII-Datenstation kann kein Bildmaterial angezeigt werden. Wird eine Verbindung zu Bildmaterial ausgewählt, wird in einer Nachricht in einer Dialogfensteranzeige mitgeteilt, daß das Bildmaterial nicht angezeigt werden kann. Der Benutzer wird aufgefordert, zum Fortfahren eine beliebige Taste zu drücken. Der Überschrift "Syntax" in Artikeln zu Befehlen kann eine Verbindung zu einem grafischen Syntaxdiagramm zugeordnet werden. Bei Auswahl dieser Verbindung wird die oben beschriebene Nachricht angezeigt.

Zum Aufrufen der nächsten Hypertextverbindung die Tabulatortaste oder die Tastenfolge Strg-F drücken. Ist die nächste Verbindung gegenwärtig nicht in der Anzeige sichtbar, wird die Anzeige aktualisiert (d. h. verschoben) und zeigt anschließend die nächste Verbindung an. Mit der Tastenfolge Strg-B kann zur vorherigen Hypertextverbindung zurückgekehrt werden.

Zum Auswählen einer Hypertextverbindung folgende Schritte ausführen:

1. Befindet sich der Textcursor nicht im Textbereich, die Tastenfolge Strg-O drücken.
2. Mit der Tabulatortaste oder der Tastenfolge Strg-F zur nächsten Hypertextverbindung wechseln.
3. Mit der Tastenfolge Strg-B zur vorherigen Hypertextverbindung wechseln.
4. Zum Auswählen der Hypertextverbindung die Eingabetaste drücken. Der Zieltext wird angezeigt.

Zur Übung mit den Tastenfolgen Strg-F und Strg-B einige Male zwischen den Hypertextverbindungen umschalten. Bevor mit dem nächsten Abschnitt fortgefahren wird, eine Verbindung im Bildschirm "Einführung in die InfoExplorer-ASCII-Schnittstelle" auswählen. Zur Erinnerung: Mit der Tastenfolge Strg-F wird zur primären Navigationsanzeige zurückgekehrt.

In Anzeigentext blättern

In einem Artikel in einer Navigations- oder Leseanzeige kann mit Hilfe der Tasten oder Tastenfolgen, die zu diesem Zweck definiert wurden, vor- und zurückgeblättert werden.

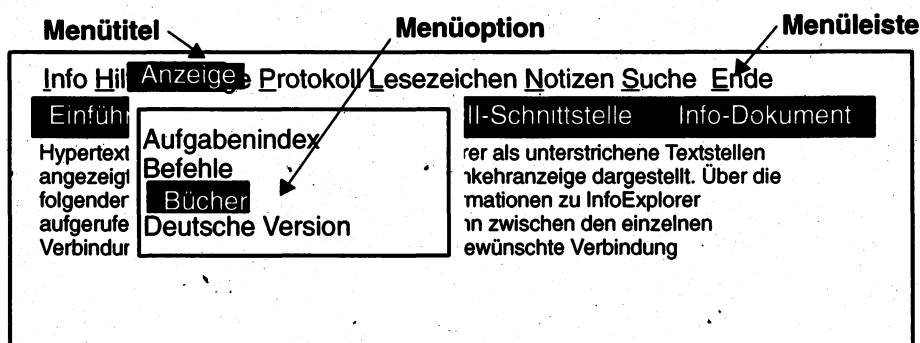
- Mit der Taste zum Rückwärtsblättern oder der Tastenfolge Strg-P zur vorherigen Anzeige zurückkehren.
- Mit der Taste zum Vorwärtsblättern oder der Tastenfolge Strg-N die nächste Anzeige aufrufen.
- Die Aufwärtspfeiltaste drücken, um den Textcursor um eine Zeile nach oben zu bewegen.
- Die Abwärtspfeiltaste drücken, um den Textcursor um eine Zeile nach unten zu bewegen.

Zur Übung den Cursor im Anzeigentext bewegen.

Angenommen, es werden Informationen zum Drucken einer Datei auf einem Zeilendrucker benötigt. Die Suche kann mit einem primären Navigationspfad, wie z. B. dem Aufgabenindex begonnen werden. Im Aufgabenindex können Verbindungen aktiviert werden, bis die Anzeige mit den gewünschten Informationen ausgegeben wird.

Menüoption auswählen

Jede InfoExplorer-Anzeige enthält am oberen Rand eine Menüleiste. Über die Menüoptionen in der Menüleiste können viele wichtige InfoExplorer-Funktionen aufgeführt werden. Beispiel: Im Menü **Anzeige** kann das Navigationsdokument ausgewählt werden, das in der Navigationsanzeige aufgerufen werden soll. Das Menü **Hilfe** enthält Informationen zur Verwendung von InfoExplorer.



Folgendermaßen vorgehen, um eine Menüoption auszuwählen:

1. Die Menüleiste mit der Tastenfolge Strg-O aktivieren.

2. Mit der Links- oder Rechtspfeiltaste den Titel des gewünschten Menüs hervorheben und die Eingabetaste drücken oder den unterstrichenen Buchstaben im Menütitel eingeben. Das Menü wird als Aktionsfenster mit verschiedenen Optionen angezeigt.
3. Mit der Aufwärts- oder Abwärtspfeiltaste eine Menüoption hervorheben oder den unterstrichenen Buchstaben in der Menüoption eingeben.
4. Die entsprechende Aktion wird ausgeführt oder ein weiteres Menü wird in einer Dialogfensteranzeige ausgegeben. Die Menüoptionen werden in der Menüleiste dieses Menüs angezeigt.

Folgendermaßen vorgehen, um eine Option in der Dialogfensteranzeige auszuwählen:

1. Die Menüleiste mit der Tastenfolge Strg-O aktivieren.
2. Eine Option hervorheben. Entweder den unterstrichenen Buchstaben der gewünschten Option eingeben oder die Links- oder Rechtspfeiltaste verwenden.
3. Die Option durch Drücken der Eingabetaste auswählen. Die Dialogfensteranzeige wird geschlossen und die entsprechende Aktion ausgeführt.

Soll ein angezeigtes Menü ohne Auswahl einer Option wieder geschlossen werden, die Tastenfolge Strg-O drücken. Der Cursor erscheint wieder in der Anzeige.

Bevor der nächste Abschnitt durchgearbeitet wird, die Arbeit mit den Menüs üben. Das Menü "Pfad" aufrufen und die Option **Liste** auswählen. Die Pfadliste wird in einer Dialogfensteranzeige ausgegeben. Zum Schließen der Dialogfensteranzeige **Ende** in der Menüleiste auswählen.

Zu einer vorherigen Position zurückkehren

In der InfoExplorer-Sitzung werden von InfoExplorer die bereits aufgerufenen Positionen in der Protokolliste und in der Pfadliste aufgezeichnet. Mit Hilfe dieser Listen kann zu zuvor aufgerufenen Positionen in der Datenbank zurückgekehrt werden.

In der Protokolliste werden alle in der Datenbank bereits aufgerufenen Positionen aufgezeichnet. Die Protokolliste kann durch Auswahl der Option **Alle auflisten** im Menü "Protokoll" angezeigt werden. Um Informationen zur Verwendung der Protokolliste anzuzeigen, die Menüoption "Hilfe" auswählen. Anschließend im Aktionsfenster die Option **Hilfaliste** auswählen. Der Artikel **In InfoExplorer** in der "Liste der Hilfetexte" enthält Informationen zur Protokolliste.

In der Pfadliste wird der Navigationspfad, ausgehend von einem primären Navigationsartikel (Aufgaben, Bücher, Befehle oder Deutsche Version) aufgezeichnet. Bei jeder Auswahl eines primären Navigationsartikels wird der vorherige Pfad gelöscht.

Gehe zu Hilfe Ende

Pfadliste

- *Mit dem Editor INed: par 1268
- Befehl just oder fjust: par 1
- *Befehl just oder fjust: par 32
- Befehl fill oder ffill: par 1
- *Befehl fill oder ffill: par 45

Mit dem Editor INed arbeiten; par 951

Im Menü Pfad stehen drei Optionen zur Verfügung: **Liste anzeigen**, **Vorheriger(r)** und **Nächste(r)**. Die Option **Liste anzeigen** auswählen, um die Pfadliste anzuzeigen. Mit den

Optionen **Vorherige(r)** und **Nächste(r)** kann zur vorherigen oder nächsten Position gewechselt werden, ohne die Pfadliste anzuzeigen.

Folgende Schritte ausführen, um zu einer vorherigen Position in InfoExplorer zurückzukehren:

1. Die Menüleiste mit der Tastenfolge Strg-O aktivieren.
2. "P" eingeben, um das Menü "Pfad" aufzurufen.
3. "L" eingeben, um die Pfadliste anzuzeigen. In einer Dialogfensteranzeige werden Positionen im Pfad aufgelistet. Wird die Pfadliste über eine primäre Navigationsanzeige aufgerufen, wird in einer Nachricht mitgeteilt, daß die Pfadliste keine Einträge enthält.
4. Die Position, zu der zurückgekehrt werden soll, mit der Aufwärts- bzw. Abwärtsfeiltaste hervorheben.
5. Die Option **Gehe zu** in der Menüleiste auswählen. Der Text wird in der Leseanzeige ausgegeben.

Zur Übung dem Pfad mit Hilfe der Optionen **Nächste(r)** und **Vorherige(r)** folgen. Es ist zu beachten, daß die Pfadliste gelöscht wird, wenn der Pfad bis zum primären Navigationsartikel (z. B. zur Aufgabenliste) zurückverfolgt wird. Das Zurückverfolgen ist erst wieder möglich, wenn ein neuer Pfad gestartet wird. Soll zu einer Position zurückgekehrt werden, die sich nicht im aktuellen Pfad befindet, die Protokollfunktion verwenden.

Hilfetext zu einer Anzeige aufrufen

InfoExplorer verwendet neben den Navigations- und Leseanzeigen viele verschiedene Anzeigearten:

- Anzeigen für das Glossar
- Anzeigen für die Liste mit Hypertextverbindungen
- Anzeigen für Dateien
- Anzeigen für einfache und komplexe Suche

Jede Anzeige dient zu einem bestimmten Zweck und stellt bestimmte Funktionen zur Verfügung. Das Menü "Hilfe" in jeder Anzeige enthält eine Option **Dieses Fenster**, mit der ein Artikel aufgerufen wird, der die in der aktuellen Anzeige verfügbaren Funktionen beschreibt.

Zum Anzeigen der Hilfetexte zu einer bestimmten Anzeige folgendermaßen vorgehen:

1. Die Menüleiste mit der Tastenfolge Strg-O aktivieren.
2. Mit den Pfeiltasten "Hilfe" hervorheben und die Eingabetaste drücken.
3. Mit den Pfeiltasten **Dieses Fenster** auswählen und die Eingabetaste drücken. Der aktuelle Artikel wird durch einen Artikel mit Hilfeinformationen ersetzt.

Bevor mit dem nächsten Abschnitt weitergearbeitet wird, zur Übung die Hilfeartikel zu den gegenwärtig geöffneten Anzeigen aufrufen. Mit der Pfadfunktion zu dieser Anzeige zurückkehren.

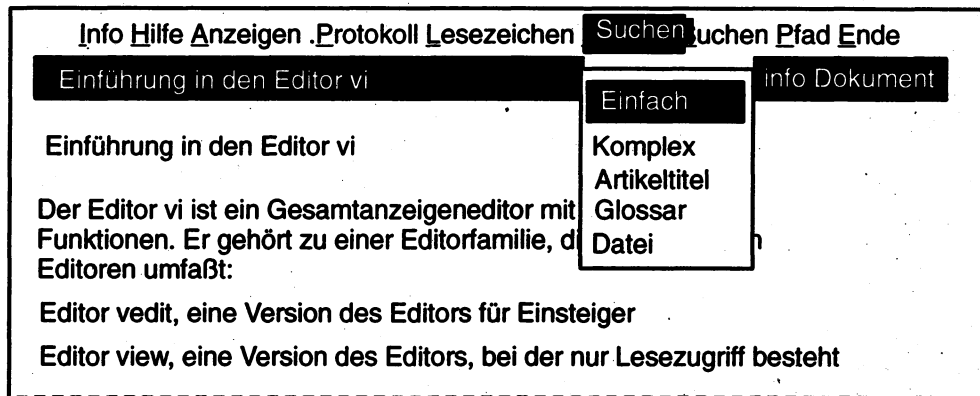
Informationen suchen

An früheren Stelle in diesem wurde erläutert, wie Informationen mit Hilfe der primären Navigationspfade gesucht werden. Voraussetzung hierfür ist jedoch, daß die gesuchte Aufgabe oder der gesuchte Befehl bekannt ist oder der Benutzer weiß, in welchem Buch die Informationen zu finden sind. Fast alle Wörter in der InfoExplorer-Datenbank sind indiziert. Mit der InfoExplorer-Suchfunktion können Informationen anhand von Wörtern oder Ausdrücken gesucht werden.

In InfoExplorer können zwei Arten der Suche ausgeführt werden: die einfache Suche und die komplexe Suche. Mit der einfachen Suche kann die gesamte Datenbank nach einem angegebenen Wort oder einer Wortgruppe durchsucht werden. Bei der komplexen Suche besteht die Möglichkeit, die Suche zu begrenzen, indem angegeben wird, welche Datenbank mit welcher Methode durchsucht werden soll. Im folgenden wird die einfache Suche nach dem Befehl **pr** beschrieben.

Zum Ausführen einer einfachen Suche nach dem Befehl **pr** folgende Schritte ausführen:

1. Das Menü "Suchen" in der Menüleiste aufrufen.



2. Die Option **Einfach** im Menü "Suchen" auswählen. Die Dialogfensteranzeige "Einfache Suche" wird aufgerufen.

3. In das Feld "Suchen" **pr** eingeben und die Eingabetaste drücken. Die Dialogfensteranzeige "Entsprechungsliste" wird aufgerufen und enthält folgende Informationen:

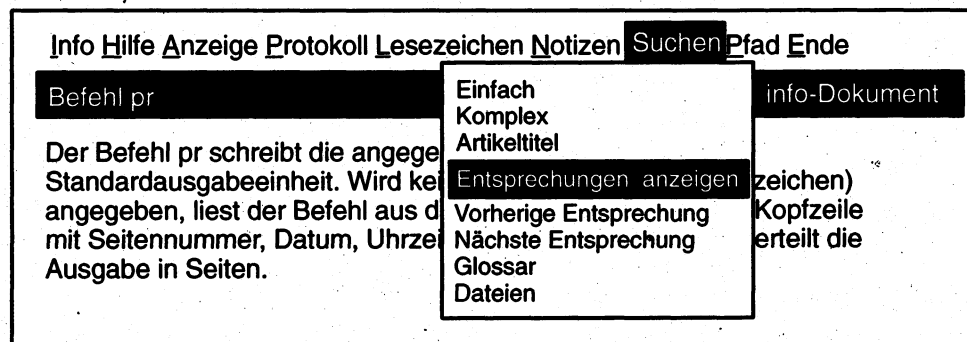
- Datenbanken, in denen Übereinstimmungen gefunden wurden
- Die Anzahl der Übereinstimmungen, die gefunden wurden
- Die Anzahl der Artikel mit Übereinstimmungen

4. Mit der Aufwärts- oder Abwärtspfeiltaste die Option **Anwendung, Verwaltung und Befehle** aus der Liste der Datenbanken auswählen und die Eingabetaste drücken. Das Dialogfenster "Suchen:Titelliste" wird angezeigt und enthält eine Titelliste der Artikel, in denen sich der Suchbegriff (**pr**) befindet.

Hinweis: Enthält nur ein Artikel den Suchbegriff, wird dieser Artikel in einer Leseanzeige ausgegeben (es werden keine Dialogfenster angezeigt).

Entspr.		Titel		Suchen:Titelliste
16		Befehl pr		
2		Befehl diff		
2		Befehl ls		
2		Befehl paste		
1		Befehl cat		
1		Befehl csh		
1		Desktop-Hilfe Symbol Alternate_Printer		

5. Den ersten Titel in der Liste "Befehl pr" auswählen. Der Artikel wird in der Leseanzeige ausgegeben. Der Ausdruck "Befehl pr" ist hervorgehoben, so daß deutlich sichtbar ist, an welcher Stelle er im Artikel erläutert wird.
6. Zum Anzeigen der nächsten Übereinstimmung in der Menüleiste das Menü **Suchen** auswählen.
7. Die Menüoption **Nächste Entsprechung** auswählen und die Eingabetaste drücken. Die nächste Entsprechung für den Befehl "pr" wird angezeigt. Mit dieser Funktion können die Informationen der Datenbank durchsucht und andere Positionen in diesem Artikel bzw. in anderen Artikeln aufgerufen werden, in denen der Ausdruck "Befehl pr" behandelt wird.
8. Zum Anzeigen der vorherigen Übereinstimmung die Option **Vorherige Entsprechung** im Menü **Suchen** auswählen und die Eingabetaste drücken. Die vorherige Übereinstimmung für "Befehl pr" wird angezeigt.
9. Um die Titelliste erneut auszuwählen, die Option **Entsprechungen anzeigen** aus dem Menü **Suchen** auswählen und die Eingabetaste drücken. Die Liste der Titel wird wieder angezeigt.



Werden viele Entsprechungen gefunden, ist es unter Umständen schwierig, die gewünschten Informationen zu finden, da hierzu mehrere Artikel gelesen werden müssen. Mit Hilfe der komplexen Suche kann das Suchergebnis eingegrenzt werden. Nähere Informationen zur komplexen Suche können über die InfoExplorer Hilfeliste aufgerufen werden, über die schrittweise Anweisungen zu häufig verwendeten InfoExplorer-Prozeduren angezeigt werden können. Aus der Liste der Hilfetexte In InfoExplorer Informationen suchen auswählen.

Die Suche kann weiter eingegrenzt werden, indem das Menü "Komplexe Suche" aufgerufen und die Datenbank ausgewählt wird, in der sich die gewünschten Informationen mit großer Wahrscheinlichkeit befinden. Abhängig vom System kann unter mehreren Datenbanken ausgewählt werden. Sollen beispielsweise nur Informationen zur Anwendung und Verwaltung gesucht werden, die Datenbank "Anwendung, Verwaltung und Befehle" auswählen.

Um eine Datenbank im Dialogfenster "Komplexe Suche" anzugeben, folgende Schritte ausführen:

1. Mit der Tastenfolge Strg-O die Menüleiste aktivieren.
2. Die Option **DB** auswählen. Die Datenbankliste wird angezeigt.
3. Das Zeichen > (größer als) links von einer Datenbank in dieser Liste gibt an, daß diese Datenbank in die aktuelle komplexe Suche aufgenommen wird.
4. Folgendermaßen vorgehen, um die Datenbanken zu markieren, die nicht durchsucht werden sollen:

5. Mit der Aufwärts- und Abwärtspfeiltaste die Datenbank auswählen, die gegenwärtig markiert ist und die nicht durchsucht werden soll, und die Eingabetaste drücken. Die Datenbank wird hervorgehoben.
6. Mit der Tastenfolge Strg-O die Menüleiste in der Dialogfensteranzeige "Datenbankliste" aktivieren.
Die Option Umgehung auswählen. Das Zeichen > wird gelöscht.
Diese Prozedur für jede Datenbank wiederholen, die nicht durchsucht werden soll.
7. Folgendermaßen vorgehen, um die Datenbanken zu markieren, die in die Suche aufgenommen werden sollen:
 - a. Mit der Aufwärts- und Abwärtspfeiltaste die Datenbank auswählen, die gegenwärtig nicht in der Suche enthalten ist. Die Eingabetaste drücken. Die Datenbank wird hervorgehoben.
 - b. Mit der Tastenfolge Strg-O die Menüleiste in der Dialogfensteranzeige "Datenbankliste" aktivieren.
 - c. Die Option Auswählen angeben. Das Zeichen > wird links von der ausgewählten Datenbank angezeigt.
Diese Prozedur für jede Datenbank wiederholen, die durchsucht werden soll.
8. Nachdem die gewünschten Datenbanken ausgewählt wurden, mit der Tastenfolge Strg-O die Menüleiste aktivieren und die Option Ende aufrufen. Das Dialogfenster "Komplexe Suche" wird erneut angezeigt (Alle ausgewählten Datenbanken werden im Feld "Suchen" angezeigt). Die Suche kann jetzt fortgesetzt werden.

Um weitere Informationen aufzurufen, **Hilfe** aus der Menüleiste auswählen, anschließend **Hilfeli**ste→**Informationen suchen in InfoExplorer**→**Datenbanken für die Suche auswählen**.

Liste der Hilfetext aufrufen

Im Menü "Hilfe" gibt es die beiden Hilfoptionen **Hilfeli**ste und **Dieses Fenster**. Mit der Option **Dieses Fenster** können Informationen zur aktuellen Anzeige aufgerufen werden. Weitere Informationen zu dieser Option enthält der Artikel "Hilfe zu einer bestimmten Anzeige aufrufen". Durch Auswahl der Option **Hilfeli**ste im Menü "Hilfe" wird eine Liste mit Themenbereichen angezeigt, zu denen in InfoExplorer Hilfetexte verfügbar sind. Zum Arbeiten mit der Hilfeli

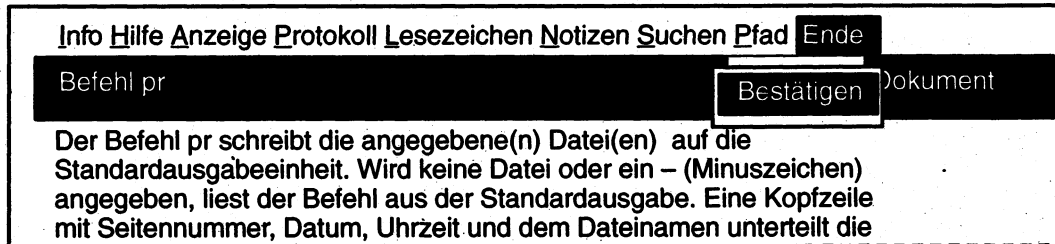
1. Die Menüleiste mit der Tastenfolge Strg-O aktivieren.
2. Mit den Pfeiltasten "Hilfe" hervorheben und die Eingabetaste drücken. Das Menü "Hilfe" wird angezeigt.
3. Die Option **Hilfeli**ste auswählen, um die Liste der verfügbaren Hilfetexte anzuzeigen.
4. Die Verbindung zu dem gewünschten Hilfetext aktivieren.

In der Liste der Hilfetexte sind die erforderlichen Prozeduren für zahlreiche Aufgaben beschrieben. Bevor mit dem nächsten Abschnitt fortgefahren wird und InfoExplorer verlassen wird, in der Liste der Hilfetexte zur Übung eine oder mehrere Optionen auswählen. Nachdem der Benutzer einen Überblick über die verfügbaren Hilfetexte gewonnen hat, sollte er anhand der Hilfetext neue Aufgaben erlernen und üben.

InfoExplorer verlassen

Soll InfoExplorer über die Navigationsanzeige oder die Leseanzeige verlassen, wie folgt vorgehen:

1. Die Menüleiste mit der Tastenfolge Strg-O aktivieren.
2. Mit den Pfeiltasten "Ende" hervorheben und die Eingabetaste drücken. Das Menü "Ende" wird angezeigt.
3. Die Option **Bestätigen** durch Drücken der Eingabetaste auswählen. Die Aufforderung zur Befehlseingabe des Betriebssystems wird angezeigt. InfoExplorer wird beendet.



InfoExplorer-Funktionen selbständig erarbeiten

Dieser Abschnitt enthält die wichtigsten Prozeduren, die zum ersten Arbeiten mit der InfoExplorer-Datenbank erforderlich sind. Mit der Suchfunktion kann z. B. eine beliebige Zeichenfolge in der Datenbank gesucht werden. In primären Navigationspfaden sind die Informationen nach Aufgabe, alphabetisch und nach Büchern geordnet. Durch die Hypertextverbindungen wird die Navigation in der Datenbank wesentlich vereinfacht.

Mit Hilfe dieser Funktionen können nicht nur Informationen zum Betriebssystem und anderen Software abgerufen werden. Darüberhinaus können Kenntnisse über das Programm InfoExplorer selbst erworben werden. Insbesondere die über das Aktionsfenster "Hilfe" abrufbare Liste der Hilfetexte enthält detaillierte Informationen zu vielen Prozeduren, einschließlich der folgenden:

- Ändern des Standardnavigationsartikels.
- Erstellen und Sichern privater Notizen.
- Erstellen und Sichern von Lesezeichen, mit denen Positionen in der Datenbank markiert werden können.
- Erstellen eines Antwortbriefs.

Darüberhinaus kann sich der Benutzer mit vielen InfoExplorer-Funktionen durch Aufrufen der verschiedenen Menüs in der Menüleiste vertraut machen.

Erste Schritte (InfoExplorer-Grafikschnittstelle)

InfoExplorer ist ein effizientes Hilfsprogramm zum Abrufen von Texten, das Navigationshilfen zum Auffinden und Verwalten von Informationen verwendet. Für den ersten Einstieg ist die Kenntnis der wichtigsten Navigationshilfen ausreichend. Mit Hilfe der in den folgenden Abschnitten dargestellten grundlegenden Schritte kann der Benutzer mit der InfoExplorer-Datenbank arbeiten, in der weitere Informationen enthalten sind.

Bei einigen Übungen ist es erforderlich, mehrere Fenster gleichzeitig zu öffnen. Um den Inhalt jedes Fensters anzuzeigen, kann es notwendig sein, das gewünschte Fenster zu verschieben oder seine Größe zu ändern. Informationen zum Verschieben und Vergrößern/Verkleinern von Fenstern sind in "Fensterposition festlegen" und "Fenstergröße in InfoExplorer festlegen" im *InfoExplorer Benutzerhandbuch* zu finden.

In diesem Kapitel werden folgende Schritte erläutert:

- InfoExplorer starten
- Fenstertypen
- Hypertextverbindung auswählen
- Knopf auswählen
- Menüoption auswählen
- Informationen durchblättern
- Lesefenster halten und freigeben
- Zu einer vorherigen Position zurückkehren
- Informationen suchen
- Hilfetext aufrufen
- InfoExplorer verlassen

InfoExplorer starten

Zum Starten von InfoExplorer folgende Schritte ausführen:

- Im AIXwindows Desktop den Mauszeiger auf das InfoExplorer-Symbol setzen und die linke Maustaste doppelt klicken. Der Aufgabenindex wird in einem Navigationsfenster angezeigt.
- In einer Befehlszeile des Betriebssystems in einem AIX-Fenster folgenden Befehl hinter der Eingabeaufforderung eingeben und die Eingabetaste drücken:

```
info
```

Im Lesefenster wird die Anzeige "Einführung in die InfoExplorer-Grafikschnittstelle" und im Navigationsfenster der Aufgabenindex angezeigt.

Der Aufgabenindex dient als Eingangspunkt für Hypertextinformationen, indem er aufeinander bezogene Aufgabengruppen unter den Hauptthemen "Verwendung", "Verwaltung", "Programmierung" und "Problemlösung" für das Betriebssystem und andere Programme zusammenfaßt, die im System installiert sind.

In der Einführung wird erläutert, wie die grundlegenden Operationen in den Fenstern durchzuführen sind und wie auf InfoExplorer-Hilfe-, Copyright- und Warenzeicheninformationen zugegriffen werden kann.

Hinweis: Überlagern sich die Fenster in der Anzeige, die Fenster so neu anordnen, daß beide gleichzeitig sichtbar sind.

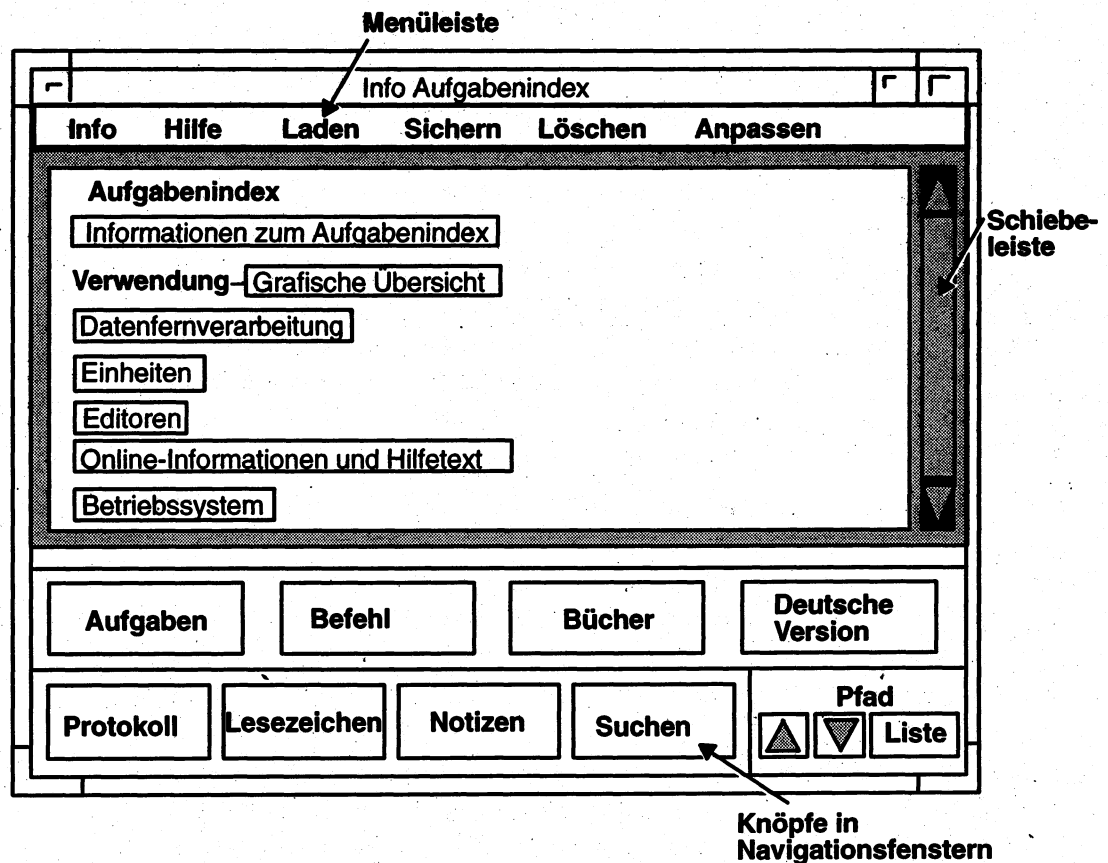
Wurde InfoExplorer noch nicht gestartet, das Programm jetzt starten.

Fenstertypen

In InfoExplorer wird zwischen zwei primären Fenstertypen unterschieden: *Navigationsfenster* und *Lesefenster*.

Navigationsfenster

Das Navigationsfenster enthält Informationen, die dem Benutzer beim Auffinden der gewünschten Dokumentation helfen. Die Informationen sind nach Themen und Aufgaben, nach Buch oder alphabetisch nach Befehlen sortiert. Die primären Navigationspfade werden im Navigationsfenster angezeigt:



Hinweis: Die Navigationsinformationen in diesem Abschnitt gelten für das Programm InfoExplorer, das mit dem Betriebssystem geliefert wurde. Die Navigationsinformationen gelten möglicherweise nicht für andere Bibliotheken.

- **Aufgaben** – zeigt Informationen nach Aufgaben geordnet an. Dies ist der Standardnavigationsartikel.
- **Befehl** – zeigt eine alphabetische Liste der verfügbaren Befehle an.
- **Bücher** – zeigt die Artikel nach Büchern geordnet an.
- **Deutsche Version** – zeigt für die deutsche Version spezifische Informationen an. (In der englischen Version ist dieser Knopf mit "Programming Reference" belegt und zeigt Listen

mit Programmierungsfunktionen an, die aber in der deutschen Version nicht verfügbar sind.)

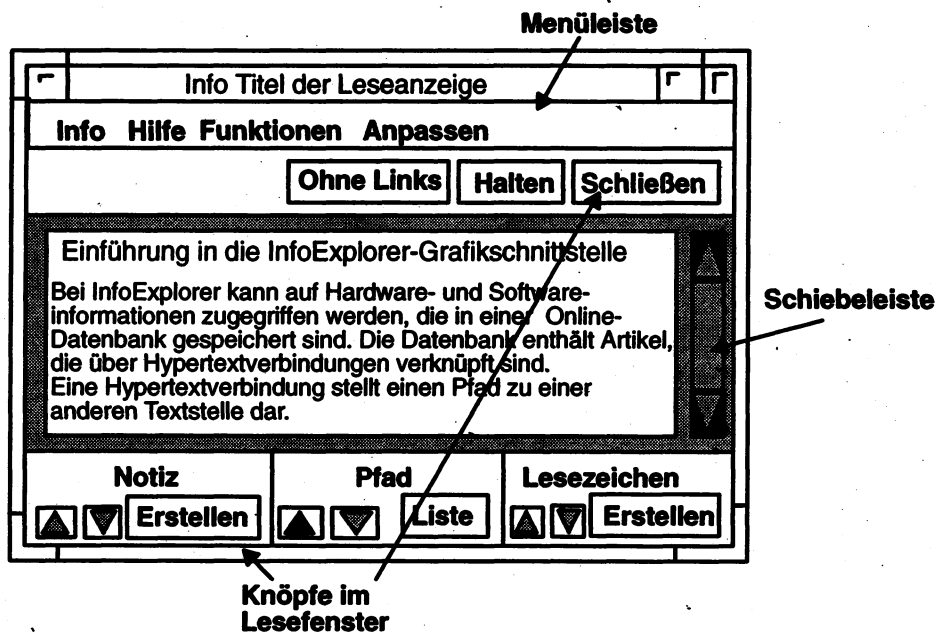
Im Navigationsfenster werden unten zwei Reihen mit Knöpfen angezeigt. In der oberen Reihe werden die vier primären Navigationspfade angezeigt, in der unteren Reihe die folgenden Hypertextnavigationsfunktionen:

- **Protokoll** – listet alle zuvor aufgerufenen Positionen in der Datenbank auf, einschließlich Suchläufe und Hilfetexte.
- **Lesezeichen** – listet alle Positionen auf, die mit einem Lesezeichen markiert wurden.
- **Notizliste** – listet alle Positionen auf, die durch eine Notiz gekennzeichnet sind.
- **Suchen** – ermöglicht die Suche in der Datenbank nach dem angegebenen Text.
- **Pfad** – Die vorherige oder nächste Position kann aufgerufen und eine Liste der Positionen kann angezeigt werden.

Der Navigationsartikel ist der Startpunkt für das Auffinden von Informationen. In InfoExplorer steht ein Navigationsfenster zur Verfügung. Jedesmal, wenn ein anderer Navigationsartikel ausgewählt wird, wird der Inhalt des Originalartikels ausgetauscht. Wird zum Beispiel der Aufgabenindex angezeigt, und anschließend die Befehlsliste ausgewechselt, wird der Aufgabenindex in der Navigationsanzeige durch eine Liste mit Befehlen ersetzt.

Lesefenster

Das Lesefenster enthält Referenzinformationen sowie ablaufbezogene und konzeptuelle Informationen. Ablaufbezogene Informationen beschreiben das Ausführen einer Aufgabe. Konzeptuelle Informationen behandeln verschiedene Themen. Die Referenzinformationen sind Beschreibungen der einzelnen Befehle, Unterroutinen etc.



Im Lesefenster werden die drei folgenden Knöpfe unten angezeigt:

- **Notiz** – Mit diesem Knopf kann eine Notiz im Text erstellt und die nächste oder vorherige Notiz aufgerufen werden.
- **Pfad, Blättern, Entsprechungen** – Mit jeder Option kann eine Liste mit Positionen angezeigt werden und die nächste oder vorherige Position angezeigt werden.

- **Lesezeichen** – Mit dieser Option kann ein Lesezeichen im Text erstellt und das nächste oder vorherige Lesezeichen aufgerufen werden.

Welche Knöpfe angezeigt werden, ist abhängig davon, welche Funktion beim Aufrufen des Lesefensters gerade durchgeführt wurde. Wurde beispielsweise gerade ein Suchlauf durchgeführt, wird der Knopf **Entsprechungen** anstelle des Knopfes **Pfad** unten im Lesefenster angezeigt.

Oben im Fenster werden drei Knöpfe angezeigt. Durch Auswahl dieser Knöpfe werden Funktionen des Lesefensters aktiviert:

- **Ohne Links** – Im Text werden keine Hypertextverbindungen angezeigt. Wird der Knopf **Mit Links** ausgewählt, werden alle Hypertextverbindungen im Text angezeigt.
- **Halten** – Mit diesem Knopf wird ein Leseartikel in einem neuen Fenster angezeigt, ohne den ursprünglichen Artikel zu ersetzen; wird der Knopf **Freigeben** ausgewählt, wird der aktuelle Artikel durch den neuen Artikel ersetzt.
- **Schließen** – Schließt das Lesefenster.

Weitere InfoExplorer-Fenstertypen

Zusätzlich zu den Navigations- und Lesefenstern verwendet InfoExplorer weitere Fenstertypen, die verschiedene Zwecke und Funktionen erfüllen. So gibt es beispielsweise Dienstprogramm-, Such-, Bildmaterial- und Glossarfenster sowie Fenster für Dateien des Betriebssystems.

Um weitere Informationen zu diesen Fenstern aufzurufen, die Option **Hilfeliste** (im Menü "Hilfe") auswählen und die Hilfeinformationen für die entsprechenden InfoExplorer-Fenster suchen. Wurde ein bestimmter Fenstertyp geöffnet, kann hierfür auch die Option **Dieses Fenster** im Menü "Hilfe" ausgewählt werden.

Hypertextverbindung auswählen.

Eine Hypertextverbindung ist eine Verbindung zwischen Informationen und wird als Text in einem Rahmen angezeigt. Über Hypertextverbindungen können Informationen angezeigt werden, die mit dem gegenwärtig angezeigten Artikel verknüpft sind.

Das Fenster "Einführung" zeigt fünf Hypertextverbindungen an. Diese werden als Liste mit fünf Einträgen angezeigt, die jeweils in einem Rahmen stehen.

Folgendermaßen vorgehen, um eine Verbindung auszuwählen:

1. Den Mauszeiger auf den Text im Rahmen setzen.
2. Die linke Maustaste drücken. Um zu verdeutlichen, daß die Verbindung ausgewechselt wurde, wird diese Verbindung in Umkehranzeige dargestellt. Kann die Aktion nicht sofort ausgeführt werden, wird ein Uhrensymbol angezeigt. Die Zielinformationen werden in demselben Fenster angezeigt.

Bevor der nächste Abschnitt durchgearbeitet wird, zur Übung eine Verbindung im Fenster "Einführung" auswählen.

Knopf auswählen

InfoExplorer verwendet verschiedene Knopftypen:

- Knöpfe in Lesefenstern, wie "Schließen", "Halten" und "Pfad".
- Knöpfe in Navigationsfenstern, wie "Aufgaben", "Bücher" und "Protokoll".
- Knöpfe mit speziellen Funktionen, wie "Gehe zu", "Abbruch" und "Blättern", die bei der Ausführung spezieller Funktionen angezeigt werden.

Auf einigen Knöpfen ist der Text fett hervorgehoben. Nur diese Knöpfe sind gegenwärtig aktiv und können ausgewählt werden.

Obwohl verschiedene Knopftypen in InfoExplorer verwendet werden, ist bei der Auswahl der Knöpfe immer dieselbe Vorgehensweise erforderlich:

1. Den Mauszeiger auf den Knopf setzen.
2. Die linke Maustaste drücken. Der Knopf wird jetzt in Umkehranzeige dargestellt. Die Umkehranzeige ist ein Hinweis darauf, daß dieser Knopf ausgewählt ist. Kann die Aktion nicht sofort ausgeführt werden, wird ein Uhrensymbol angezeigt.

Vor dem nächsten Schritt das Lesefenster mit dem Knopf "Schließen" schließen. Zur Übung einige Knöpfe im Navigationsfenster auswählen.

Angenommen, es werden Informationen zum Drucken einer Datei auf einem Zeilendrucker benötigt. Die Suche kann mit einem primären Navigationspfad, wie z. B. dem Aufgabenindex, begonnen werden. Beim Aufgabenindex beginnend können Verbindungen nacheinander ausgewählt werden, bis das Fenster mit den gewünschten Informationen angezeigt wird.

Menüoption auswählen

Über die Menüs können allgemeine Aufgaben ausgeführt werden, wie z. B. Lesezeichen sichern und löschen, Artikel drucken und Hilfetext anfordern. Am oberen Rand jedes InfoExplorer-Fensters wird eine Menüleiste angezeigt. Menüleisten enthalten zwei bis fünf Menütitel, abhängig vom Fenstertyp. Über jeden Menütitel kann ein Aktionsfenster mit den verfügbaren Optionen aufgerufen werden.

Folgendermaßen vorgehen, um eine Menüoption auszuwählen:

1. Den Mauszeiger auf den Menütitel in der Menüleiste setzen.
2. Die linke Maustaste drücken. Eine Liste mit Menüoptionen wird angezeigt.
3. Den Mauszeiger im Aktionsfenster auf die gewünschte Menüoption setzen.
4. Die linke Maustaste drücken. Die entsprechende Aktion wird ausgeführt.

Soll ein angezeigtes Menü ohne Auswahl einer Option wieder geschlossen werden, den Mauszeiger aus dem Menü herausbewegen und die linke Maustaste drücken. Das Menü wird gelöscht.

Informationen durchblättern

In vielen Fällen ist der Text in einem Fenster länger als das Fenster selbst. Um den Text zu lesen, muß geblättert werden. Entweder mit der Taste zum Vorwärtsblättern oder Zurückblättern jeweils um eine Anzeige vor-/zurückblättern oder die Schiebeleiste verwenden.

In der Text in einem Fenster breiter als das Fenster selbst (durch Pfeile nach rechts gekennzeichnet), muß das Fenster eventuell vergrößert werden, um den vollständigen Text anzuzeigen. Das Ändern der Fenstergröße wird in "Fenstergröße mit InfoExplorer festlegen" im *InfoExplorer Benutzerhandbuch* beschrieben.

Schiebeleisten

In InfoExplorer-Fenstern befindet sich eine vertikale Schiebeleiste an der rechten Seite. An jedem Ende der Schiebeleiste befindet sich ein Pfeil. Fenster mit Bildmaterial verfügen außerdem über eine horizontale Schiebeleiste am unteren Rand des Fensters, mit der das Bildmaterial nach rechts oder links verschoben werden kann.

Schiebeleisten dienen nicht nur zum Durchblättern von Text, sie geben dem Benutzer außerdem die ungefähre Länge des Artikels und die aktuelle Position im Artikel an.

Anhand der Größe der Schiebeleiste im Verhältnis zur Größe des Schiebereichs kann der Benutzer feststellen, welcher Teil des Artikels gegenwärtig angezeigt wird und wie lang der Artikel insgesamt ist. Beispiel: Ist die Schiebeleiste ungefähr halb so lang wie der Schiebereich, handelt es sich bei dem angezeigten Text um die Hälfte des Artikels. Der Artikel erstreckt sich also über zwei Anzeigen.

Die Position der Schiebeleiste im Schiebereich gibt an, welcher Teil des Artikels angezeigt wird. Befindet sich die Schiebeleiste oben im Schiebereich, ist der Anfang des Artikels in der Anzeige zu sehen. Befindet sich die Schiebeleiste unten im Schiebereich, ist das Ende des Artikels in der Anzeige zu sehen.

Blättern

Mit der Schiebeleiste kann auf zwei Arten geblättert werden: jeweils um eine Anzeige oder schrittweise.

Soll jeweils um eine Anzeige geblättert werden, folgendermaßen vorgehen:

1. Den Mauszeiger auf den kleinen Pfeil am unteren Ende der Schiebeleiste setzen.
2. Die linke Maustaste drücken. Die nächste Anzeige mit Text wird ausgegeben.

Soll schrittweise geblättert werden, folgendermaßen vorgehen:

1. Den Mauszeiger in den hellen Teil der Schiebeleiste setzen.
2. Die linke Maustaste drücken und gedrückt halten.
3. Die Maus langsam nach oben oder unten ziehen. Der Artikeltext wird entsprechend im Anzeigebereich verschoben.

Bevor mit dem nächsten Abschnitt weitergearbeitet wird, zur Übung mit beiden Methoden im Lesefenster blättern.

Lesefenster halten und freigeben

Das verwendete System ist möglicherweise so konfiguriert, daß bei jeder Auswahl einer Verbindung der Zieltext den zuvor angezeigten Artikel im Lesefenster ersetzt.

Mit dem Knopf "Halten" können jeweils mehrere Lesefenster geöffnet werden. Zur Übung mit der linken Maustaste den Knopf "Halten" (bzw. "Freigeben") mehrmals anklicken. Jedesmal, wenn der Knopf angeklickt wird, wird zwischen "Halten" und "Freigeben" hin- und hergeschaltet.

Wird auf dem Knopf "Halten" angegeben, den Knopf auswählen, um mehrere Fenster geöffnet zu lassen. Wird auf dem Knopf "Freigeben" angezeigt, wurden die Fenster bereits auf "Halten" gesetzt. Wenn "Freigeben" ausgewählt wird, können die alten Artikel durch neue Artikel ersetzt werden.

Um die Funktionen des Knopfes "Halten" zu prüfen, folgende Übung im Artikel "Befehl lp" ausführen:

1. Wird "Freigeben" auf dem Knopf angezeigt, nichts tun. Wird "Halten" angezeigt, den Knopf auswählen.
2. Zum Abschnitt "Empfohlene Dokumentation" am Ende des Kapitels blättern.
3. Eine Verbindung zu einem Artikel herstellen, der in diesem Abschnitt aufgelistet wird. Der Artikel wird in einem neuen Lesefenster angezeigt, das den ursprünglichen Artikel überlagert.

4. Das zweite Fenster schließen und den Knopf "Freigeben" im ersten Lesefenster auswählen. Auf dem Knopf sollte jetzt "Halten" angezeigt werden.
5. Erneut eine Verbindung zu einem Artikel herstellen, der im Abschnitt "Empfohlene Dokumentation" aufgelistet wird. Der neue Artikel ersetzt den zuvor angezeigten Artikel in diesem Fenster.

Bevor der nächste Abschnitt durchgearbeitet wird, zur Übung Verbindungen zu mehreren Artikeln im Haltemodus und im Freigabemodus auswählen.

Zu einer vorherigen Position zurückkehren

In der InfoExplorer-Sitzung zeichnet InfoExplorer die zuvor aufgerufenen Positionen auf zwei Arten auf: in der Protokolliste und der Pfadliste. Mit Hilfe dieser Listen kann zu zuvor aufgerufenen Positionen in der Datenbank zurückgekehrt werden.

In der Protokolliste werden alle in der Datenbank bereits aufgerufenen Positionen aufgezeichnet. Um die Protokolliste anzuzeigen, den Knopf "Protokoll" im Navigationsfenster auswählen. Sollen Informationen zur Verwendung der Protokolliste angezeigt werden, das Menü "Hilfe" aufrufen und die Option **Hilfelim** im Aktionsfenster auswählen. **Artikel aufrufen** der Liste der Hilfetexte enthält Informationen zur Protokollfunktion.

In der Pfadliste wird der Navigationspfad, ausgehend von einem primären Navigationsartikel (Aufgaben, Bücher, Befehle oder Deutsche Version) aufgezeichnet. Bei jeder Rückkehr zu einem primären Navigationsartikel wird die vorherige Pfadliste gelöscht.

Die Pfadfunktion umfaßt drei Knöpfe:

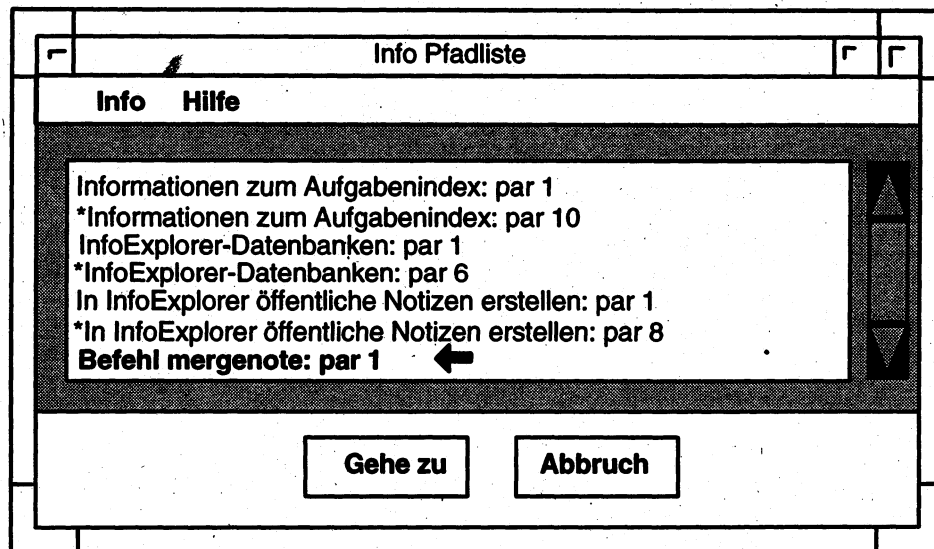
- Der Knopf "Fensterliste", der die Pfadliste in einem Dienstprogramm aufruft.
- Der Knopf mit dem Aufwärtspfeil, mit dem die Fenster aufgerufen werden, die zuvor bei der Auswahl von Verbindungspunkten angezeigt wurden.
- Der Knopf mit dem Abwärtspfeil, mit dem zu den Fenstern zurückgekehrt wird, die angezeigt wurden, bevor der Pfad (mit dem Aufwärtspfeil) zurückverfolgt wurde. Wurde die Pfadfunktion noch nicht verwendet, ist der Knopf mit dem Abwärtspfeil möglicherweise noch nicht aktiv, da keine Artikel im Pfad vorhanden sind.

Zur Übung den Pfad mit den Pfeilknöpfen vor- und zurückverfolgen. Es ist zu beachten, daß die Pfadliste gelöscht wird, wenn der Pfad bis zum primären Navigationsartikel (z. B. zur Aufgabenliste) zurückverfolgt wird. Das Zurückverfolgen eines Pfades ist erst wieder möglich, wenn ein neuer Pfad gestartet wird. Soll zu einer Position zurückgekehrt werden, die sich nicht im aktuellen Pfad befindet, die Protokollfunktion verwenden.

Über die Pfadliste kann eine beliebige Position im Pfad direkt aufgerufen werden:

1. Die Pfadliste durch Auswahl des Knopfes "Liste" aufrufen. Ein Dienstprogrammfenster wird geöffnet. Es enthält die Pfadliste.

In der Pfadliste gibt jeder Eintrag den Namen und die Position (Abschnitt) des Artikels an, zu dem eine Verbindung hergestellt wurde. Die Liste endet mit dem aktuellen Artikel. Dieser Eintrag ist in Fettdruck hervorgehoben.



2. Den Cursor auf eine Position in der Pfadliste setzen.
3. Die Position durch Anklicken mit der linken Maustaste auswählen.
4. Den Knopf "Gehe zu" auswählen. Das Lesefenster zeigt die angegebene Position an.
5. Den Knopf "Abbruch" auswählen, um das Fenster mit der Pfadliste zu schließen.

Bevor mit dem nächsten Abschnitt fortgefahren wird, zur Übung über die Pfadliste in einem geöffneten Lesefenster (nicht dieses Fenster)

Informationen suchen

In InfoExplorer können Informationen nicht nur durch das Zurückverfolgen primärer Navigationspfade aufgerufen werden. Mit der InfoExplorer-Suchfunktion können Informationen außerdem anhand eines Begriffs oder Ausdrucks gesucht werden. Praktisch jeder Begriff in der Hypertextdatenbank ist indexiert, so daß mit verschiedenen Suchmethoden nach Informationen gesucht werden kann, selbst dann, wenn der Name der Aufgabe oder des Befehls bzw. die Position, an der die Informationen gespeichert sind, nicht bekannt ist.

In InfoExplorer können einfache und komplexe Suchläufe ausgeführt werden. Bei der einfachen Suche kann die gesamte Datenbank nach einem einzelnen Begriff oder einer Wortgruppe durchsucht werden. Bei der komplexen Suche besteht die Möglichkeit, die Suche zu begrenzen, indem angegeben wird, welche Datenbank mit welcher Methode durchsucht werden soll. Beispielsweise können Artikeltitle gesucht werden, die eine bestimmte Zeichenfolge enthalten, oder Artikel, die eine Kombination aus mehreren Zeichenfolgen enthalten.

Mit den folgenden Schritten wird eine einfache Suche nach dem Befehl **pr** durchgeführt.

Zum Ausführen einer einfachen Suche nach dem Befehl **pr** folgende Schritte ausführen:

1. Im Navigationsfenster den Knopf "Suchen" auswählen. Wird das Fenster "Komplexe Suche" angezeigt, den Knopf "Einfache Suche" am unteren Rand des Fensters "Komplexe Suche" auswählen. Das Fenster Einfache Suche wird angezeigt.

2. Den Mauszeiger in das Feld "Suchen nach" setzen und die linke Maustaste drücken. `Befehl pr` eingeben. Für diese Übung die Option "Titel suchen" nicht auswählen.
3. Den Knopf "Eingabe" auswählen oder die Eingabetaste auf der Tastatur drücken: Ist der Suchbegriff in mehreren Datenbanken vorhanden, wird eine Entsprechungsliste angezeigt. Diese Liste gibt die Datenbanken und die Zahl der Artikel an, in denen die Übereinstimmungen gefunden wurden, sowie die Anzahl der Übereinstimmungen. Neben jeder Datenbank wird ein Kästchen mit einem Haken angezeigt :

Hinweis: Der Inhalt dieser Liste ist abhängig von den Datenbanken, die im System installiert sind.

4. Um die Übereinstimmungen in allen aufgelisteten Datenbanken anzuzeigen, den Knopf "Weiter" auswählen oder die Eingabetaste auf der Tastatur drücken. Um nur die Übereinstimmungen in einer Datenbank anzuzeigen, den Haken in dem Kästchen links von jeder Datenbank entfernen, deren Übereinstimmungen nicht angezeigt werden sollen. Um beispielsweise nur die Übereinstimmungen in der Datenbank **Anwendung, Verwaltung und Befehle** anzuzeigen, die Haken neben den beiden anderen Datenbanken in der Liste löschen.

Hinweis: Wird eine Datenbank, die durchsucht werden sollte, nicht in der Entsprechungsliste angezeigt, kann der Knopf "Suche erneut angeben"

ausgewählt werden. Anschließend im Fenster "Komplexe Suche" die Datenbanken auswählen, die durchsucht werden sollen.

5. Den Knopf "Weiter" anklicken, um die ausgewählte Datenbank zu durchsuchen. Eine Liste der Artikeltitel, die den Suchbegriff enthalten, wird im Dienstprogrammfenster "Titelliste" angezeigt.
6. Eine Verbindung zum Titel: "Befehl pr" herstellen. Der Artikel wird im Lesefenster ausgegeben. Es ist zu beachten, daß die Funktion "Entsprechungen" anstelle der Funktion "Pfad" am unteren Rand des Fensters angezeigt wird.
7. Den Abwärtspfeil unter "Entsprechungen" anklicken. Die nächste Entsprechung für "Befehl pr" wird angezeigt. Mit der Funktion "Entsprechungen" kann in der Datenbank zu weiteren Stellen in demselben Artikel oder in einem anderen Artikel vor- oder zurückgeblättert werden, an denen der Ausdruck "Befehl pr" behandelt wird. Der Suchausdruck "Befehl pr" wird im Text kursiv hervorgehoben.

Hinweis: Wird im Fenster mit den Übereinstimmungen eine Verbindung ausgewählt, wird die Funktion "Entsprechungen" durch die Funktion "Pfad" ersetzt. Hierdurch wird angegeben, daß die Suchfunktion verlassen wurde. Das Fenster "Titelliste" bleibt jedoch geöffnet und ermöglicht den Neubeginn der Suche nach Übereinstimmungen, bis das Fenster manuell geschlossen wird.

8. Zum Schließen des Fensters "Titelliste" den Knopf "Schließen" am oberen Fensterrand anklicken.
9. Um die Suche zu beenden, den Knopf "Abbruch" am unteren Rand des Fensters "Einfache Suche" anklicken.

Werden viele Übereinstimmungen gefunden, ist es unter Umständen schwierig, die gewünschten Informationen zu finden. Es ist möglicherweise erforderlich, eine komplexe Suche durchführen, oder mehrere Artikel durchzulesen, um die gewünschten Informationen zu finden. Bei der komplexen Suche nur die Datenbanken angeben, die diese Informationen am wahrscheinlichsten enthalten. Die Suche kann auch auf Artikeltitel begrenzt werden, die den Suchbegriff enthalten. Weitere Informationen zur komplexen Suche befinden sich im Abschnitt "Komplexe Suche" und im Hilfetext zum Fenster "Komplexe Suche" im *InfoExplorer Benutzerhandbuch*.

Hilfetext aufrufen

In den InfoExplorer-Fenstern kann Hilfetext zu den Funktionen und Optionen aufgerufen werden, die im entsprechenden Fenster angezeigt werden. Außerdem kann in einer Liste nach Themen gesucht werden, zu denen Hilfetext verfügbar ist.

Hilfetext zu einem bestimmten Fenster aufrufen

Es gibt verschiedene InfoExplorer-Fenstertypen, die jeweils einen bestimmten Zweck erfüllen und verschiedene Funktionen enthalten. Das Menü "Hilfe" in jedem Fenster enthält eine Option **Dieses Fenster**, mit der ein Artikel aufgerufen wird, in dem die Funktionen, Menüs und Knöpfe im aktuellen Fenster beschrieben werden.

Soll Hilfetext zu einem Fenster angezeigt werden, wie folgt, vorgehen:

1. Das Menü "Hilfe" aufrufen. Hierzu "Hilfe" in der Menüleiste mit der linken Maustaste anklicken.
2. Die Option **Dieses Fenster** mit der linken Maustaste anklicken. Der Artikel mit Hilfetext wird in einem neuen Lesefenster angezeigt.

Bevor mit dem nächsten Abschnitt weitergearbeitet wird, zur Übung den Hilfetext für die Fenster aufrufen, die gegenwärtig geöffnet sind.

Hilfetext über die Hilfeliste aufrufen

Das Menü "Hilfe" in jedem InfoExplorer-Fenster enthält eine Option **Hilfeliste**, mit der die Liste der InfoExplorer-Themen aufgerufen wird. Die Hilfetextartikel enthalten die erforderlichen Informationen und Prozeduren für zahlreiche Aufgaben.

Folgendermaßen vorgehen, um die Hilfeliste aufzurufen und zu verwenden:

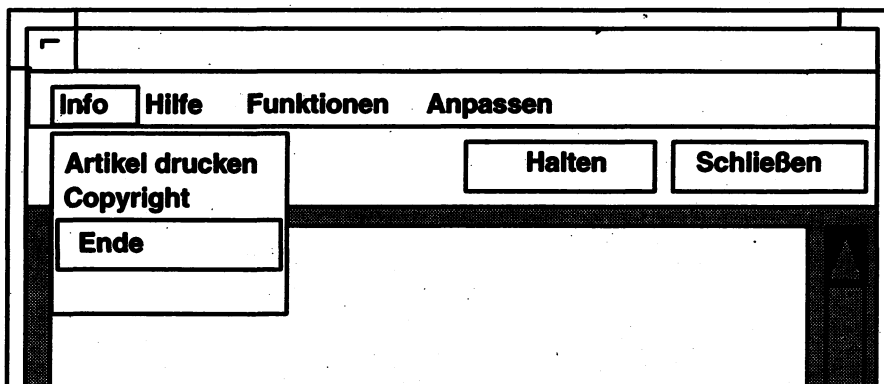
1. Das Menü "Hilfe" aufrufen. Hierzu "Hilfe" in der Menüleiste mit der linken Maustaste anklicken.
2. Die Hilfeliste durch Auswahl der Option **Hilfeliste** aufrufen.
3. Die Verbindung zu dem gewünschten Hilfethema aktivieren.

Bevor mit dem nächsten Abschnitt weitergearbeitet wird, in dem InfoExplorer verlassen wird, zur Übung einige Themen aus der **Hilfeliste** auswählen. Den Hilfetext durchlesen und die beschriebenen Funktionen üben.

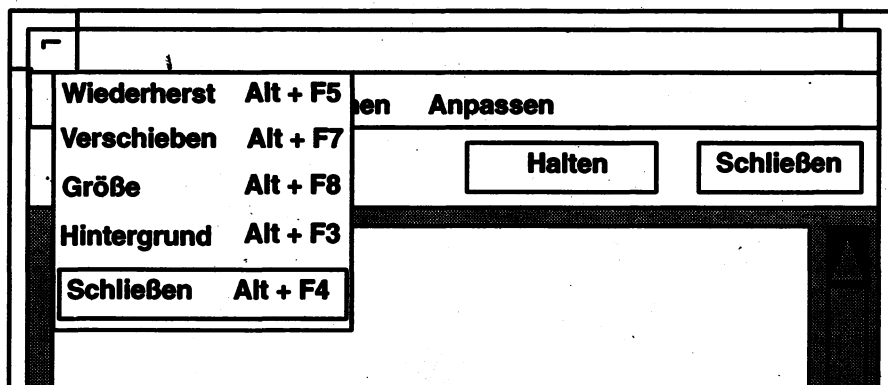
InfoExplorer verlassen

InfoExplorer kann über das Navigationsfenster oder das Lesefenster verlassen werden.

Um InfoExplorer zu verlassen und alle Lesezeichen und Notizen zu sichern, die Option **Ende** im Menü Info auswählen. Alle geöffneten Fenster werden geschlossen, das Programm wird verlassen.



Um InfoExplorer ohne Sichern aller Lesezeichen und Notizen zu verlassen, **Schließen** im Fenstermenü links von der Titelliste auswählen. Alle geöffneten Fenster werden geschlossen, das Programm wird verlassen.



InfoExplorer-Funktionen selbständig erarbeiten

Dieser Artikel enthält die wichtigsten Prozeduren, die zu Beginn der Arbeit mit der InfoExplorer-Datenbank benötigt werden. Mit der Suchfunktion kann z. B. eine beliebige Zeichenfolge in der Datenbank gesucht werden. In primären Navigationspfaden sind die Informationen nach Aufgabe, alphabetisch und nach Büchern geordnet. Durch die Hypertextverbindungen wird die Navigation in der Datenbank wesentlich vereinfacht.

Mit Hilfe dieser Funktionen können nicht nur Informationen zum Betriebssystem und anderer Software abgerufen werden. Darüber hinaus können Kenntnisse über InfoExplorer selbst erworben werden. Insbesondere die über das Aktionsfenster "Hilfe" abrufbare Hilfeliste enthält detaillierte Informationen zu vielen Prozeduren, einschließlich der folgenden:

- Ändern des Standardnavigationsartikels.
- Erstellen und Sichern privater Notizen.
- Erstellen und Sichern von Lesezeichen, mit denen Positionen in der Datenbank markiert werden können.
- Erstellen eines Antwortbriefs.

Der Benutzer kann Informationen zu vielen InfoExplorer-Funktionen nicht nur über die Hilfeliste erhalten, sondern auch durch Blättern in den Menüs der Menüleiste.

Informationen über Befehle anzeigen

Mit dem Befehl **help** kann eine einseitige Informationsanzeige für neue Benutzer aufgerufen werden. Es werden Informationen zu den Befehlen angezeigt, die für folgende Themen verwendet werden:

- Dateien verknüpfen oder anzeigen
- Zeilen interaktiv editieren
- Post senden und empfangen
- Systemnachrichten lesen
- Kennwortdateiinformationen ändern
- Aktuelle Benutzer des Systems identifizieren
- Nachrichten an die anderen Benutzer des System senden
- Inhalt von Verzeichnissen anzeigen
- Informationen zum Quellencodetrollsystem anzeigen
- Datenstationsmodi festlegen

Erforderliche Schritte

Nach Eingabe von **help** erscheint folgende Anzeige:

Befehle:

```
man -k Schlüsselwort    Befehle zu einem Schlüsselwort
auflisten
man Befehl              Dokumentationsseiten zu einem Befehl
                        ausgeben;
```

Andere Basisbefehle sind:

```
cat      -Dateien verknüpfen (und ausgeben)
ex       -Texteditor
finger   -Nachschlageverzeichnis für Benutzerinformationen
ls       -Inhalt eines Verzeichnisses auflisten
mail     -Post senden und empfangen
msgs    -Systemnachrichten und Post
passwd   -Anmeldekennwort ändern
sccshelp -Informationen zum Quellencodetrollsystem
        anzeigen
tset     -Datenstationsmodi festlegen
who      -Benutzer am System
write    -An einen anderen Benutzer schreiben
```

Empfohlene Dokumentation

Weitere Informationen zu den angezeigten Befehlen können der *Befehlsreferenz* entnommen werden.

Elektronische Handbuchseiten (Manual Pages) anzeigen

Mit dem Befehl **man** werden Informationen zu verschiedenen Referenzartikeln wie Befehlen, Unterroutinen und Dateien angezeigt.

Beispiel

Zum Anzeigen von Informationen zu einem Befehl folgendes eingeben:

```
man Befehlsname
```

Sollen z. B. Informationen zum Befehl **grep** angezeigt werden, folgendes eingeben:

```
man grep
```

Daraufhin werden Informationen zum Befehl **grep** am Bildschirm angezeigt.

Empfohlene Dokumentation

Befehle:

apropos, catman, more, whatis, whereis

Dateien:

/usr/bin/man Befehl **man**

/usr/share/man Standardmäßige Handbuchverzeichnisstruktur.

/usr/share/man/cat?/* Verzeichnis mit vorformatierten Seiten.

/usr/share/man/whatis Suchbegriffdatenbank.

/usr/share/man/man?/* Verzeichnis mit elektronischen Handbuchseiten mit dem Format **nroff**.

Kapitel 9. Editoren und Textformatierung

Dieses Kapitel enthält einen Überblick über Editoren und die Textformatierungsfunktion.

"Überblick über Editoren" beschreibt die drei wichtigsten und zwei verwandte Editoren:

- Der Editor ed
- Der Editor INed
- Der Editor vi
 - Der Editor ex
 - Der Editor edit

Weitere Informationen zu diesen Editoren und deren Benutzung enthält das Handbuch *Editorkonzepte und -Prozeduren*.

In "Überblick über die Textformatierung" werden die Funktionen des Textformatierungssystem beschrieben. Weitere Informationen dazu enthält das Handbuch *Text Formatting Reference* (nicht in deutscher Sprache verfügbar).

Überblick über Editoren

Ein Editor ist ein Programm, mit dem Textdateien, Programmdateien oder Dateien mit anderen Daten erstellt und geändert werden können. Ein Editor verfügt jedoch nicht über die Formatierungs- und Druckfunktionen, die ein Textverarbeitungsprogramm bietet.

Ein Editor wird gestartet, indem ein entsprechender Befehl in der Befehlszeile eingegeben wird (z. B. `vi` für den Editor `vi`). Anschließend können durch Eingabe einer Kombination aus Unterbefehlen und Daten folgende Funktionen ausgeführt werden:

- Dateien erstellen, lesen und schreiben
- Text anzeigen und suchen
- Text hinzufügen, ersetzen und entfernen
- Text verschieben und kopieren
- Betriebssystembefehle ausführen

Das Editieren findet in einem Editierpuffer statt, der gesichert oder gelöscht werden kann.

Editoren

Die wichtigsten Editoren des Betriebssystems heißen:

- `ed`
- `INed`
- `vi`

Zu jedem Editor stehen eine Reihe von spezifischen Unterbefehlen zur Verfügung. Eingabe und Anzeige von Unterbefehlen und Daten erfolgen bei den einzelnen Editoren auf unterschiedliche Weise.

Außerdem kann der Benutzer auf folgende, mit dem Editor `vi` verwandte Editoren zugreifen:

- `ex`
- `edit`

Editor `ed`

Der Editor `ed` ist ein Zeileneditor, der auf allen UNIX-Systemen zur Verfügung steht und der unabhängig von der gewählten Übertragungsgeschwindigkeit auf jeder Datenstation ausgeführt werden kann. Beim Editor `ed` wird ebenso wie beim Editor `vi` zwischen Befehls- und Eingabemodus unterschieden. Mit dem Editor `ed` können jedoch nur vollständige Datenzeilen verschoben und kopiert werden. Mit dem Editor `ed` können folgende Aufgaben ausgeführt werden:

- Sehr große Dateien editieren
- Innerhalb eines Shell-Programms editieren

Der Editor `ed` umfasst:

- Unterbefehle zum Suchen und Ersetzen
- Befehle und Syntax, die die Grundlage für andere Programme (einschließlich Befehle) bilden, z. B. `grep`, `awk`, `sed` und `lex`

Editor `INed`

Der Editor `INed` ist ein Gesamtanzeigeditor, mit dem folgende Funktionen ausgeführt werden können:

- Unterbefehle über Befehlstasten und nicht über die Befehlszeile eingeben
- Mehrere Dateien in mehreren Fenstern in der Bildschirmanzeige editieren
- In Dateien horizontal sowie vertikal blättern
- Datenblöcke sowie Datenzeilen verschieben, kopieren und löschen

Weitere Bestandteile des Editors INed:

- Ein Dateimanager, mit dem folgende Funktionen ausgeführt werden können:
 - Dateien und Verzeichnisse erstellen, ändern, löschen und wiederherstellen
 - Zwischen Dateien und Verzeichnissen wechseln
 - Dateien und Verzeichnisse verschieben und kopieren
 - Zugriff auf Dateien und Verzeichnisse einschränken
- Befehle zum Arbeiten mit strukturierten Dateien und zugehörigen Protokollen
- Befehle zur Textformatierung (in eingeschränktem Rahmen)

Editor vi

Der Editor vi ist ein Gesamtanzeigeditor, der auf zahlreichen UNIX-Systemen zur Verfügung steht und von vielen Programmierern verwendet wird. Der Editor vi verfügt über einen Befehlsmodus zum Eingeben von Unterbefehlen und einen Eingabemodus zum Eingeben von Daten. Mit dem Editor vi können folgende Funktionen ausgeführt werden:

- Datenzeilen teilweise oder vollständig verschieben und kopieren
- Optionen zum Auswählen von Editierpräferenzen setzen

Der Editor vi umfaßt folgende Funktionen:

- Eine Vielzahl von Unterbefehlen
- Einen Zeileneditiermodus

Verwandte Editoren

Die Editoren ex und edit gehören derselben Editorenfamilie an wie der Editor vi. Der Editor ex ist ein Zeileneditor mit umfassenden Funktionen, dessen Anzeigefunktionen an die des Editors vi heranreichen. Der Editor edit stellt eine vereinfachte Version des Editors vi dar und wurde speziell für Neueinsteiger konzipiert.

Empfohlene Dokumentation

Zugehörige Informationen

Die Abschnitte zu den Befehlen **grep**, **awk**, **sed** und **lex** im Handbuch *Befehlsreferenz Endbenutzerbefehle*

Überblick über die Textformatierung

Das Textformatierungssystem bietet Unterstützung für das Drucken und Formatieren einer Vielzahl von Dokumenten wie Geschäftsberichten, technischen Berichten, Journalartikeln, Büchern und Berichten, die für Drucker oder Fotosatzgeräte formatiert sind. Das Textformatierungssystem enthält die Dienstprogramme *nroff* und *troff* sowie zugehörige Befehle und unterstützende Funktionen aus dem Paket *Documenter's Workbench Release* 2.0 für das Betriebssystem. Darüber hinaus enthält es Verbesserungen und zusätzliche Befehle für BSD-4.3-Funktionen.

Das Handbuch *Text Formatting Reference* enthält Informationen über die Befehle, Dateien und Makros, die für das Formatieren von Text-, Daten- und Grafikdateien zur Verfügung stehen.

Empfohlene Dokumentation

Text Formatting Reference.

Kapitel 10. Problemlösung und Hardwareinformationen

Dieses Kapitel bietet einen Überblick über die für Systemoperationen und Systemprobleme verfügbaren Informationen. Außerdem wird die Vorgehensweise bei Anzeige eines Fehlers auf dem Bildschirm beschrieben.

Unter "Problemlösung" werden die im *Handbuch Problemlösung* enthaltenen Informationen aufgeführt.

Unter "Hilfe zu Fehlermeldungen" wird beschrieben, wie eine Fehlermeldung aufgebaut ist und wie Hilfetexte zu Fehlermeldungen aufgerufen werden können.

Unter "Hardwareinformationen" werden die im *Bedienerhandbuch* enthaltenen Informationen aufgeführt.

Problemlösung

Das *Handbuch Problemlösung* (IBM Form SC12-2076) enthält Informationen über das Untersuchen, Bestimmen und Beheben von Systemfehlern. Das Handbuch richtet sich an Benutzer von Datenverarbeitungssystemen, die Probleme mit Hard- oder Softwarebetrieb, dem Betriebssystem oder anderen auf dem System installierten Programmen haben.

Das *Handbuch Problemlösung* enthält folgende Abschnitte:

Allgemeine Symptome

Im Abschnitt "Allgemeine Symptome" wird der Benutzer Schritt für Schritt durch Fehlerbehebungsprozeduren geführt. Dieser Abschnitt enthält Kapitel zu folgenden Themen:

- Fehlerbehebungsprozeduren durchführen
- Codes in der dreistelligen Anzeige
- Fehlerprotokollbericht abrufen
- Systemspeicherauszug abrufen
- Hardwarefehlerbestimmung

Spezifische Fehler

Der Abschnitt "Spezifische Fehler" enthält Informationen zum Beheben folgender Fehler:

- Fehler beim Urladen
- Fehler bei Einheiten
- Fehler bei Datenstationen
- Fehler bei Plattenlaufwerken
- Fehler bei Disketten
- Fehler bei Dateien und beim Speichern von Dateien
- Fehler bei logischen Datenträgern
- Fehler bei Netzen
- Fehler bei der Abrechnungsfunktion

Hilfsprogramme zur Problemlösung

Der Abschnitt "Hilfsprogramme zur Problemlösung" enthält Informationen zu Hilfsprogrammen für das Untersuchen und Beheben von Fehlern:

- Eine Fehlerprotokollierungseinrichtung für das Aufzeichnen von Hardware- und Softwarefehlern, die Informationsnachrichten zum Systembetrieb bereitstellt
- Eine Ablaufverfolgungseinrichtung für das Überwachen von Systemereignissen zum Eingrenzen von Systemfehlern
- Eine Systemspeicherauszugsfunktion für das Generieren einer Datei, die Informationen zum Status des Arbeitsspeichers bei Auftreten eines schwerwiegenden Fehlers enthält
- Ein Befehl zur Softwareüberprüfung, mit dem überprüft werden kann, ob ein installiertes Softwareprogramm (oder Lizenzprogramm) beschädigt ist

- Ein Alert-Mechanismus (Alert-Manager), mit dem die Erfassung von Informationen zu signifikanten Netzproblemen ermöglicht wird
- Ein Nachrichtenindex, in dem Verweise zu Artikeln in Büchern und Online-Artikeln in InfoExplorer enthalten sind, mit deren Hilfe Fehler behoben werden können, die Nachrichtennummern generieren
- Fehlernachrichten, die vom System zusammen mit einer möglichen Ursache und einer Behebungsprozedur für jede Nachricht generiert werden können

Grundlegende Informationen

Die vollständige Dokumentation für einige der bei der Problemlösung verwendeten Befehle ist dem Benutzer bei der Durchführung der Problemlösungs-Tasks behilflich.

HL-Instruktionen beschreiben das Starten des AIX-Wartungsmodus mit einer Diskette, einer CD-ROM, einem 8mm-Bandlaufwerk und anderen Bandlaufwerken.

Ein "Formular zur Fehlererfassung" kann fotokopiert werden, um Informationen beim Durcharbeiten der Problemlösungsschritte aufzuzeichnen.

Hilfe zu Fehlernachrichten

Wird eine Fehlernachricht mit einer vorausgehenden 7stelligen Zahl angezeigt, stehen weitere Informationen zu dem Befehl, dem Systemaufruf oder der Unterroutine, durch die der Fehler generiert wurde, zur Verfügung.

Jede Nachricht ist durch zwei Teile eindeutig gekennzeichnet. Der erste Teil der Nachricht ist die Komponenten-ID. Der zweite Teil ist die Nachrichten-ID. Beispiel: Bei der Nachricht **0001-007** ist die Komponenten-ID **0001** und die Nachrichten-ID **007**.

Informationen zu Fehlernachrichten aufrufen

1. Die 7stellige ID der Fehlernachricht notieren.
2. Im Artikel "Nachrichtenindex" im *Handbuch Problemlösung* nachschlagen. In diesem Abschnitt werden die Fehler-IDs und die zugehörigen Artikel aufgelistet.

ODER

Mit der Suchfunktion in InfoExplorer nach dem Artikel mit dem Titel "Nachrichtenindex" suchen. Mit Hilfe dieses Artikels können Informationen über die Funktion angezeigt werden, durch die die Fehlernachricht generiert wurde. (Mit Hilfe des Aufgabenindex von InfoExplorer kann der Artikel "Nachrichtenindex" ebenfalls gesucht werden.) Nähere Informationen über InfoExplorer können unter "InfoExplorer: Ein neuer Zugriff auf Dokumentation" nachgeschlagen werden.

Hardwareinformationen

Nähere Informationen zur Systemhardware können im *Bedienerhandbuch* der Systemeinheit nachgeschlagen werden. Darin werden der Systembetrieb und die für das System verfügbaren Diagnosefunktionen beschrieben. Das *Bedienerhandbuch* enthält Informationen zu folgenden Themen:

- Beschreibung der Systemeinheit
- Benutzung der Systemeinheit
- Benutzung der Diagnoseprogramme
- Benutzung der Servicehilfen
- Benutzung der Systemtestprozedur
- Hardwarefehlerbestimmung
- Systemeinheit transportieren
- Zubehör
- Zahlen in der dreistelligen Anzeige

Anhang A. Korn-Shell

Bei der Korn-Shell handelt es sich um einen interaktiven Befehls-Interpreter und eine befehlsorientierte Programmiersprache. Sie enthält Funktionen, die auch von der Bourne-Shell und der C-Shell zur Verfügung gestellt werden.

Verwendung der Korn-Shell

Die folgenden Abschnitte skizzieren die Funktionen der Korn-Shell und liefern hilfreiche Informationen über die Verwendung der Korn-Shell:

- Funktionen der Korn-Shell
- Shell-Umgebung (auf Seite A-1)
- Arbeiten mit der Shell (auf Seite A-2)
- Koprozeßeinrichtung (auf Seite A-3)

Funktionen der Korn-Shell

Die Korn-Shell bietet viele Funktionen, die auch in anderen Shells vorhanden sind, z. B. E/A-Umadressierung, Variablensubstitution und Dateinamensubstitution. Sie enthält außerdem einige zusätzliche Befehle und Funktionen der Programmiersprache:

Arithmetische Berechnungen

Die Korn-Shell kann Ganzzahlrechenoperationen (auf der Basis von 2 bis 36) ausführen, indem der integrierte Befehl `let` verwendet wird. Im Abschnitt "Arithmetische Berechnungen in der Korn-Shell" auf Seite A-23 wird diese Funktion näher beschrieben.

Befehlsprotokoll

Die Korn-Shell speichert eine Datei, in der alle eingegebenen Befehle aufgezeichnet werden. Mit einem Texteditor kann ein Befehl in dieser Protokolldatei geändert werden, der dann erneut eingegeben werden kann. Weitere Informationen über die Funktion des Befehlsprotokolls enthält der Abschnitt "Das Befehlsprotokoll der Korn-Shell" auf Seite A-22.

Koprozeßeinrichtung

Mit der Koprozeßeinrichtung kann der Benutzer Programme im Hintergrund ausführen und Informationen zu diesen Hintergrundprozessen senden und von dort empfangen.

Editieren

Mit den Optionen für Befehlszeileneditierung kann der Benutzer die Befehlszeile editieren. Editoren wie der Editor `emacs`, der Editor `gmacs` und der Editor `vi` stehen zur Verfügung.

Shell-Umgebung

Die Umgebung besteht aus einer Liste von Name/Wert-Paaren. Diese Liste wird an ein ausgeführtes Programm übergeben wie eine normale Argumentenliste. Die Namen müssen Kennungen sein und die Werte sind Zeichenfolgen. Die Shell steht auf unterschiedliche Art und Weise mit der Umgebung im Dialog. Ist die Shell aufgerufen, durchsucht sie die Umgebung und erstellt für jeden gefundenen Namen einen Parameter. Diesem Parameter

wird ein entsprechender Wert zugeordnet und er wird zum Exportieren markiert. Ausgeführte Befehle übernehmen die Umgebung.

Wenn der Benutzer die Werte der Shell-Parameter ändert oder neue erstellt, indem er die Befehle **export** oder **typeset -x** verwendet, werden diese Parameter Teil der Umgebung. Von einem ausgeführten Befehl aus gesehen setzt sich die Umgebung demzufolge aus beliebigen Name/Wert-Paaren zusammen, die ursprünglich von der Shell übernommen wurden, deren Werte von der aktuellen Shell verändert werden können, und die durch die Verwendung der Befehle **export** oder **typeset-x** Zusätze erhalten können.

Die Umgebung für jeden einzelnen Befehl oder jede einzelne Funktion wird durch ein Präfix, d. h. ein oder mehrere Parameterzuordnungen, erweitert. Ein Parameterzuordnungsargument ist ein Begriff im Format *Kennung=Wert*. Deshalb sind die zwei folgenden Ausdrücke äquivalent (was die Ausführung des Befehls betrifft):

```
TERM=450 Befehl Argumente
```

```
(export TERM; TERM=450; Befehl Argumente)
```

Wurde die Option **-k** gesetzt, werden *alle* Parameterzuordnungsargumente in die Umgebung gestellt, auch wenn sie hinter dem Befehlsnamen stehen. Mit dem folgenden Befehl wird erst **a=b c** und dann **c** ausgegeben:

```
echo a=b c
set -k
echo a=b c
```

Hinweis: Diese Funktion ist nur für Prozeduren gedacht, die für frühere Versionen der Shell geschrieben wurden. Von der Verwendung in neuen Prozeduren wird dringend abgeraten.

Arbeiten mit der Shell

Wurde die Shell mit dem Befehl **exec** aufgerufen und ist das erste Zeichen des Nullarguments (**\$0**) ein Minuszeichen (**-**), handelt es sich bei der Shell um eine Anmelde-Shell. Die Shell liest zuerst Befehle aus der Datei **/etc/profile** und danach, je nachdem, welche Datei existiert, aus der Datei **.profile** im aktuellen Verzeichnis oder aus der Datei **\$HOME/.profile**. Als nächstes liest die Shell Befehle aus der Datei (falls vorhanden), deren Name durch Parametersubstitution des Wertes der Umgebungsvariablen **ENV** festgelegt wurde.

Wurde beim Aufrufen der Korn-Shell der Parameter *Datei* [*Parameter*] angegeben, führt die Shell die Prozedurdatei aus, die durch den Parameter *Datei* gekennzeichnet wurde, einschließlich der ansonsten angegebenen Parameter. Die angegebene Prozedurdatei muß Lesezugriff besitzen; jegliche Einstellungen für **setuid** und **setgid** (Definitionen für Benutzer- bzw. Gruppen-ID) werden ignoriert. Die Shell liest dann die Befehle.

Hinweis: Bei der Angabe der Prozedurdatei nicht die Optionen **-c** oder **-s** verwenden.

Weitere Informationen zum Positionsparameter befinden sich im Abschnitt "Parametersubstitution in der Korn-Shell" auf Seite A-28.

Shell-Eingabeaufforderung

Im interaktiven Modus erscheint als Shell-Eingabeaufforderung der Wert des Parameters **PS1**, bevor ein Befehl gelesen wird. Wird zu einer beliebigen Zeit eine neue Zeile eingegeben und benötigt die Shell weitere Eingaben, um den Befehl zu vervollständigen, wird eine zweite Eingabeaufforderung ausgegeben (der Wert des Parameters **PS2**).

Signalverarbeitung

Die Signale **INT** und **QUIT** für einen aufgerufenen Befehl werden ignoriert, wenn auf den Befehl ein **&** (Et-Zeichen) folgt und die Jobüberwachungsoption **monitor** nicht aktiv ist. Ansonsten haben Signale die Werte, die die Shell von den Vätern übernommen hat.

Ausführungsstatus

Wird von der Shell ein Fehler, z. B. ein Syntaxfehler festgestellt, wird ein Ausführungsstatus ungleich Null zurückgegeben. Ansonsten gibt die Shell den Ausführungsstatus des letzten ausgeführten Befehls zurück. Wird die Shell nicht interaktiv verwendet, wird die Ausführung der Shell-Datei abgebrochen. Die Shell meldet Fehler bei der Ausführung, indem sie den Befehls- oder Funktionsnamen und die Fehlerbedingung ausgibt. Ist die Nummer der Zeile, in der ein Fehler auftrat, größer als 1, wird die Zeilennummer in [] (eckigen Klammern) nach dem Befehls- oder Funktionsnamen ausgegeben.

Hinweis: Unterbrechungen werden nicht verarbeitet, während ein Job auf einen Vordergrundprozeß wartet. Somit wird eine Unterbrechung bei dem Signal **CHLD** erst ausgeführt, wenn der Vordergrundprozeß beendet ist.

Koprozeßeinrichtung

In der Korn-Shell ist es möglich, einen oder mehrere Befehle als Hintergrundprozeß auszuführen. Diese Befehle, die innerhalb einer Shell-Prozedur ausgeführt werden, heißen Koprozesse. Koprozesse sind bei der Kommunikation mit einem Programm nützlich.

Ein Koprozeß wird dadurch bestimmt, daß der Operator **|&** (Balken, Et-Zeichen) hinter den Befehl gestellt wird. Sowohl die Standardeingabe als auch die -ausgabe des Befehls werden über eine Pipe zur Prozedur geleitet.

Ein Koprozeß muß die folgenden Bedingungen erfüllen:

- ein Zeilenvorschubzeichen am Ende jeder Nachricht beinhalten
- jede Ausgabenachricht zur Standardausgabe senden
- den Inhalt der Standardausgabe nach jeder Nachricht löschen.

Das folgende Beispiel zeigt, wie die Eingabe an einen Koprozeß übergeben und von ihm zurückgegeben wird:

```
echo "Erster Prozeß"
./FileB.sh |&
read -p a b c d
echo "Aus dem Koprozeß lesen: $a $b $c $d"
print -p "Dem Koprozeß übergeben"
read -p a b c d
echo "Vom Koprozeß zurückgegeben: $a $b $c $d"
```

```
FileB.sh
    echo "Der Koprozeß wird ausgeführt"
    read a b c d
    echo $a $b $c $d
```

Das Ergebnis ist folgende Standardausgabe:

```
Erster Prozeß
Aus dem Koprozeß lesen: Der Koprozeß wird ausgeführt
Vom Koprozeß zurückgegeben: Dem Koprozeß übergeben
```

Mit dem Befehl **print -p** kann in den Koprozeß geschrieben werden. Um aus dem Koprozeß zu lesen, den Befehl **read -p** eingeben.

Koprozeßein- und -ausgabe umleiten

Die Standardein- und -ausgabe eines Koprozesses wird einem numerierten Dateideskriptor über die E/A-Umadressierung neu zugeordnet. Beispielsweise leitet der Befehl:

```
exec 5>$p
```

die Eingabe vom Koprozeß zum Dateideskriptor 5.

Danach kann die Standardumadressierungssyntax verwendet werden, um die Ausgabe des Befehls zum Koprozeß umzuleiten. Es kann auch ein weiterer Koprozeß aufgerufen werden. Die Ausgabe von beiden Koprozessen ist mit derselben Pipe verbunden und wird mit dem Befehl **read -p** gelesen. Um den Koprozeß zu beenden, folgendes eingeben:

```
read -u5
```

Weitere Informationen zur E/A-Umadressierung enthält der Abschnitt "Ein- und Ausgabe in der Korn-Shell umleiten" auf Seite A-35.

Verwendung von Korn-Shell-Befehlen

Im folgenden Abschnitt werden die Korn-Shell-Befehle erläutert und wie die Shell Befehle und Zeichen interpretiert:

- Befehlsinterpretation in der Korn-Shell
- Quotierungszeichen (auf Seite A-6)
- Zusammengesetzte Befehle in der Korn-Shell (auf Seite A-7)
- Reservierte Wörter (auf Seite A-8)

Befehlsinterpretation in der Korn-Shell

Wird in der Korn-Shell ein Befehl eingegeben, wertet die Shell den Befehl aus und reagiert wie folgt:

- Führt alle angezeigten Substitutionen durch.
- Stellt fest, ob der Befehl einen / (Schrägstrich) enthält. Ist das der Fall, führt die Shell das Programm aus, das im Pfadnamen angegeben ist.

Enthält der Befehl keinen / (Schrägstrich), führt die Korn-Shell folgende Aktionen aus:

- Stellt fest, ob der Befehl ein Sonderbefehl der Korn-Shell ist. Ist das der Fall, führt die Shell den Befehl innerhalb des aktuellen Shell-Prozesses aus.
- Vergleicht den Befehl mit benutzerdefinierten Funktionen. Stimmt der Befehl mit einer benutzerdefinierten Funktion überein, werden die Positionsparameter gesichert und dann in den Argumenten des **Funktionsaufrufs** zurückgesetzt. Ist die Funktion beendet oder wird eine Meldung zurückgegeben, wird die Liste der Positionsparameter zurückgeschrieben und jede Hardwareprogrammverzweigung, die innerhalb der Funktion auf **EXIT** gesetzt ist, wird ausgeführt. Der Wert einer Funktion ist der Wert des letzten ausgeführten Befehls. Eine Funktion wird im aktuellen Shell-Prozeß ausgeführt.
- Erstellt einen Prozeß und versucht, den Befehl durch Verwendung des Befehls **exec** auszuführen (falls es sich bei dem Befehl weder um einen integrierten Befehl noch um eine benutzerdefinierte Funktion handelt).

Die Korn-Shell durchsucht jedes Verzeichnis in einem bestimmten Pfad nach einer ausführbaren Datei. Die Shell-Variable **PATH** definiert den Suchpfad für das Verzeichnis, das den Befehl enthält. Alternative Verzeichnisnamen werden durch einen : (Doppelpunkt) abgetrennt. Der Standardpfad ist **/usr/bin**: (wodurch erst das Verzeichnis **/usr/bin** angegeben wird und dann das aktuelle Verzeichnis). Das aktuelle Verzeichnis wird durch zwei oder mehr Doppelpunkte nebeneinander angegeben oder durch einen Doppelpunkt am Anfang oder Ende der Pfadliste.

Verfügt die Datei über eine Ausführungsberechtigung, ist aber weder ein Verzeichnis noch eine **a.out**-Datei, nimmt die Shell an, daß die Datei Shell-Befehle enthält. Der aktuelle Shell-Prozeß erzeugt eine Subshell, um die Datei zu lesen. Alle nicht exportierten Alias-Namen, Funktionen und benannten Parameter werden aus der Datei gelöscht. Verfügt die Shell-Befehlsdatei über Leseberechtigung oder sind die Definitionsbits für **Benutzer-ID** oder **Gruppen-ID** für die Datei gesetzt, führt die Shell einen Agenten aus, der die Berechtigungen setzt und die Shell ausführt, wobei die Shell-Befehlsdatei als offene Datei übergeben wird. Ein Klammerbefehl wird in einer Subshell ausgeführt, ohne die nicht exportierten Bestände zu löschen.

Quotierungszeichen

Soll die Korn-Shell ein Zeichen als normales Zeichen lesen, ohne eine ihm zugeordnete Bedeutung, muß es in Quotierungszeichen gesetzt werden. Soll die Sonderbedeutung eines Metazeichens aufgehoben werden, muß ihm ein \ (umgekehrter Schrägstrich) vorangestellt werden.

Jedes Metazeichen hat für die Shell eine Sonderbedeutung und führt, wenn es nicht in Quotierungszeichen gesetzt ist, zum Abschluß eines Begriffs. Folgende Zeichen werden von der Korn-Shell als Metazeichen interpretiert:

- | (Balken)
- & (Et-Zeichen)
- ; (Semikolon)
- < (Kleiner-als-Zeichen) und > (Größer-als-Zeichen)
- ((linke runde Klammer) und) (rechte runde Klammer)
- \$ (Dollarzeichen)
- ' (linkes halbes Anführungszeichen) und ' (rechtes halbes Anführungszeichen)
- \ (umgekehrter Schrägstrich)
- " (Anführungszeichen)
- Zeilenvorschubzeichen
- Leerzeichen
- Tabulatorzeichen.

Folgende Bedingungen gelten für Metazeichen und Quotierungszeichen in der Korn-Shell:

- Die Korn-Shell ignoriert die Kombination der Metazeichen \ (umgekehrter Schrägstrich) und Zeilenvorschubzeichen.
- Alle Zeichen zwischen halben Anführungszeichen (' ') gelten als Zeichen in Quotierungszeichen.
- Innerhalb von halben Anführungszeichen darf kein weiteres halbes Anführungszeichen stehen.
- Die Bedeutung von \$* (Dollarzeichen, Stern) und \$@ (Dollarzeichen, kommerzielles A) ist gleich (wenn sie nicht in Quotierungszeichen gesetzt sind) bei der Verwendung als Parameterzuordnungszeichen oder Dateiname.
- Bei der Verwendung als Befehlsargument entspricht "\$*" (Anführungszeichen, Dollarzeichen, Stern, Anführungszeichen) "\$1 \$2...", wobei *d* das erste Zeichen des Parameters IFS ist.
- "\$@" (Anführungszeichen, Dollarzeichen, kommerzielles A, Anführungszeichen) entspricht "\$1" "\$2"
- Innerhalb von Gravis (`) werden durch den umgekehrten Schrägstrich die Zeichen \ (umgekehrter Schrägstrich), ' (halbes Anführungszeichen) und \$ (Dollarzeichen) in Quotierungszeichen gesetzt. Stehen die Gravis innerhalb von " " (Anführungszeichen), werden durch den umgekehrten Schrägstrich auch Anführungszeichen in Quotierungszeichen gesetzt.
- Parameter- und Befehlssubstitution erfolgt innerhalb von " " (Anführungszeichen).
- Die Sonderbedeutung von reservierten Wörtern oder Alias-Namen wird aufgehoben, wenn ein beliebiges Zeichen des reservierten Wortes in Quotierungszeichen gesetzt wird. Funktionsnamen und Namen von Sonderbefehlen können nicht in Quotierungszeichen gesetzt werden.

Zusammengesetzte Befehle in der Korn-Shell

Folgende zusammengesetzte Befehle gibt es in der Korn-Shell:

- Pipeline (ein oder mehrere einfache Befehle, die durch das Symbol | getrennt sind)
- Liste von einfachen Befehlen
- Befehl, der mit einem reservierten Wort beginnt
- Befehl, der mit dem Steueroperator (linke runde Klammer) beginnt.

Falls nicht anders angegeben, ist der Wert, der von einem zusammengesetzten Befehl zurückgegeben wird, der des letzten ausgeführten einfachen Befehls.

Liste der zusammengesetzten Befehle

for *Kennung* [*in Wort ...*] ;**do** *Liste* ;**done**

Bei jeder Ausführung des Befehls **for** wird der Parameter *Kennung* auf das nächste Wort aus der Liste *in Wort...* gesetzt. Wird der Befehl *in Wort ...* übergangen, führt der Befehl **for** die Liste **do** für jeden gesetzten Positionsparameter aus. Die Ausführung ist beendet, wenn die Liste keine weiteren Wörter enthält. Der Abschnitt "Parametersubstitution in der Korn-Shell" auf Seite A-28 enthält weitere Informationen über Positionsparameter.

select *Kennung* [*in Wort ...*] ;**do** *Liste* ;**done**

Beim Befehl **select** erfolgt die Ausgabe der angegebenen Wortgruppe in die Standarddatei für Fehlerausgabe (Dateideskriptor 2). Jedem Wort wird hierbei eine Zahl vorangestellt. Wird der Befehl *in Wort ...* übergangen, werden stattdessen die Positionsparameter verwendet. Die Eingabeaufforderung **PS3** wird ausgegeben und eine Zeile von der Standardeingabe gelesen. Besteht diese Zeile aus der Nummer eines der aufgelisteten Wörter, wird der Wert des Parameters *Kennung* auf das Wort gesetzt, das dieser Nummer entspricht.

Ist die Zeile, die von der Standardeingabe gelesen wird, leer, wird die Auswahlliste erneut ausgegeben. Ansonsten wird der Wert des Parameters *Kennung* auf Null gesetzt. Der Inhalt der Zeile, die von der Standardeingabe gelesen wird, wird im Parameter **REPLY** gespeichert. Der Parameter *Liste* wird für jede Auswahl ausgeführt, bis ein Unterbrechungszeichen oder ein Dateiendezeichen auftritt. Der Abschnitt "Parametersubstitution in der Korn-Shell" auf Seite A-28 enthält weitere Informationen über Positionsparameter.

case *Wort* **in** [(*Muster* [| *Muster*] ...) *Liste* ;;] ... **esac**

Der Befehl **case** führt den Parameter *Liste* aus, der dem ersten Parameter *Muster* zugeordnet ist, der mit dem Parameter *Wort* übereinstimmt. Das Format der Muster ist das gleiche, das auch zur Dateinamenssubstitution verwendet wird.

if *Liste* ;**then** *Liste* [**elif** *Liste* ;**then** *Liste*] ... [**else** *Liste*] ;**fi**

Der Parameter *Liste* gibt die Liste der Befehle an, die ausgeführt werden sollen. Die Shell führt zuerst den Befehl **if** *Liste* aus. Wird der Ausführungsstatus Null zurückgegeben, führt sie den Befehl **then** *Liste* aus. Ansonsten werden die Befehle, die im Parameter *Liste*, der auf dem Befehl **elif** folgt, ausgeführt.

Ist der Wert, der vom letzten Befehl in der Liste des Befehls **elif** *Liste* zurückgegeben wurde, gleich Null, wird der Befehl **then** *Liste* ausgeführt. Ist der Wert, der vom letzten Befehl in der Liste des Befehls **then** *Liste*

zurückgegeben wurde, gleich Null, wird der Befehl **else** *Liste* ausgeführt. Werden keine in den Parametern *Liste* angegebenen Befehle für die Befehle **else** oder **then** ausgeführt, gibt der Befehl **if** den Ausführungsstatus Null zurück.

while *Liste* ;**do** *Liste* ;**done**
until *Liste* ;**do** *Liste* ;**done**

Der Parameter *Liste* gibt die Liste der Befehle an, die ausgeführt werden sollen. Der Befehl **while** führt wiederholt die Befehle aus, die im Parameter *Liste* angegeben sind. Ist der Ausführungsstatus des letzten Befehls in der Liste des Befehls **while** *Liste* gleich Null, wird der Befehl **do** *Liste* ausgeführt. Ist der Ausführungsstatus nicht Null, wird die Schleife beendet. Werden keine Befehle der Liste des Befehls **do** *Liste* ausgeführt, gibt der Befehl **while** den Ausführungsstatus Null zurück. Anstelle des Befehls **while** kann der Befehl **until** verwendet werden, um den Schleifenabschlußtest aufzuheben.

(*Liste*)

Der Parameter *Liste* gibt die Liste der Befehle an, die ausgeführt werden sollen. Die Shell führt den Parameter *Liste* in einer separaten Umgebung aus.

Hinweis: Sind für die Verschachtelung zwei linke runde Klammern erforderlich, muß dazwischen ein Leerzeichen eingefügt werden, um zwischen dem Befehl und der arithmetischen Berechnung zu unterscheiden.

{*Liste*;}

Der Parameter *Liste* gibt die Liste der Befehle an, die ausgeführt werden sollen. Der Parameter *Liste* wird einfach ausgeführt.

Hinweis: Anders als bei den Metazeichen () (runde Klammern), werden durch { } (geschweifte Klammern) reservierte Wörter gekennzeichnet (für spezielle Zwecke verwendet, nicht als benutzerdefinierte Kennungen). Um erkannt zu werden, müssen diese reservierten Wörter am Anfang einer Zeile stehen oder nach einem ; (Semikolon).

[[*Ausdruck*]]

Wertet den Parameter *Ausdruck* aus. Ist der Ausdruck wahr, wird der Ausführungsstatus Null zurückgegeben.

function *Kennung* {*Liste* ;}
Kennung () {*Liste* ;}

Definiert eine Funktion, auf die durch den Parameter *Kennung* verwiesen wird. Der Hauptteil der Funktion ist die angegebene Liste von Befehlen, die durch die { } (geschweiften Klammern) eingeschlossen ist.

time *Pipeline*

Führt den Parameter *Pipeline* aus. Die abgelaufene Zeit, die Benutzerzeit und die Systemzeit werden in die Standarddatei für Fehlerausgabe ausgegeben.

Reservierte Wörter

Die folgenden reservierten Wörter werden nur erkannt, wenn sie ohne Quotierungszeichen als erstes Wort eines Befehls erscheinen:

case	do	done
elif	else	esac
fi	for	function
if	in	select
then	time	until
while	{ }	[[]]

Funktionen

Das reservierte Wort **function** definiert Shell-Funktionen. Die Shell liest und speichert Funktionen intern. Alias-Namen werden beim Lesen der Funktion aufgelöst. Die Shell führt Funktionen auf dieselbe Weise aus wie Befehle, wobei die Argumente als Positionsparameter übergeben werden. Der Abschnitt "Parametersubstitution in der Korn-Shell" auf Seite A-28 enthält weitere Informationen über Positionsparameter.

Funktionen werden in demselben Prozeß ausgeführt wie das aufrufende Programm und benutzen gemeinsam mit dem aufrufenden Programm Dateien, Hardwareprogrammverzweigungen (nicht **EXIT** und **ERR**) und das aktuelle Arbeitsverzeichnis. Hardwareprogrammverzweigungen, auf die das aufrufende Programm trifft, werden innerhalb der Funktion auf ihre Standardaktion zurückgesetzt. Eine Bedingung der Hardwareprogrammverzweigung, auf die die Funktion nicht trifft oder sie ignoriert, führt dazu, daß die Funktion beendet wird und die Bedingung dem aufrufenden Programm übergeben wird.

Eine Hardwareprogrammverzweigung auf **EXIT** innerhalb einer Funktion wird ausgeführt, nachdem die Funktion in der Umgebung des aufrufenden Programms beendet ist. Normalerweise werden Variablen vom aufrufenden Programm und der Funktion gemeinsam benutzt. Der Sonderbefehl **typeset** jedoch definiert, wenn er innerhalb einer Funktion verwendet wird, lokale Variablen, deren Bereich die aktuelle Funktion und alle Funktionen, die von ihr aufgerufen werden, umfaßt.

Der Sonderbefehl **return** wird verwendet, um von Funktionsaufrufen zurückzukehren. Fehler innerhalb von Funktionen geben die Steuerung an das aufrufende Programm zurück.

Funktionsangaben können mit der Option **-f** oder **+f** des Sonderbefehls **typeset** aufgelistet werden. Die Option **-f** listet auch den Text von Funktionen auf. Funktionen werden mit der Option **-f** des Sonderbefehls **unset** aufgehoben.

Normalerweise werden Funktionen inaktiviert, wenn die Shell eine Shell-Prozedur ausführt. Mit der Option **-xf** des Sonderbefehls **typeset** kann eine Funktion in Prozeduren exportiert werden, die ohne einen separaten Aufruf der Shell ausgeführt werden. Funktionen, die über einen separaten Aufruf der Shell definiert werden müssen, sollten in der Datei **ENV** angegeben werden mit der Option **-xf** des Sonderbefehls **typeset**.

Bedingungsausdrücke

Ein Bedingungsausdruck wird mit dem zusammengesetzten Befehl **[[** (doppelte eckige Klammern) verwendet, um Dateiattribute zu testen und Zeichenfolgen zu vergleichen. Bei Wörtern zwischen **[[** und **]]** (doppelte eckige Klammern) erfolgt keine Silbentrennung und Dateinamenssubstitution. Jeder Ausdruck wird aus einem oder mehreren der folgenden monadischen oder Binärausdrücke gebildet:

- a Datei** Wahr, wenn die angegebene Datei existiert.
- b Datei** Wahr, wenn die angegebene Datei existiert und es sich dabei um eine blockorientierte Gerätedatei handelt.
- c Datei** Wahr, wenn die angegebene Datei existiert und es sich dabei um eine zeichenorientierte Gerätedatei handelt.
- d Datei** Wahr, wenn die angegebene Datei existiert und es sich dabei um ein Verzeichnis handelt.
- f Datei** Wahr, wenn die angegebene Datei existiert und es sich dabei um eine gewöhnliche Datei handelt.
- g Datei** Wahr, wenn die angegebene Datei existiert und ihr Definitionsbit für Gruppen-ID gesetzt ist.

- k Datei** Wahr, wenn die angegebene Datei existiert und ihr Sticky Bit gesetzt ist.
- n Zeichenfolge** Wahr, wenn die Länge der angegebenen Zeichenfolge ungleich Null ist.
- o Option** Wahr, wenn die angegebene Option eingestellt ist.
- p Datei** Wahr, wenn die angegebene Datei existiert und es sich dabei um eine FIFO-Geräte-datei oder eine Pipe handelt.
- r Datei** Wahr, wenn die angegebene Datei existiert und vom aktuellen Prozeß gelesen werden kann.
- s Datei** Wahr, wenn die angegebene Datei existiert und größer als Null ist.
- t Dateideskriptor** Wahr, wenn die angegebene Dateideskriptornummer geöffnet und einer Datenstationseinheit zugeordnet ist.
- u Datei** Wahr, wenn die angegebene Datei existiert und ihr Definitionsbit für Benutzer-ID gesetzt ist.
- w Datei** Wahr, wenn die angegebene Datei existiert und vom aktuellen Prozeß nicht geschrieben werden kann.
- x Datei** Wahr, wenn die angegebene Datei existiert und vom aktuellen Prozeß ausgeführt werden kann. Wenn die angegebene Datei existiert und es sich dabei um ein Verzeichnis handelt, verfügt der aktuelle Prozeß über die Berechtigung, im Verzeichnis zu suchen.
- z Zeichenfolge** Wahr, wenn die Länge der angegebenen Zeichenfolge gleich Null ist.
- L Datei** Wahr, wenn die angegebene Datei existiert und es sich dabei um eine symbolische Verbindung handelt.
- O Datei** Wahr, wenn die angegebene Datei existiert und der Eigner die aktuelle Benutzer-ID dieses Prozesses ist.
- G Datei** Wahr, wenn die angegebene Datei existiert und ihre Gruppe mit der aktuellen Gruppen-ID dieses Prozesses übereinstimmt.
- S Datei** Wahr, wenn die angegebene Datei existiert und es sich dabei um ein Socket handelt.
- Datei1 -nt Datei2** Wahr, wenn *Datei1* existiert und neuer ist als *Datei2*.
- Datei1 -ot Datei2** Wahr, wenn *Datei1* existiert und älter ist als *Datei2*.
- Datei1 -ef Datei2** Wahr, wenn *Datei1* und *Datei2* existieren und sich auf dieselbe Datei beziehen.
- Zeichenfolge = Muster** Wahr, wenn die angegebene Zeichenfolge mit dem angegebenen Muster übereinstimmt.
- Zeichenfolge != Muster** Wahr, wenn die angegebene Zeichenfolge nicht mit dem angegebenen Muster übereinstimmt.

Zeichenfolge1 < *Zeichenfolge2*

Wahr, wenn ausgehend vom ASCII-Wert ihrer Zeichen *Zeichenfolge1* vor *Zeichenfolge2* steht.

Zeichenfolge1 > *Zeichenfolge2*

Wahr, wenn ausgehend vom ASCII-Wert ihrer Zeichen *Zeichenfolge1* hinter *Zeichenfolge2* steht.

Ausdruck1 **-eq** *Ausdruck2*

Wahr, wenn *Ausdruck1* gleich *Ausdruck2* ist.

Ausdruck1 **-ne** *Ausdruck2*

Wahr, wenn *Ausdruck1* ungleich *Ausdruck2* ist.

Ausdruck1 **-lt** *Ausdruck2*

Wahr, wenn *Ausdruck1* kleiner als *Ausdruck2* ist.

Ausdruck1 **-gt** *Ausdruck2*

Wahr, wenn *Ausdruck1* größer als *Ausdruck2* ist.

Ausdruck1 **-le** *Ausdruck2*

Wahr, wenn *Ausdruck1* kleiner-gleich *Ausdruck2* ist.

Ausdruck1 **-ge** *Ausdruck2*

Wahr, wenn *Ausdruck1* größer-gleich *Ausdruck2* ist.

Hinweis: Für jeden der vorangegangenen Ausdrücke gilt: Ist die Variable *Datei* gleich */dev/fd/n*, wobei *n* eine ganze Zahl ist, wird ein Test an der offenen Datei ausgeführt, deren Deskriptornummer *n* ist.

Aus diesen Basiselementen können zusammengesetzte Ausdrücke gebildet werden, wenn einer der folgenden Ausdrücke, nach Vorrangstellung in absteigenden Reihenfolge aufgelistet, verwendet wird:

(Ausdruck) Wahr, wenn der angegebene Ausdruck wahr ist. Wird verwendet, um Ausdrücke zu gruppieren.

! Ausdruck Wahr, wenn der angegebene Ausdruck falsch ist.

Ausdruck1 && Ausdruck2

Wahr, wenn sowohl *Ausdruck1* als auch *Ausdruck2* wahr sind.

Ausdruck1 || Ausdruck2

Wahr, wenn entweder *Ausdruck1* oder *Ausdruck2* wahr ist.

Sonderbefehle in der Korn-Shell

Sonderbefehle sind in die Korn-Shell integriert und werden im Shell-Prozeß ausgeführt. Wenn nicht anders angegeben, wird die Ausgabe in den Dateideskriptor 1 geschrieben und ist der Ausführungsstatus Null, es sei denn, der Befehl enthält Syntaxfehler. Ein- und Ausgabeumadressierung ist erlaubt.

Die folgenden Sonderbefehle werden etwas anders verarbeitet als andere Sonderbefehle:

:	continue	export	times
.	eval	readonly	trap
alias	exec	return	typeset
break	exit	shift	wait

Die Korn-Shell verarbeitet diese Befehle wie folgt:

- Parameterzuordnungslisten, die dem Befehl vorangehen, bleiben gültig, wenn der Befehl beendet ist.
- E/A-Umadressierungen werden nach der Parameterzuordnung verarbeitet.
- Fehler in der Shell-Prozedur führen dazu, daß die Prozedur die Verarbeitung stoppt.

Zusätzlich werden Wörter, die im Format einer Parameterzuordnung den Sonderbefehlen **alias**, **export**, **readonly** und **typeset** folgen, mit denselben Regeln erweitert wie Parameterzuordnungen. Das bedeutet, daß die Tildensubstitution nach dem = (Gleichheitszeichen) ausgeführt wird, Silbentrennung und Dateinamenssubstitution jedoch nicht.

Liste der Sonderbefehle

Die folgenden Befehle sind in die Korn-Shell integriert:

: [*Argument ...*] Erweitert nur Parameter.

. *Datei* [*Argument ...*]

Liest die ganze angegebene Datei und führt dann die Befehle aus. Die Befehle werden in der aktuellen Shell-Umgebung ausgeführt. Der durch die Variable **PATH** angegebene Suchpfad wird verwendet, um das Verzeichnis mit der angegebenen Datei zu suchen. Sind Argumente angegeben, werden sie zu Positionsparametern. Ansonsten bleiben die Positionsparameter unverändert. Der Ausführungsstatus ist der des letzten ausgeführten Befehls. Der Abschnitt "Parametersubstitution in der Korn-Shell" auf Seite A-28 enthält weitere Informationen über Positionsparameter.

Hinweis: Der Befehl . *Datei* [*Argument ...*] liest die ganze Datei, bevor Befehle ausgeführt werden. Deshalb sind die Befehle **alias** und **unalias** in der Datei nicht auf Funktionen anwendbar, die in der Datei definiert sind.

alias [-*tx*] [*Name* [=*Wert*]] ...

Gibt die Liste der Alias-Namen im Format *Name*=*Wert* in die Standardausgabe aus. Ein Alias-Name wird für jeden Namen definiert, dem ein Wert zugeordnet wurde. Enthält der Parameter *Wert* ein abschließendes Leerzeichen, wird das nächste Wort auf Alias-Namenssubstitution überprüft.

Die Option **-t** setzt Alias-Namen mit absoluter Pfadangabe und listet sie auf. Der Wert eines Alias-Namens mit absoluter Pfadangabe ist der vollständige Pfadname eines Programms, das durch den Parameter *Name* bestimmt wird. Die Definition dieses Wertes wird aufgehoben, wenn der Wert der Variable **PATH** zurückgesetzt wird, die Alias-Namen bleiben jedoch mit absoluten Pfadnamen versehen. Beispielsweise erzeugen die Befehle:

```
alias -t ls
alias -t
```

die Systemantwort `ls=/usr/bin/ls`.

Die Option **-x** setzt exportierte Alias-Namen oder gibt sie aus. Ein exportierter Alias-Name bleibt definiert für Prozeduren, die mit dem Parameter *Name* aufgerufen werden und die keinen separaten Aufruf des Befehls **ksh** veranlassen. Beispielsweise erzeugen die Befehle:

```
alias -x ee=echo
alias_ee
```

die Systemantwort `ee alias set`. Der Ausführungsstatus ist ungleich Null, wenn ein Name angegeben wurde. Er hat jedoch keinen Wert, wenn kein Alias-Name definiert wurde.

Hinweis: Wird ein Befehl ausgeführt, bei dem es sich um einen Alias-Namen mit absoluter Pfadangabe handelt, und danach ein Befehl mit demselben Namen in einem Suchpfadverzeichnis installiert, das im Suchpfad vor dem Verzeichnis des ursprünglichen Befehls steht, führt die Shell weiterhin den ursprünglichen Befehl aus. Um diesen Fall zu korrigieren, die Option `-t` des Befehls **alias** verwenden.

bg [*Job...*] Stellt jeden angegebenen Job in den Hintergrund. Ist kein Parameter *Job* angegeben, wird der aktuelle Job in der Hintergrund gestellt. Der Abschnitt "Jobsteuerung in der Korn-Shell" auf Seite A-37 enthält weitere Informationen zur Jobsteuerung.

break [*n*] Verläßt die Programmschleife **for**, **while**, **until** oder **select**, falls vorhanden. Wird der Parameter *n* angegeben, unterbricht der Befehl die Anzahl der Ebenen, die durch *n* angegeben werden. Der Wert von *n* ist eine beliebige ganze Zahl größer oder gleich 1.

cd [*Argument*]
cd *Alt Neu*

Dieser Befehl kann beide Formate haben. Mit dem ersten Format wird vom aktuellen Verzeichnis in das gewechselt, das im Parameter *Argument* angegeben wird. Ist der Wert des Parameters *Argument* ein `-` (Minuszeichen), wird in das vorherige Verzeichnis gewechselt. Die Shell-Variable **HOME** ist der Standardwert des Parameters *Argument*. Die Variable **PWD** ist für das aktuelle Verzeichnis gesetzt.

Die Shell-Variable **CDPATH** definiert den Suchpfad für das Verzeichnis, das den Wert des Parameters *Argument* enthält. Alternative Verzeichnisnamen werden durch einen `:` (Doppelpunkt) abgetrennt. Der Standardpfad ist leer, wodurch das aktuelle Verzeichnis angegeben ist. Das aktuelle Verzeichnis wird durch einen leeren Pfadnamen angegeben, der unmittelbar hinter dem Gleichheitszeichen steht oder zwischen begrenzenden Doppelpunkten an irgendeiner Stelle in der Pfadliste. Beginnt das angegebene Argument mit einem `/` (Schrägstrich), wird der Suchpfad nicht verwendet. Ansonsten wird in jedem Verzeichnis im Pfad nach dem Argument gesucht.

Mit dem zweiten Format des Befehls **cd** wird die durch die Variable *Alt* angegebene Zeichenfolge durch die Zeichenfolge der Variablen *Neu* im aktuellen Verzeichnisnamen, **PWD**, ersetzt und es wird versucht, in dieses neue Verzeichnis zu wechseln.

continue [*n*] Fährt mit der nächsten Iteration der Programmschleife **for**, **while**, **until** oder **select** fort. Wird die Variable *n* angegeben, fährt der Befehl mit der *n*ten Programmschleife fort. Der Wert von *n* ist eine beliebige ganze Zahl größer oder gleich 1.

echo [*Argument ...*]

Schreibt Zeichenfolgen in die Standardausgabe. Weitere Informationen enthält der Artikel zum Befehl **echo**.

eval [*Argument ...*]

Lest die angegebenen Argumente als Eingabe und führt die daraus resultierenden Befehle aus.

exec [*Argument ...*]

Führt anstelle der Shell den Befehl, der durch das Argument angegeben wird, aus (ohne einen neuen Prozeß zu erstellen). Ein- und Ausgabeargumente können erscheinen und den aktuellen Prozeß beeinflussen. Wird kein Argument angegeben, ändert der Befehl **exec** die Dateideskriptoren, und zwar so, wie in der Ein-/Ausgabeumadressierungsliste vorgeschrieben. In diesem Fall werden alle Dateideskriptoren mit Nummern größer als 2, die auf diese Weise geöffnet wurden, beim Aufrufen eines anderen Programms wieder geschlossen.

exit [*n*]

Verläßt die Shell mit dem im Parameter *n* angegebenen Ausführungsstatus. Wird der Parameter *n* übergangen, ist der Ausführungsstatus der des letzten ausgeführten Befehls. Mit einem Dateiendezeichen kann die Shell ebenfalls verlassen werden, es sei denn, die Option **ignoreeof** des Sonderbefehls **set** ist aktiv.

export [*Name[=Wert]*] ...

Markiert die angegebenen Namen für den automatischen Export in die Umgebung von nachfolgend ausgeführten Befehlen.

fc [**-e** *Editorname*] [**-nlr**] [*Erster* [*Letzter*]]

fc **-e** **-** [*Alt=Neu*] [*Befehl*]

Mit dem ersten Format wird durch die Parameter *Erster* bis *Letzter* eine Gruppe von Befehlen aus den letzten Befehlen mit der Variable **HISTSIZE** ausgewählt, die an der Datenstation eingegeben wurden. Die Parameter *Erster* und *Letzter* werden als Zahlen oder Zeichenfolgen angegeben. Mit einer Zeichenfolge wird der aktuellste Befehl gesucht, angefangen mit der ersten angegebenen Zeichenfolge. Eine negative Zahl wird als Relativzeiger zur aktuellen Befehlsnummer verwendet. Wird kein Parameter *Letzter* angegeben, wird der Befehl auf den Wert des Parameters *Erster* gesetzt. Wird kein Parameter *Erster* angegeben, ist der Standardwert für das Editieren der vorherige Befehl und für das Auflisten -16. Die Optionen haben folgende Funktionen:

- l** Listet die Befehle in der Standardausgabe auf.
- n** Unterdrückt Befehlsnummern bei der Auflistung.
- r** Dreht die Reihenfolge der Befehle in der Liste um.

Wird die Option **-l** nicht angegeben, ruft der Befehl den Editor, der durch den Parameter *Editorname* angegeben ist, auf für eine Datei, die diese Tastaturbefehle enthält. Wird kein Parameter *Editorname* angegeben, wird der Wert der Variable **FCEDIT** (standardmäßig die Datei **/usr/bin/ed**) als Wert für den Editor verwendet. Ist der Editiervorgang beendet, führt der Befehl den oder die editierten Befehle aus.

Mit dem zweiten Format wird der angegebene Befehl nach Ausführung der Substitution *Alt=Neu* erneut ausgeführt.

Hinweis: Die Verwendung des integrierten Befehls **fc** innerhalb eines zusammengesetzten Befehls führt dazu, daß der Befehl aus der Protokolldatei entfernt wird.

fg [*Job...*]

Bringt jeden angegebenen Job in den Vordergrund. Wird kein Job angegeben, bringt der Befehl den aktuellen Job in den Vordergrund.

getopts *Optionszeichenfolge Name [Argument ...]*

Überprüft den Parameter *Argument* auf gültige Optionen. Wird der Parameter *Argument* übergangen, werden die Positionsparameter verwendet. (Der Abschnitt "Parametersubstitution in der Korn-Shell" auf Seite A-28 enthält weitere Informationen über Positionsparameter.) Ein Optionsargument beginnt mit einem + (Pluszeichen) oder einem - (Minuszeichen). Beginnt eine Option weder mit einem +, einem - noch mit dem Argument -, wird sie beendet.

Der Parameter *Optionszeichenfolge* enthält die Buchstaben, die vom Befehl **getopts** erkannt werden. Folgt auf einen Buchstaben ein Doppelpunkt, hat diese Option ein Zeichenfolgeargument. Solche Optionen können durch Leerzeichen vom Zeichenfolgeargument getrennt werden. Der Befehl **getopts** stellt den nächsten gefundenen Buchstaben der Option in den Parameter *Name* und zwar jedesmal, wenn der Befehl durch ein + (Pluszeichen) vor dem Parameter *Argument* aufgerufen wird.

Der Index des nächsten Parameters *Argument* wird in der Variablen **OPTIND** gespeichert. Das Optionsargument wird, falls vorhanden, in der Variablen **OPTARG** gespeichert. Ein führender Doppelpunkt im Parameter *Optionszeichenfolge* führt dazu, daß der Befehl **getopts** den Buchstaben einer ungültigen Option in der Variablen **OPTARG** speichert. Er führt auch dazu, daß der Befehl den Parameter *Name* auf ? (Fragezeichen) setzt für eine unbekannte Option und auf : (Doppelpunkt), wenn eine erforderliche Option fehlt. Ansonsten gibt der Befehl **getopts** eine Fehlermeldung aus. Der Ausführungsstatus ist ungleich Null, wenn keine weiteren Optionen angegeben sind.

jobs [-lnp] [*Job ...*]

Listet Informationen zu den angegebenen Jobs auf. Wird die Variable *Job* übergangen, listet der Befehl Informationen zu allen aktiven Jobs auf. Die Optionen haben folgende Funktionen:

- l Listet zusätzlich zu den üblichen Informationen Prozeß-IDs auf.
- n Zeigt nur Jobs an, die seit der letzten Auflistung gestoppt oder verlassen wurden.
- p Listet nur Prozeßgruppen auf.

kill [-Signal] *Job ...*

kill -l

Sendet entweder das Signal **TERM** (beenden) oder die angegebenen Signale zu den angegebenen Jobs oder Prozessen. Signale werden entweder durch Zahlen oder Namen angegeben. Über die Datei **/usr/include/signal.h** werden Signalnamen definiert (ohne das Präfix **SIG**). Wird das Signal **TERM** oder **HUP** (nichtprogrammierter Stopp) gesendet, wird das Signal **CONT** (weiter) zum Job oder Prozeß gesendet, wenn er gestoppt wurde. Der Parameter *Job* gibt die Prozeß-ID eines Prozesses an, der nicht Member eines aktiven Jobs ist. Mit dem zweiten Format dieses Befehls werden die Signalnummern und -namen aufgelistet. Der Befehl **kill** -l zeigt auch Fehlermeldungen für Signale an, die keinen Namen haben.

let *Ausdruck ...*

Wertet die angegebenen arithmetischen Ausdrücke aus. Der Ausführungsstatus ist Null, wenn der Wert des letzten Ausdrucks ungleich Null ist, und umgekehrt. Der Abschnitt "Arithmetische Berechnungen in der Korn-Shell" auf Seite A-23 enthält weitere Informationen.

newgrp [*Gruppe*] [*Argument ...*]

Äquivalent zum Befehl **exec/usr/bin/newgrp** [*Gruppe ...*].

Hinweis: Dieser Befehl gibt keine Ausgabe zurück.

print [-Rnprsu[*n*]] [*Argument ...*]

Gibt die Shell-Ausgabe aus. Werden keine Optionen angegeben oder werden die Optionen - (Minuszeichen) oder — (Gedankenstrich) angegeben, werden die Argumente in die Standardausgabedatei ausgegeben, wie beim Befehl **echo** beschrieben. Die Optionen haben folgende Funktionen:

- R Gibt eine unformatierte Ausgabe aus (die Escape-Vereinbarungen des Befehls **echo** werden ignoriert). Die Option -R gibt alle folgenden Argumente und Optionen aus mit Ausnahme von -n.
- n Verhindert, daß das Zeilenvorschubzeichen der Ausgabe hinzugefügt wird.
- p Schreibt die Argumente nicht in die Standardausgabe, sondern in die Pipe des Prozesses, der mit I& (Balken, Et-Zeichen) ausgeführt wird.
- r Gibt eine unformatierte Ausgabe aus. Die Escape-Vereinbarungen des Befehls **echo** werden ignoriert.
- s Schreibt die Argumente in die Protokolldatei statt in die Standardausgabe.
- u Gibt eine einstellige Einheitennummer, *n*, für den Dateideskriptor an, wohin die Ausgabe gestellt wird. Der Standardwert ist 1.

pwd Äquivalent zu **print -r - \$PWD**.

Hinweis: Der interne Korn-Shell-Befehl **pwd** unterstützt keine symbolische Verbindungen.

read [-prsu[*n*]] [*Name?Eingabeaufforderung*] [*Name ...*]

Liest die Shell-Eingabe. Jeweils eine Zeile wird gelesen und in Felder aufgeteilt, wobei die Zeichen in der Variablen **IFS** als Trennzeichen verwendet werden:

- p Liest die Eingabezeile aus der Eingabe-Pipe eines Prozesses, der von der Shell unter Verwendung von I& (Balken, Et-Zeichen) ausgeführt wird.
- r Liest eine unformatierte Eingabe. Die Escape-Vereinbarungen werden ignoriert. Der \ (umgekehrte Schrägstrich) am Ende einer Zeile hat nicht die Bedeutung der Zeilenfortsetzung.
- s Speichert die Eingabe als Befehl in der Protokolldatei.
- u Gibt eine einstellige Einheitennummer, *n*, für den Dateideskriptor an, von wo die Eingabe gelesen werden soll. Der Dateideskriptor kann mit dem Sonderbefehl **exec** geöffnet werden. Der Standardwert der Variable *n* ist 0.

Wird der Parameter *Name* übergangen, wird **REPLY** als Standardname verwendet. Der Ausführungsstatus ist Null, es sei denn, der Befehl trifft auf ein Dateiendezeichen. Das Dateiendezeichen zusammen mit der Option -p löscht diesen Prozeß, so daß ein neuer Prozeß aufgesetzt werden kann. Falls das erste Argument ein ? (Fragezeichen) enthält, wird der Rest des Wortes als eine Eingabeaufforderung auf Standardfehler verwendet, wenn die Shell interaktiv ist. Der Ausführungsstatus ist Null, es sei denn, der Befehl trifft auf ein Dateiendezeichen.

readonly [*Name[=Wert]*] ...

Markiert die im Parameter *Name* angegebenen Namen als

schreibgeschützt. Diese Namen können durch eine nachfolgende Zuordnung nicht geändert werden.

return [*n*] Bewirkt, daß eine Shell-Funktion zur aufrufenden Prozedur zurückkehrt. Der Rückkehrstatus wird in der Variablen *n* angegeben. Wird die Variable *n* übergangen, ist der Rückkehrstatus der des letzten ausgeführten Befehls. Wird der Befehl **return** außerhalb einer Funktion oder Prozedur aufgerufen, hat er dieselbe Wirkung wie der Befehl **exit**.

set [+|-aefhkmnostuvx] [+|-o *Option*]... [+|-A *Name*] [*Argument* ...]

Die Optionen für diesen Befehl werden folgendermaßen interpretiert:

- A** Feldgruppenzuordnung. Hebt die Einstellung des Parameters *Name* auf und ordnet ihm nacheinander die Werte aus der angegebenen Parameterliste *Argument* zu. Bei Verwendung der Option **+A** wird der Parameter *Name* nicht als erster inaktiviert.
- a** Exportiert automatisch alle nachfolgenden definierten Parameter.
- e** Führt, falls gesetzt, die Hardwareprogrammverzweigung **ERR** aus, und beendet die Ausführung, wenn der Ausführungsstatus eines Befehls ungleich Null ist. Dieser Modus ist während des Lesens von Profilen inaktiviert.
- f** Inaktiviert die Dateinamenssubstitution.
- h** Kennzeichnet jeden Befehl beim ersten Auftreten als einen Alias-Namen mit absoluter Pfadangabe.
- k** Stellt alle Parameterzuordnungsargumente in die Umgebung für einen Befehl, nicht nur die Argumente, die vor dem Befehlsnamen stehen.
- m** Führt Hintergrundjobs in einem separaten Prozeß aus und gibt beim Abschluß eine Zeile aus. Der Ausführungsstatus von Hintergrundjobs wird durch eine Abschlußnachricht mitgeteilt. Bei Systemen mit Jobsteuerung wird diese Option automatisch für interaktive Shells aktiviert (siehe Abschnitt "Jobsteuerung in der Korn-Shell" auf Seite A-37.)
- n** Liest Befehle und überprüft sie auf Syntaxfehler, führt sie jedoch nicht aus. Diese Option wird für interaktive Shells ignoriert.
- o** Gibt die aktuelle Einstellung der Optionen aus, wenn keine Argumente angegeben sind. Argumente können, wie in der Variablen **geticon** angegeben, die folgenden sein:
 - allexport** Wie Option **-a**.
 - errexit** Wie Option **-e**.
 - bgnice** Führt alle Hintergrundjobs mit einer niedrigeren Priorität aus. Dies ist der Standardmodus.
 - emacs** Tritt zur Befehlseingabe in eine Variante des Zeileneditors emacs ein.
 - gmacs** Tritt zur Befehlseingabe in eine Variante des Zeileneditors gmacs ein.
 - ignoreeof** Verläßt die Shell nicht beim Auftreten eines Dateieindezeichens. Entweder muß der Befehl **exit** verwendet werden, oder die Option überschrieben und die Shell durch Drücken der Tastenfolge Strg-D (mehr als 11 Mal) verlassen werden.
 - keyword** Wie Option **-k**.
 - markdirs** Hängt an alle Verzeichnisnamen, die das Ergebnis der Dateinamenssubstitution sind, einen / (Schrägstrich) an.
 - monitor** Wie Option **-m**.

- noclobber** Verhindert, daß bei der Umadressierung vorhandene Dateien abgeschnitten werden. Wird diese Option angegeben, muß auf das Umadressierungssymbol ein Balken folgen (>|), wenn eine Datei abgeschnitten werden soll.
- noexec** Wie Option **-n**.
- noglob** Wie Option **-f**.
- nolog** Verhindert, daß Funktionsdefinitionen in der Protokolldatei gespeichert werden.
- nounset** Wie Option **-u**.
- privileged** Wie Option **-p**.
- verbose** Wie Option **-v**.
- trackall** Wie Option **-h**.
- vi** Tritt zur Befehlseingabe in den Einfügemodus einer Variante des Zeileneditors vi ein. Die Eingabe des Escape-Zeichens 033 bringt den Editor in den Übertragungsmodus. Durch Drücken der Eingabetaste wird der Befehl an das System übergeben.
- viraw** Verarbeitet jedes Zeichen so, wie es im vi-Modus eingegeben wurde.
- xtrace** Wie Option **-x**.
- p** Inaktiviert die Verarbeitung der Datei **\$HOME/.profile** und verwendet die Datei **/etc/suid_profile** anstelle der Datei **ENV**. Dieser Modus wird immer aktiviert, wenn die aktuelle Benutzer-ID (UID) oder Gruppen-ID (GID) nicht mit der realen UID oder GID übereinstimmt. Wird diese Option ausgeschaltet, wird die aktuelle UID oder GID durch die reale UID bzw. GID ersetzt.
Hinweis: Die Option **-p** wird vom System nicht unterstützt, da das Betriebssystem die Shell-Prozedur **setuid** nicht unterstützt.
- s** Sortiert die Positionsparameter lexikographisch.
- t** Bricht die Ausführung ab, nachdem ein Befehl gelesen und ausgeführt wurde.
- u** Behandelt nicht gesetzte Parameter bei der Substitution wie Fehler.
- v** Gibt Shell-Eingabezeilen so aus, wie sie gelesen werden.
- x** Gibt Befehle und ihre Argumente so aus, wie sie ausgeführt werden.
- Inaktiviert die Optionen **-x** und **-v** und stoppt die Überprüfung von Argumenten nach Optionen.
- Verhindert, daß Optionen geändert werden. Diese Option ist nützlich, wenn der Parameter **\$1** auf einen Wert gesetzt werden soll, der mit einem **-** (Minuszeichen) beginnt. Folgen auf diese Option keine Argumente, werden die Positionsparameter nicht gesetzt.

Geht einer der Optionen des Befehls **set** ein **+** (Pluszeichen) voraus statt eines **-** (Minuszeichen), wird die Option ausgeschaltet. Diese Optionen können beim Aufrufen der Shell verwendet werden. Die aktuelle Gruppe von Optionen befindet sich im Parameter **\$-**. Wird die Option **-A** nicht angegeben, sind die verbleibenden Argumente Positionsparameter und sind in Reihenfolge **\$1**, **\$2** usw. zugeordnet. Werden keine Argumente angegeben, werden die Namen und Werte aller benannten Parameter in der Standardausgabe ausgegeben.

shift [*n*] Benennt die Positionsparameter um, angefangen mit $\$_{n+1}$... bis $\$1$ Der Standardwert des Parameters *n* ist 1. Der Parameter *n* ist jeder beliebige arithmetische Ausdruck, der eine positive Zahl berechnet, die kleiner-gleich dem Parameter $\$#$ ist.

times Gibt die aufgelaufenen Benutzer- und Systemzeiten für die Shell und für Prozesse, die von der Shell aus ausgeführt werden, aus.

trap [*Befehl*] [*Signal*] ...

Führt den angegebenen Befehl aus, wenn die Shell das/die angegebene/n Signal/e empfängt. Der Parameter *Befehl* wird einmal gelesen, wenn die Hardwareprogrammverzweigung gesetzt wird und ein weiteres Mal, wenn die Hardwareprogrammverzweigung durchgeführt wird. Der Parameter *Signal* kann als Zahl oder als Name des Signals angegeben werden. Hardwareprogrammverzweigungsbefehle werden in der Reihenfolge der Signalnummern ausgeführt. Der Versuch, eine Hardwareprogrammverzweigung auf ein Signal zu setzen, das beim Eingang zur aktuellen Shell ignoriert wurde, ist wirkungslos.

Ist der Befehl ein - (Minuszeichen), werden alle Hardwareprogrammverzweigungen auf ihren ursprünglichen Wert zurückgesetzt. Wird der Befehl übergangen und ist das erste Signal ein numerisches Signal, setzt der Befehl **ksh** die Werte des oder der Parameter *Signal* auf den Anfangswert zurück.

Hinweis: Wird der Befehl übergangen und ist das erste Signal ein symbolischer Name, wird das Signal als Befehl interpretiert.

Ist der Wert des Parameters *Signal* das Signal **ERR**, wird der angegebene Befehl immer ausgeführt, wenn ein Befehl den Ausführungsstatus Null hat. Ist das *Signal* **DEBUG**, wird der angegebene Befehl nach jedem Befehl ausgeführt. Ist der Wert des Parameters *Signal* das Signal **0** oder **EXIT** und wird der Befehl **trap** innerhalb des Hauptteils einer Funktion ausgeführt, wird der angegebene Befehl nach Beendigung der Funktion ausgeführt. Ist der Parameter *Signal* **0** oder **EXIT** für den Befehl **trap**, der außerhalb einer Funktion gesetzt wurde, wird der angegebene Befehl beim Verlassen der Shell ausgeführt. Der Befehl **trap** ohne Argumente gibt eine Liste mit Befehlen aus, die jeder Signalnummer zugeordnet sind.

Für eine vollständige Liste der Parameterwerte *Signal* siehe die Abschnitte zu den Unterroutinen **sigaction**, **sigvec** oder **signal**.

typeset [+HLRZfirtux[*n*]] [*Name*[=*Wert*]]...

Setzt Attribute und Werte für Shell-Parameter. Wird dieser Befehl innerhalb einer Funktion aufgerufen, wird ein neues Exemplar des Parameters *Name* erstellt. Parameterwert und -art werden zurückgeschrieben, wenn die Funktion beendet ist. Die folgenden Optionen können mit dem Befehl **typeset** angegeben werden:

- H Dateiabbildung AIX-zu-Host .
- L Richtet den Parameter *Wert* linksbündig aus und entfernt führende Leerzeichen. Hat der Parameter *n* einen Wert ungleich Null, definiert er die Satzweite des Feldes; ansonsten wird sie durch die Satzweite des Wertes der ersten Zuordnung festgelegt. Ist der Parameter zugeordnet, wird er rechts mit Leerzeichen gefüllt oder, falls notwendig, abgeschnitten, um in das Feld zu passen. Führende Nullen werden entfernt, wenn die Option **-Z** ebenfalls gesetzt wurde. Die Option **-R** wird ausgeschaltet.

- R Richtet den Parameter rechtsbündig aus und füllt das Feld mit führenden Leerzeichen auf. Hat der Parameter *n* einen Wert ungleich Null, definiert er die Satzweite des Feldes; ansonsten wird sie durch die Satzweite des Wertes der ersten Zuordnung festgelegt. Das Feld bleibt mit Leerzeichen aufgefüllt oder wird vom Ende her abgeschnitten, wenn der Parameter neu zugeordnet wird. Die Option **-L** wird ausgeschaltet.
- Z Richtet den Parameter rechtsbündig aus und füllt das Feld mit führenden Nullen auf, wenn das erste belegte Zeichen eine Ziffer ist und die Option **-L** nicht gesetzt ist. Hat der Parameter *n* einen Wert ungleich Null, definiert er die Satzweite des Feldes; ansonsten wird sie durch die Satzweite des Wertes der ersten Zuordnung festgelegt.
- f Zeigt an, daß die Namen sich auf Funktionen statt auf Parameter beziehen. Zuordnungen können nicht vorgenommen werden und die einzigen weiteren gültigen Optionen sind **-t**, **-u** und **-x**. Die Option **-t** schaltet die Ausführungsablaufverfolgung für diese Funktion ein. Die Option **-u** führt dazu, daß diese Funktion als undefiniert markiert wird. Die Variable **FPATH** wird durchsucht, um die Funktionsdefinition zu finden, wenn auf die Funktion verwiesen wird. Die Option **-x** ermöglicht es, daß die Funktionsdefinition über Shell-Prozeduren hinweg gültig bleibt, wenn es sich dabei nicht um separate Aufrufe des Befehls **ksh** handelt.
- i Identifiziert den Parameter als ganze Zahl, wodurch Berechnungen beschleunigt werden. Hat der Parameter *n* einen Wert ungleich Null, definiert er die arithmetische Basis der Ausgabe; ansonsten wird diese Basis durch die erste Zuordnung festgelegt.
- l Wandelt alle Großbuchstaben in Kleinbuchstaben um. Die Option **-u** (Umwandlung in Großbuchstaben) ist ausgeschaltet.
- r Markiert die im Parameter *Name* angegebenen Namen als schreibgeschützt. Diese Namen können durch eine nachfolgende Zuordnung nicht geändert werden.
- t Kennzeichnet die benannten Parameter. Kennzeichen können vom Benutzer definiert werden und haben in der Shell keine Sonderbedeutung.
- u Wandelt alle Kleinbuchstaben in Großbuchstaben um. Die Option **-l** (Umwandlung in Kleinbuchstaben) ist ausgeschaltet.
- x Markiert den im Parameter *Name* angegebenen Namen für den automatischen Export in die Umgebung von nachfolgend ausgeführten Befehlen.

Bei Verwendung eines + (Pluszeichen) statt eines - (Minuszeichen) werden die Optionen des Befehls **typeset** ausgeschaltet. Werden keine Parameter *Name* angegeben, aber Optionen, wird eine Liste mit Namen (und wahlweise die Werte) der Parameter ausgegeben, die über diese Optionen verfügen. (Bei Verwendung eines + statt eines - werden die Werte nicht ausgegeben.) Werden keine Namen oder Optionen angegeben, werden die Namen und Attribute von allen Parametern ausgegeben.

ulimit [-HSacdfmst] [*Limit*]

Setzt die Grenzen für Benutzerprozeßressourcen oder zeigt sie an, wie sie in der Datei **/etc/security/limits** definiert sind. Diese Datei enthält sechs Standardgrenzwerte:

```
fsize = 2097151
core = 2048
cpu = 3600
data = 131072
rss = 65536
stack = 8192
```

Diese Werte werden als Standardeinstellung verwendet, wenn ein neuer Benutzer dem System hinzugefügt wird. Diese Werte werden bei Einrichtung eines neuen Benutzers mit dem Befehl **mkuser** eingestellt oder mit dem Befehl **chuser** geändert.

Grenzwerte können entweder veränderlich oder fest sein. Veränderliche Grenzwerte können vom Benutzer geändert werden bis zu dem in den festen Grenzwerten festgelegten Maximum und zwar mit dem Befehl **ulimit**. Um die festen Grenzwerte zu ändern, muß Root-Benutzerberechtigung vorliegen.

Viele Systeme enthalten keinen oder nicht mehrere dieser Grenzwerte. Der Grenzwert für eine angegebene Ressource wird eingestellt, wenn der Parameter *Limit* angegeben wird. Der Wert des Parameters *Limit* kann eine Zahl in der Einheit, die für jede Ressource angegeben wird, sein oder der Wert *unlimited* (uneingeschränkt). Die folgenden Optionen können mit dem Befehl **ulimit** angegeben werden:

- H** Gibt an, daß der feste Grenzwert für eine bestimmte Ressource gesetzt wurde. Liegt Root-Benutzerberechtigung vor, kann der feste Grenzwert erhöht werden. Er kann von jedem Benutzer vermindert werden.
- S** Gibt an, daß der veränderliche Grenzwert für eine bestimmte Ressource gesetzt wurde. Ein veränderlicher Grenzwert kann bis zur Höhe des festen Grenzwerts erhöht werden. Ist weder die Option **-H** noch die Option **-S** angegeben, gilt der Grenzwert für beide.
- a** Listet alle aktuellen Ressourcengrenzen auf.
- c** Gibt die Anzahl der 512-Byte-Blöcke für die Größe der Kernspeicherauszüge an.
- d** Gibt die Größe des Datenbereichs in Kilobyte an.
- f** Gibt die Anzahl der 512-Byte-Blöcke für Dateien an, die von Sohnprozessen geschrieben wurden (Dateien jeglicher Größe können gelesen werden).
- m** Gibt die Anzahl der Kilobyte für die Größe des physischen Speichers an.
- s** Gibt die Anzahl der Kilobyte für die Größe des Stapelspeicherbereichs an.
- t** Gibt die Anzahl der Sekunden an, die von jedem Prozeß verwendet werden dürfen.

Die aktuelle Ressourcengrenze wird ausgegeben, wenn die Variable *Limit* übergangen wird. Der veränderliche Grenzwert wird ausgegeben, wenn die Option **-H** nicht angegeben ist. Wird mehr als eine Ressource angegeben, werden der Grenzwertname und die Einheit vor dem Wert ausgegeben. Ist keine Option angegeben, wird die Option **-f** angenommen.

Weitere Informationen über Benutzer- und Systemressourcengrenzen siehe die Abschnitte zu den Unterroutinen **getrlimit**, **setr limit** oder **vlimit**.

umask [*Maske*]

Setzt die Dateimoduserstellungsmaske des Benutzers auf den im Parameter *Maske* angegebenen Wert. Der Wert des Parameters *Maske* ist entweder eine Oktalzahl oder ein symbolischer Wert, wie bei dem Befehl **chmod** beschrieben. Ist ein symbolischer Wert gegeben, ist der neue Wert **umask** die Ergänzung des Parameterwertes *Maske*, angewandt auf die Ergänzung des vorherigen Wertes **umask**. Wird der Parameter *Maske* übergangen, wird der aktuelle Wert der Maske ausgegeben.

unalias *Name* ...

Entfernt die durch die Namensliste angegebenen Parameter aus der Alias-Namenliste.

unset [-f] *Name* ...

Löscht die Werte und Attribute der durch die Namensliste angegebenen Parameter. Schreibgeschützte Variablen können nicht inaktiviert werden. Ist die Option **-f** gesetzt, beziehen sich die Namen auf Funktionsnamen. Das Inaktivieren der Variablen **ERRNO**, **LINENO**, **MAILCHECK**, **OPTARG**, **OPTIND**, **RANDOM**, **SECONDS**, **TMOU** und **_** (Unterstrichszeichen) löscht deren Sonderbedeutung, auch wenn sie nachfolgend zugeordnet wird.

wait [*Job*]

Wartet auf den angegebenen Job und informiert über dessen Beendigungsstatus. Wird kein Job angegeben, wartet der Befehl auf alle momentan aktiven Sohnprozesse. Der Ausführungsstatus von diesem Befehl bezieht sich auf den Prozeß, auf den er wartet.

whence [-pv] *Name* ...

Zeigt für jeden angegebenen Namen an, wie er bei Verwendung als Befehlsname interpretiert würde.

-p Führt eine Pfadsuche für den oder die angegebenen Namen durch, sogar wenn es sich dabei um Alias-Namen, Funktionen oder reservierte Wörter handelt.

-v Erzeugt einen umfassenderen Bericht.

Das Befehlsprotokoll in der Korn-Shell

Die Korn-Shell speichert Befehle, die an einer Datenstationseinheit eingegeben wurden, in einer Protokolldatei. Falls gesetzt, ist der Wert der Variable **HISTFILE** der Name der Protokolldatei. Ist die Variable **HISTFILE** nicht gesetzt oder kann sie nicht geschrieben werden, wird die Datei **\$HOME/.sh_history** als Protokolldatei verwendet. Die Shell greift auf die Befehle aller interaktiven Shells unter Verwendung derselben benannten Variablen **HISTFILE** zu.

Standardmäßig speichert die Korn-Shell den Text der letzten 128 Befehle, die an einer Datenstationseinheit eingegeben wurden. Die Größe der Protokolldatei (durch die Variable **HISTSIZE** angegeben) ist nicht begrenzt, eine sehr große Protokolldatei kann jedoch dazu führen, daß die Korn-Shell nur langsam startet.

Befehlsprotokollsubstitution

Den Sonderbefehl **fc** verwenden, um Teile der Protokolldatei aufzulisten oder zu editieren. Um einen Teil der Datei, die editiert oder aufgelistet werden soll, auszuwählen, die Nummer oder den oder die ersten Zeichen des Befehls angeben. Es kann ein einzelner Befehl oder eine Reihe von Befehlen angegeben werden.

Wird kein Editor als ein Argument des Sonderbefehls **fc** angegeben, wird der Editor, der durch die Variable **FCEDIT** definiert ist, verwendet. Ist die Variable **FCEDIT** nicht definiert, wird die Datei **/usr/bin/ed** verwendet. Die editierten Befehle werden ausgegeben und ausgeführt, wenn der Editor verlassen wird.

Der Editornamen – (Gedankenstrich) wird verwendet, um die Editierphase zu überspringen und den Befehl erneut auszuführen. In diesem Fall kann ein Substitutionsparameter wie **Alt=Neu** verwendet werden, um den Befehl vor der Ausführung zu ändern. Ist beispielsweise für **r** der Alias-Name **fc -e** – definiert, wird durch die Eingabe von **r schlecht=gut c** der letzte Befehl ausgeführt, der mit dem Buchstaben **c** beginnt, und beim ersten Auftreten die Zeichenfolge **schlecht** durch die Zeichenfolge **gut** ersetzt.

Arithmetische Berechnungen in der Korn-Shell

In der Korn-Shell können mit dem Sonderbefehl **let** Ganzzahlberechnungen ausgeführt werden. Konstanten haben das Format [**Basis**]**Zahl**. Der Parameter **Basis** ist eine Dezimalzahl zwischen 2 und 36 (einschließlich), die die arithmetische Basis darstellt. Der Parameter **Zahl** ist eine Zahl in dieser Basis. Wird der Parameter **Basis** übergangen, verwendet die Shell die Basis 10.

Arithmetische Ausdrücke verwenden dieselbe Syntax, Vorrangstellung und Assoziativität der Ausdrücke wie die Programmiersprache C. Alle Integraloperatoren, mit Ausnahme von **++** (doppeltes Pluszeichen), **—** (Gedankenstrich), **?:** (Fragezeichen, Doppelpunkt) und **,** (Komma) werden unterstützt. In der folgenden Tabelle werden die gültigen Korn-Shell-Operatoren nach Vorrangstellung in absteigender Reihenfolge dargestellt:

Operator	Definition
-	Monadisches Minus
!	Logische Negation
~	Bitweise Negation
*	Multiplikation
/	Division
%	Rest
+	Addition
-	Subtraktion
<<, >>	Verschieben nach links, Verschieben nach rechts
<=, >=, <>, ==, !=	Vergleich
&	Bitweises AND
^	Bitweises exklusives OR

	Bitweises OR
&&	Logisches AND
	Logisches OR
=	Zuordnung
*=, /=, &=	
+=, -=, <<=,	
>>=, &=, ^=, =	

Tabelle 9. Arithmetische Operatoren in der Korn-Shell

Viele arithmetische Operatoren, z.-B. *, &, < und >, haben in der Korn-Shell eine Sonderbedeutung. Diese Zeichen müssen in Anführungszeichen gesetzt werden. Um beispielsweise den aktuellen Wert von `y` mit 5 zu multiplizieren und das Ergebnis `y` neu zuzuordnen, folgenden Ausdruck verwenden:

```
let "y = y * 5"
```

Wird der Ausdruck in Anführungszeichen gesetzt, entfällt die Sonderbedeutung des Zeichens * (Stern).

Innerhalb des Befehls `let` können Operationen gruppiert werden, um eine Zusammenfassung zu erzielen. Im Ausdruck:

```
let "z = q * (z - 10)"
```

beispielsweise multipliziert der Befehl den Wert `q` mit dem verkleinerten Wert von `z`.

Die Korn-Shell beinhaltet ein alternatives Format für den Befehl `let`. Die Korn-Shell behandelt Befehle, die in `(())` (doppelte linke und doppelte rechte runde Klammern) eingeschlossen sind, als Ausdrücke in Anführungszeichen. Demzufolge ist der Ausdruck:

```
((x = x / 3))
```

äquivalent zu:

```
let "x = x / 3"
```

Auf benannte Parameter wird innerhalb eines arithmetischen Ausdrucks durch den Namen verwiesen, ohne die Parametersubstitutionssyntax zu verwenden. Wird auf einen benannten Parameter verwiesen, wird sein Wert als arithmetischer Ausdruck ausgewertet.

Die interne Darstellung der ganzen Zahl eines benannten Parameters wird durch die Option `-i` des Sonderbefehls `typeset` angegeben. Mit der Option `-i` wird die arithmetische Berechnung für den Wert jeder Zuordnung eines benannten Parameters ausgeführt. Wird keine arithmetische Basis angegeben, wird sie durch die erste Zuordnung des Parameters festgelegt. Diese Basis wird bei der Parametersubstitution verwendet.

Befehlssubstitution in der Korn-Shell

Bei der Befehlssubstitution führt die Shell einen bestimmten Befehl aus und ersetzt diesen Befehl durch die Ausgabe. Zur Befehlssubstitution in der Korn-Shell muß der Befehl oder die Zeichenfolge zwischen `$()` (Dollarzeichen, runde Klammern) oder `' '` (Gravis) gestellt werden. Mit dem zweiten Format überprüft die Shell vor der Ausführung den in Anführungszeichen gesetzten Befehl auf besondere Quotierungszeichen hin. Für die meisten Sonderbefehle der Shell, die keine Ein-/Ausgabeumadressierung ausführen, führt die Shell eine Befehlssubstitution aus, ohne einen separaten Prozeß zu erstellen.

Im folgenden Beispiel zeigen die Zeichen `$()` (Dollar-Zeichen, runde Klammern) um den Befehl an, daß die Ausgabe des Befehls `whoami` ersetzt wird:

```
echo Mein Name ist: $(whoami)
```

Dieselbe Befehlssubstitution wird ausgeführt durch:

```
echo Mein Name ist: 'whoami'
```

Die Ausgabe von beiden Beispielen lautet für den Benutzer dee:

```
Mein Name ist: dee
```

Auch arithmetische Ausdrücke können durch das Einschließen in () (runde Klammern) ersetzt werden. Zum Beispiel führt der Befehl:

```
echo Jede Stunde hat $((60 * 60)) Sekunden
```

zu folgendem Ergebnis:

```
Jede Stunde hat 3600 Sekunden
```

Die Korn-Shell löscht bei der Befehlssubstitution alle abschließenden Zeilenvorschubzeichen. Enthält das aktuelle Verzeichnis beispielsweise die Dateien Datei1, Datei2 und Datei3, löscht der Befehl:

```
echo $(ls)
```

die Zeilenvorschubzeichen und erzeugt die folgende Ausgabe:

```
Datei1 Datei2 Datei3
```

Um Zeilenvorschubzeichen beizubehalten, muß der ersetzte Befehl in " " (Anführungszeichen) gesetzt werden:

```
echo "$(ls)"
```

Interpretation von Leerzeichen

Nach der Befehlssubstitution sucht die Korn-Shell im Ergebnis nach den Feldtrennzeichen, die sich in der Variable **IFS** befinden. Wo solche Zeichen gefunden werden, teilt die Shell die Substitution in bestimmte Argumente auf. Die Shell behält explizite Nullargumente (" " oder "") bei und löscht implizite Nullargumente (die aus Parametern ohne Werte resultieren).

Befehls-Alias-Namenprüfung in der Korn-Shell

In der Korn-Shell ist es möglich, Alias-Namen zu erstellen, um Befehle anzupassen. Der Befehl **alias** definiert ein Wort als einen Alias-Namen. Die Korn-Shell speichert definierte Alias-Namen in einer Datei. Wird als erstes Wort in einer Befehlszeile ein Alias-Name verwendet, sucht die Korn-Shell in der Datei für Alias-Namen und ersetzt den Alias-Namen durch den entsprechenden Text.

Das erste Zeichen eines Alias-Namens kann ein beliebiges druckbares Zeichen (kein Sonderzeichen) sein. Die verbleibenden Zeichen müssen dieselben wie für eine gültige Kennung sein. Die ersetzende Zeichenfolge kann eine beliebige Shell-Prozedur enthalten, einschließlich Metazeichen. Die Shell prüft das erste Wort jedes Befehls im ersetzten Text auf Alias-Namen, mit Ausnahme der Wörter, die gerade ersetzt werden.

Ist das letzte Zeichen des Wertes eines Alias-Namens ein Leerzeichen, prüft die Shell auch das folgende Wort zur Alias-Namensubstitution. Alias-Namen können auch verwendet werden, um besondere integrierte Befehle, allerdings keine reservierten Wörter, neu zu definieren. Exportierte Alias-Namen bleiben für Prozeduren, die über ihren Namen aufgerufen werden, gültig, müssen aber für separate Aufrufe der Shell neu initialisiert werden.

Um Alias-Namen zu erstellen, aufzulisten und zu exportieren, den Befehl **alias** verwenden. Mit dem Befehl **unalias** Alias-Namen löschen.

Das Format zum Erstellen von Alias-Namen lautet:

alias *Name=Zeichenfolge*

wobei der Parameter *Name* den Alias-Namen angibt und der Parameter *Zeichenfolge* eine Folge von Zeichen.

Die Shell führt beim Lesen von Prozeduren eine Aliasnamenprüfung durch, aber nicht, wenn sie die Prozeduren ausführt. Daher muß, um einen Alias-Namen aktiv zu setzen, erst zum Definieren der Befehl **alias** ausgeführt werden, bevor die Shell den Befehl liest, in dem auf den Alias-Namen verwiesen wird.

Alias-Namen mit absoluter Pfadangabe

Oft werden Alias-Namen als Kurzform für vollständige Pfadnamen verwendet. Eine Option der Einrichtung zur Aliasnamenprüfung ermöglicht es, daß der Wert eines Alias-Namens automatisch durch den vollständigen Pfadnamen des entsprechenden Befehls ersetzt wird. Diese besondere Art von Alias-Name ist ein Alias-Name *mit absoluter Pfadangabe*.

Alias-Namen mit absoluter Pfadangabe beschleunigen die Ausführung, da die Shell nicht mehr in der Variablen **PATH** nach dem vollständigen Pfadnamen zu suchen braucht.

Der Befehl **set -h** aktiviert die Befehlsprotokollierung, so daß die Shell jedesmal, wenn auf einen Befehl verwiesen wird, den Wert eines Alias-Namens mit absoluter Pfadangabe definiert. Die Definition dieses Wertes wird wieder aufgehoben, wenn die Variable **PATH** zurückgesetzt wird.

Die Alias-Namen selbst bleiben mit der absoluten Pfadangabe erhalten, so daß durch den nächst folgenden Verweis der Wert neu definiert wird. Einige Alias-Namen mit absoluter Pfadangabe sind in der Shell kompiliert.

Die folgenden exportierten Alias-Namen sind in der Shell kompiliert, können aber inaktiviert oder neu definiert werden:

```
autoload='typeset -fu'  
false='let 0'  
functions='typeset -f'  
hash='alias -t'  
history='fc -l'  
integer='typeset -i'  
nohup='nohup '  
r='fc -e -'  
true=':'  
type='whence -v'
```

Tildensubstitution

Nach der Alias-Namenssubstitution überprüft die Shell jedes Wort, ob es mit einer ~ (Tilde), die nicht in Anführungszeichen steht, beginnt. Findet sie ein solches Wort, prüft sie es bis zum ersten / (Schrägstrich), ob es mit einem Benutzernamen in der Datei **/etc/passwd** übereinstimmt. Wird eine Übereinstimmung gefunden, ersetzt die Shell das Zeichen ~ und den Namen durch das Anmeldeverzeichnis des übereinstimmenden Benutzers. Dieser Prozeß wird *Tildensubstitution* genannt.

Die Shell ändert den Ursprungstext nicht, wenn keine Übereinstimmung gefunden wird. Die Korn-Shell führt auch Substitutionen durch, wenn das Zeichen ~ das einzige Zeichen im Wort ist oder von einem / (Schrägstrich), + (Pluszeichen) oder - (Minuszeichen) gefolgt wird:

- ~ oder ~/ Wird ersetzt durch den Wert des Parameters **HOME**.
- ~+ Wird ersetzt durch den Parameter **\$PWD** (den vollständigen Pfadnamen des aktuellen Verzeichnisses).
- Wird ersetzt durch den Parameter **\$OLDPWD** (den vollständigen Pfadnamen des vorherigen Verzeichnisses).

Zusätzlich versucht die Shell, die Tilde zu ersetzen, wenn der Wert eines Variablenzuordnungsparameters mit einer Tilde (~) beginnt.

Parametersubstitution in der Korn-Shell

Ein Parameter kann in mehrere Kategorien eingeteilt werden:

- Eine Kennung von einer oder zwei Stellen, oder einem der folgenden Zeichen: * (Stern), @ (kommerzielles A), # (Pfundzeichen), ? (Fragezeichen), – (Gedankenstrich), \$ (Dollarzeichen) und ! (Ausrufezeichen).
- Ein Argument, das durch eine Nummer (*Positionsparameter*) bezeichnet ist.
- Ein Parameter, der durch eine Kennung bezeichnet ist, mit einem Wert und keinem, einem oder mehreren Attributen (*benannte Parameter*).

Der Sonderbefehl **typeset** ordnet benannten Parametern Werte und Attribute zu. Die von der Korn-Shell unterstützten Attribute werden im Abschnitt zum Sonderbefehl **typeset** beschrieben. Exportierte Parameter übergeben Werte und Attribute an die Umgebung.

Der Wert eines benannten Parameters wird folgendermaßen zugeordnet:

```
Name=Wert `[ Name=Wert ] ...
```

Wird das Ganzzahlattribut **-i** für den Parameter *Name* gesetzt, wird der Parameter *Wert* durch die arithmetische Berechnung ermittelt. Der Abschnitt "Arithmetische Berechnungen in der Korn-Shell" auf Seite A-23 enthält weitere Informationen über die Berechnung arithmetischer Ausdrücke.

Die Shell unterstützt eine eindimensionale Feldgruppenfunktion. Auf ein Element eines Feldgruppenparameters wird durch einen Index verwiesen. Ein Index wird durch einen arithmetischen Ausdruck, der durch [] (eckige Klammern) eingeschlossen ist, gekennzeichnet. Um einer Feldgruppe Werte zuzuordnen, `set -A Name Wert ...` verwenden. Der Wert aller Indizes muß zwischen 0 und 511 liegen. Feldgruppen müssen nicht festgelegt werden. Jeder Verweis auf einen benannten Parameter durch einen gültigen Index ist zulässig und eine Feldgruppe wird, falls erforderlich, erstellt. Der Verweis auf eine Feldgruppe ohne einen Index ist äquivalent zu dem Verweis auf das Element Null.

Positionsparametern werden mit dem Sonderbefehl **set** Werte zugeordnet. Der Parameter **\$0** wird vom Nullargument gesetzt, wenn die Shell aufgerufen wird. Das Zeichen **\$** wird verwendet, um ersetzbare Parameter einzuführen.

Folgende ersetzbare Parameter gibt es:

\${Parameter} Die Shell liest alle Zeichen vom **{** (Dollarzeichen, linke geschweifte Klammer) bis zur übereinstimmenden **}** (rechte geschweifte Klammer) als Teil desselben Wortes, auch wenn das Wort geschweifte Klammern oder Metazeichen enthält. Der Wert des angegebenen Parameters, falls vorhanden, wird ersetzt. Die geschweiften Klammern sind erforderlich, wenn auf den Parameter *Parameter* ein Buchstabe, eine Ziffer oder ein Unterstrichungszeichen folgt, das nicht als Teil des Namens interpretiert werden soll, oder wenn ein benannter Parameter indexiert ist.

Enthält der angegebene Parameter eine oder mehr Ziffern, handelt es sich um einen *Positionsparameter*. Ein Positionsparameter mit mehr als einer Ziffer muß in geschweifte Klammern eingeschlossen sein. Ist der Wert der Variable ein * (Stern) oder ein @ (kommerzielles A), wird jeder Positionsparameter, angefangen mit **\$1**, ersetzt (getrennt durch ein Feldtrennzeichen). Wird eine Feldgruppenkennung mit einem Index * (Stern) oder @ (kommerzielles A) verwendet, wird der Wert für jedes Element (getrennt durch ein Feldtrennzeichen) ersetzt.

`${#Parameter}` Ist der Wert des Parameters *Parameter* ein * oder ein @, wird die Nummer der Positionsparameter ersetzt. Andernfalls wird die im Parameter *Parameter* angegebene Feldlänge ersetzt.

`${#Kennung[*]}` Die Zahl der Elemente in der Feldgruppe, die durch den Parameter *Kennung* angegeben ist, wird ersetzt.

`${Parameter:-Wort}` Ist der Parameter *Parameter* gesetzt und ungleich Null, wird dessen Wert ersetzt, andernfalls der Wert des Parameters *Wort*.

`${Parameter:=Wort}` Ist der Parameter *Parameter* nicht gesetzt oder gleich Null, wird er auf den Wert des Parameters *Wort* gesetzt. Positionsparameter können möglicherweise auf diese Weise nicht zugeordnet werden.

`${Parameter:?Wort}` Ist der Parameter *Parameter* gesetzt und ungleich Null, wird sein Wert ersetzt. Andernfalls wird der Wert der Variable *Wort* ausgegeben und die Shell verlassen. Wird die Variable *Wort* übergangen, wird eine Standardnachricht ausgegeben.

`${Parameter:+Wort}` Ist der Parameter *Parameter* gesetzt und ungleich Null, wird der Wert der Variable *Wort* ersetzt. Andernfalls wird nichts ersetzt.

`${Parameter#Muster}` | **`${Parameter##Muster}`**
Stimmt der angegebene Shell-Parameter *Muster* mit dem Anfang des Wertes für den Parameter *Parameter* überein, ist der Wert dieser Substitution der des Parameters *Parameter*, wobei der übereinstimmende Teil gelöscht wird. Andernfalls wird der Wert des Parameters *Parameter* ersetzt. Mit dem ersten Format wird das kleinste passende Muster gelöscht, mit dem zweiten Format das größte passende Muster.

`${Parameter%Muster}` | **`${Parameter%%Muster}`**
Stimmt der angegebene Shell-Parameter *Muster* mit dem Ende des Wertes für den Parameter *Parameter* überein, ist der Wert dieser Substitution der des Parameters *Parameter*, wobei der übereinstimmende Teil gelöscht wird. Andernfalls wird der Wert der Variable *Parameter* ersetzt. Mit dem ersten Format wird das kleinste passende Muster gelöscht, mit dem zweiten Format das größte passende Muster.

In den vorangegangenen Ausdrücken wird die Variable *Wort* erst ausgewertet, wenn sie als ersetzte Zeichenfolge verwendet wird. Demzufolge wird im folgenden Beispiel der Befehl **`pwd`** nur ausgeführt, wenn die Option **`-d`** nicht gesetzt oder gleich Null ist:

```
echo ${d:-$(pwd)}
```

Hinweis: Wird von den vorangegangenen Ausdrücken her der **`:`** (Doppelpunkt) übergangen, überprüft die Shell lediglich, ob der Parameter *Parameter* gesetzt ist.

Die folgenden Variablen werden von der Shell automatisch gesetzt:

Gibt die Zahl (als Dezimalzahl) der Positionsparameter an, die an die Shell übergeben wurden, ohne den Namen der Shell-Prozedur zu berücksichtigen. Die Variable **`$#`** liefert demzufolge die Zahl des Positionsparameters mit der höchsten Nummer, der gesetzt wurde. Diese Variable dient hauptsächlich zur Überprüfung, ob die erforderliche Zahl von Argumenten vorhanden ist.

- Liefert der Shell Optionen beim Aufruf oder mit dem Befehl **set**.
- ?** Gibt den Rückkehrcode des letzten ausgeführten Befehls an. Sein Wert ist eine dezimale Zeichenfolge. Die meisten Befehle geben den Code 0 zurück, um den erfolgreichen Abschluß anzuzeigen. Die Shell selbst gibt als Rückkehrcode den aktuellen Wert der Variable **\$?** zurück.
- \$** Gibt die Prozeßnummer dieser Shell an.

Das folgenden Beispiel zeigt die empfohlene Vorgehensweise zum Erstellen von temporären Dateien in einem eigens dafür angelegten Verzeichnis:


```
temp=$HOME/temp/$$  
ls >$temp  
  
rm $temp
```
- Zeigt anfänglich den absoluten Pfadnamen der Shell oder Prozedur, die so, wie in der Umgebung übergeben, ausgeführt wird. Nachfolgend ist der Parameter dem letzten Argument des vorherigen Befehls zugeordnet. Dieser Parameter wird nicht für asynchrone Befehle gesetzt. Er wird ebenfalls verwendet, um bei der Überprüfung der elektronischen Post den Namen der übereinstimmenden Datei **MAIL** beizubehalten.
- !** Gibt die Prozeßnummer des letzten aufgerufenen Hintergrundbefehls an.
- ERRNO** Gibt den Wert an, der von der letzten fehlgeschlagenen Unteroutine gesetzt wurde. Dieser Wert ist systemabhängig und dient zur Fehlerbehebung.
- LINENO** Gibt die Zeilennummer der aktuellen Zeile innerhalb der Prozedur oder Funktion an, die ausgeführt wird.
- OLDPWD** Zeigt das vorherige Arbeitsverzeichnis, das vom Befehl **cd** gesetzt wurde.
- OPTARG** Gibt den Wert des letzten Optionsarguments an, das vom Sonderbefehl **getopts** verarbeitet wurde.
- OPTIND** Gibt den Index des letzten Optionsarguments an, das vom Sonderbefehl **getopts** verarbeitet wurde.
- PPID** Gibt die Prozeßnummer des Vaters der Shell an.
- PWD** Zeigt das aktuelle Arbeitsverzeichnis an, das vom Befehl **cd** gesetzt wurde.
- RANDOM** Generiert eine wahlfreie ganze Zahl zwischen 0 und 32767. Die Serie von Zufallszahlen kann durch die Zuordnung eines numerischen Wertes zum Parameter **RANDOM** initialisiert werden.
- REPLY** Wird durch die Anweisung **select** und den Sonderbefehl **read** gesetzt, wenn keine Argumente geliefert wurden.
- SECONDS** Gibt die Zahl der Sekunden an, die seit Aufruf der Shell vergangen sind. Ist diesem Parameter ein Wert zugeordnet, wird der hier zurückgegebene Wert der zugeordnete Wert plus die Zahl der Sekunden seit der Zuordnung sein.

Die folgenden Parameter werden von der Shell verwendet:

- CDPATH** Zeigt den Suchpfad für den Befehl **cd** (Verzeichnis wechseln) an.
- COLUMNS** Definiert die Satzweite des Editierfensters für den Editiermodus der Shell und für die Ausgabe von Listen des Befehls **select**.

EDITOR	Endet der Wert dieses Parameter mit emacs, gmacs oder vi, und wurde die Variable VISUAL mit dem Befehl set nicht gesetzt, wird die entsprechende Option eingeschaltet.
ENV	Wurde dieser Parameter gesetzt, wird für den Wert eine Parametersubstitution ausgeführt, um den Pfadnamen der Prozedur zu generieren, die nach dem Aufruf der Shell ausgeführt wird. Diese Datei wird normalerweise für die Definition von Alias-Namen und Funktionen verwendet.
FCEDIT	Gibt den Standardeditornamen für den Befehl fc an.
FPATH	Gibt den Suchpfad für Funktionsdefinitionen an. Dieser Pfad wird gesucht, wenn auf eine Funktion mit der Option -u verwiesen wird und wenn ein Befehl nicht gefunden wird. Wird eine ausführbare Datei gefunden, wird sie gelesen und in der aktuellen Umgebung ausgeführt.
IFS	Gibt interne Feldtrennzeichen an (normalerweise Leerzeichen, Tabulatorzeichen und Zeilenvorschubzeichen), die verwendet werden, um Befehlswörter zu trennen, die aus der Befehls- oder Parametersubstitution resultieren, und um Wörter mit dem Sonderbefehl read zu trennen. Das erste Zeichen des Parameters IFS wird verwendet, um Argumente für die \$\$ -Substitution abzutrennen.
HISTFILE	Ist dieser Parameter gesetzt, wenn die Shell aufgerufen wird, ist sein Wert der Pfadname der Datei, die zum Speichern des Befehlsprotokolls verwendet wird.
HISTSIZE	Ist dieser Parameter gesetzt, wenn die Shell aufgerufen wird, ist die Zahl der vorher eingegebenen Befehle, auf die diese Shell zugreifen kann, größer-gleich dieser Zahl. Der Standardwert ist 128.
HOME	Zeigt den Namen des Anmeldeverzeichnisses an, das nach Abschluß der Anmeldung zum aktuellen Verzeichnis wird. Das Anmeldeprogramm initialisiert diese Variable. Der Befehl cd verwendet den Wert des Parameters \$HOME als Standardwert. Durch die Verwendung dieser Variablen in einer Shell-Prozedur statt eines expliziten Pfadnamens ist es möglich, Prozeduren von unterschiedlichen Verzeichnissen aus ohne Veränderungen auszuführen.
LANG	Liefert den Standardwert für die Variable LC_* .
LC_ALL	Überschreibt den Wert der Variablen LANG und LC_* .
LC_COLLATE	Bestimmt die Funktionsweise von Bereichsausdrücken innerhalb der Mustererkennung.
LC_CTYPE	Definiert die Zeichenklassifikation, den Wechsel zwischen Groß- und Kleinschreibung und andere Zeichenattribute.
LC_MESSAGES	Bestimmt, in welcher Sprache Nachrichten geschrieben werden.
LINES	Bestimmt die Spaltenlänge für die Ausgabe von Auswahllisten. Auswahllisten werden ausgegeben, bis vertikal etwa zwei Drittel der im Parameter LINES angegebenen Zeilen gefüllt sind.
MAIL	Gibt den Dateipfadnamen an, der vom Postsystem verwendet wird, um neue Post zu finden. Ist dieser Parameter auf den Namen einer Nachrichtendatei gesetzt, ist der Parameter MAILPATH jedoch nicht

gesetzt, informiert die Shell den Benutzer in der angegebenen Datei über neue Post.

- MAILCHECK** Gibt (in Sekunden) an, wie oft die Shell Veränderungen der Änderungszeit in einer der Dateien überprüft, die in den Parametern **MAILPATH** oder **MAIL** angegeben wurden. Der Standardwert ist 600 Sekunden. Wenn diese Zeit abgelaufen ist, führt die Shell vor der Ausgabe der nächsten Eingabeaufforderung eine Überprüfung durch.
- MAILPATH** Gibt eine Liste mit Dateinamen an, die durch Doppelpunkte getrennt sind. Ist dieser Parameter gesetzt, informiert die Shell den Benutzer über Änderungen in den angegebenen Dateien, die innerhalb der Zeitspanne (in Sekunden) auftreten, die im Parameter **MAILCHECK** angegeben ist. Nach jedem Dateinamen kann ein ? (Fragezeichen) und eine Nachricht folgen, die ausgegeben wird. Die Nachricht wird einer Parametersubstitution unterzogen mit dem Parameter \$_, der als Name der geänderten Datei definiert ist. Die Standardnachricht ist `you have mail in $_`.
- PATH** Zeigt den Suchpfad für Befehle an, wobei es sich um eine geordnete Liste von Verzeichnispfadnamen handelt, die durch Doppelpunkte getrennt sind. Die Shell durchsucht diese Verzeichnisse in der angegebenen Reihenfolge, wenn sie nach Befehlen sucht. Eine leere Zeichenfolge an einer beliebigen Stelle in der Liste stellt das aktuelle Verzeichnis dar.
- PS1** Gibt die Zeichenfolge an, die als primäre Eingabeaufforderung verwendet werden soll. Der Wert dieses Parameters wird für die Parametersubstitution erweitert, um die primäre Eingabeaufforderungszeichenfolge zu definieren, die standardmäßig ein \$ (Dollarzeichen) ist. Das ! (Ausrufezeichen) in der primären Eingabeaufforderungszeichenfolge wird durch die Befehlsnummer ersetzt.
- PS2** Gibt den Wert der zweiten Eingabeaufforderungszeichenfolge an, die standardmäßig ein > (Größer-als-Zeichen) ist.
- PS3** Gibt den Wert der Eingabeaufforderungszeichenfolge für die Auswahl an, die innerhalb einer Schleife `select` verwendet wird, und die standardmäßig #? (Pfundzeichen, Fragezeichen) ist.
- PS4** Der Wert dieses Parameters wird für die Parametersubstitution erweitert und steht jeder Zeile der Ausführungsablaufverfolgung voran. Falls übergangen, ist die Eingabeaufforderung der Ausführungsablaufverfolgung ein + (Pluszeichen).
- SHELL** Gibt den Pfadnamen der Shell an, die in der Umgebung existiert.
- TMOUT** Gibt die Zahl der Sekunden an, die die Shell vor dem Verlassen inaktiv wartet. Ist der Parameter **TMOUT** auf einen Wert größer als 0 (Null) gesetzt, wird die Shell verlassen, wenn innerhalb der vorgegebenen Zahl an Sekunden nach Ausgabe der Eingabeaufforderung **PS1** kein Befehl eingegeben wurde. (Zu beachten ist, daß die Shell mit einem maximalen Grenzwert kompiliert werden kann, der für den hier beschriebenen Wert nicht überschritten werden darf.)
- Hinweis:** Nach Ablauf des Zeitlimits erfolgt eine 60 Sekunden lange Pause, bevor die Shell verlassen wird.
- VISUAL** Endet der Wert dieses Parameter mit `emacs`, `gmacs` oder `vi`, wird die entsprechende Option eingeschaltet.

Die Shell vergibt Standardwerte für die Parameter **PATH**, **PS1**, **PS2**, **MAILCHECK**, **TMOUT** und **IFS**, *nicht* aber für die Parameter **HOME**, **SHELL**, **ENV** und **MAIL** (obwohl der Parameter **HOME** vom Befehl **login** gesetzt wird).

Interpretation von Leerzeichen

Nach der Parametersubstitution werden die Ergebnisse nach Feldtrennzeichen durchsucht (denen, die im Parameter **IFS** definiert sind) und dort, wo solche Zeichen gefunden werden, in bestimmte Argumente eingeteilt. Explizite Nullargumente (" " oder ") werden beibehalten. Implizite Nullargumente (die aus Parametern ohne Werte resultieren) werden gelöscht.

Dateinamensubstitution in der Korn-Shell

Die Korn-Shell führt die Dateinamensubstitution durch, indem jedes Befehlsword, das in der Variable *Wort* angegeben ist, nach bestimmten Zeichen durchsucht wird. Enthält das Befehlsword die Zeichen * (Stern), ? (Fragezeichen) oder [(linke eckige Klammer) und ist die Option **-f** nicht gesetzt, sieht die Shell das Wort als Muster an. Die Shell ersetzt das Wort durch lexikographisch sortierte Dateinamen, die mit diesem Muster übereinstimmen. Findet die Shell keinen Dateinamen, der mit dem Muster übereinstimmt, wird keine Änderung vorgenommen.

Wenn die Shell ein Muster für die Dateinamensubstitution verwendet, müssen die Zeichen . (Punkt) und / (Schrägstrich) genau übereinstimmen.

Hinweis: In anderen Fällen des Abgleichs mit Mustern behandelt die Korn-Shell diese Zeichen in keiner besonderen Weise.

Die folgenden Platzhalterzeichen zeigen die entsprechenden Substitutionen an:

- * Gleich jede Zeichenfolge ab, einschließlich der leeren Zeichenfolge.
- ? Gleich jedes einzelne Zeichen ab.
- [...] Gleich jedes der eingeschlossenen Zeichen ab. Ein Zeichenpaar, das durch einen - (Bindestrich) getrennt ist, bedeutet, daß alle Zeichen, die aufgrund ihrer Position innerhalb des binären Codierungsschemas zwischen diesen beiden Zeichen liegen, ebenfalls abgeglichen werden. Ist das erste Zeichen nach der linken eckigen Klammer [ein ! (Ausrufezeichen), stimmt jedes Zeichen, das nicht eingeschlossen ist, damit überein. Ein - (Bindestrich) kann in den Zeichensatz eingefügt werden, indem er als erstes oder letztes Zeichen gesetzt wird.

Die Schreibweise **[:Zeichenklasse:]** kann ebenfalls verwendet werden, um Dateinamen innerhalb eines Bereichs abzugleichen. Mit diesem Format gleicht das System jedes einzelne Zeichen, das zu *Zeichenklasse* gehört, ab. Die Definition, welche Zeichen eine bestimmte Zeichenklasse bilden, kann über die Kategorie **LC_CTYPE** der Unteroutine **setlocale** abgerufen werden. Alle Zeichenklassen, die in den aktuellen länderspezifischen Angaben zu finden sind, werden anerkannt. Die Namen einiger Zeichenklassen sind:

alnum
alpha
cntrl
digit
graph
lower
print
punct
space

upper
xdigit.

Beispielsweise gleicht `[:upper:]` jeden Großbuchstaben ab.

Hinweis: Die Korn-Shell unterstützt keine Dateinamenerweiterung, die auf der Sortierung von Elementen, Symbolen oder Äquivalenzklassen basiert.

Eine *Musterliste* ist eine Liste mit einem oder mehreren Mustern, die mit einem senkrechten Balken (|) voneinander getrennt sind. Zusammengesetzte Muster können auf folgende Weise gebildet werden:

?(Musterliste) Gleicht wahlweise ein beliebiges der angegebenen Muster ab.

***(Musterliste)** Gleicht die angegebenen Muster ab, wenn sie kein Mal oder mehrmals auftreten.

+(Musterliste) Gleicht die angegebenen Muster ab, wenn sie einmal oder mehrmals auftreten.

@(Musterliste) Gleicht genau eines der angegebenen Muster ab.

!(Musterliste) Gleicht alles ab, außer den angegebenen Mustern.

Die Verwendung von Platzhalterzeichen unterliegt einigen Beschränkungen. Ist das erste Zeichen in einem Dateinamen ein `.` (Punkt), kann es nur mit einem Muster übereinstimmen, das auch mit einem Punkt beginnt. Beispielsweise stimmt `*` (Stern) mit den Dateinamen `meindat` und `deindat` überein, aber nicht mit den Dateinamen `.meindat` und `.deindat`. Um diese Dateien abzugleichen, ein ähnliches Muster wie das folgenden verwenden:

```
.*Datei
```

Stimmt das Muster mit keinem Dateinamen überein, wird das Muster selbst als Ergebnis des versuchten Abgleichs zurückgegeben.

Datei- und Verzeichnisnamen sollten die Zeichen `*` (Stern), `?` (Fragezeichen), `[` (linke eckige Klammer) oder `]` (rechte eckige Klammer) nicht enthalten, da dadurch eine endlose Rekursion (d. h. Endlosschleifen) während der Mustersuche verursacht werden kann.

Ein- und Ausgabe in der Korn-Shell umleiten

Bevor die Korn-Shell einen Befehl ausführt, sucht sie in der Befehlszeile nach Umadressierungszeichen. Diese Sonderzeichen weisen die Shell an, die Ein- und Ausgabe umzuleiten. Umadressierungszeichen können überall in einem einfachen Befehl vorkommen oder vor oder nach einem Befehl stehen. Sie werden nicht an den aufgerufenen Befehl übergeben.

Die Shell führt Befehls- und Parametersubstitution aus, bevor die Parameter *Wort* oder *Ziffer* verwendet werden. Ausnahmen werden aufgeführt. Dateinamenssubstitution erfolgt nur, wenn das Muster mit einer einzelnen Datei übereinstimmt und keine Interpretation von Leerzeichen ausgeführt wird.

- | | |
|-------------------------|---|
| <Wort | Verwendet die im Parameter <i>Wort</i> angegebene Datei als Standardeingabe (Dateideskriptor 0). |
| >Wort | Verwendet die im Parameter <i>Wort</i> angegebene Datei als Standardausgabe (Dateideskriptor 1). Existiert die Datei noch nicht, wird sie von der Shell erstellt. Existiert die Datei und ist die Option noclobber eingeschaltet, folgt ein Fehler; andernfalls wird die Datei auf die Feldlänge Null abgeschnitten. |
| > Wort | Wie der Befehl >Wort , außer daß diese Umadressierungsanweisung die Option noclobber überschreibt. |
| >>Wort | Verwendet die im Parameter <i>Wort</i> angegebene Datei als Standardausgabe. Existiert die Datei bereits, hängt die Shell die Ausgabe an diese Datei an (wobei sie zuerst nach dem Dateiendezeichen sucht). Existiert die Datei noch nicht, wird sie von der Shell erstellt. |
| <>Wort | Öffnet die im Parameter <i>Wort</i> angegebene Datei als Standardeingabe zum Lesen und Schreiben. |
| <<[-]Wort | Liest jede Zeile der Shell-Eingabe, bis eine Zeile lokalisiert wird, die nur den Wert des Parameters <i>Wort</i> enthält oder ein Dateiendezeichen. Die Shell führt für die angegebene Datei keine Parameter-, Befehls- oder Dateinamenssubstitution aus. Das sich ergebende Dokument, " <i>bin da</i> "-Dokument genannt, wird zur Standardeingabe. Ist irgendein Zeichen im Parameter <i>Wort</i> in Anführungszeichen gesetzt, wird für die Zeichen des Dokuments keine Interpretation durchgeführt.

Die Shell führt die Parametersubstitution für die umgeleiteten Daten aus. Um zu verhindern, daß die Shell die Zeichen \ (umgekehrter Schrägstrich), \$ (Dollarzeichen), ' (halbes Anführungszeichen) und das erste Zeichen im Parameter <i>Wort</i> interpretiert, diesen Zeichen einen umgekehrten Schrägstrich voranstellen.

Ist ein - (Minuszeichen) an << angehängt, übergeht die Shell alle führenden Tabulatorzeichen im Parameter <i>Wort</i> und im Dokument. |
| <&Ziffer | Kopiert die Standardeingabe von dem im Parameter <i>Ziffer</i> angegebenen Dateideskriptor. |
| >& Ziffer | Kopiert die Standardausgabe von dem im Parameter <i>Ziffer</i> angegebenen Dateideskriptor. |
| <&- | Schließt die Standardeingabe. |
| >&- | Schließt die Standardausgabe. |
| <&p | Überträgt die Eingabe vom Koprozeß zur Standardeingabe. |

>&p Überträgt die Ausgabe vom Koprozessor zur Standardausgabe.

Steht einer der beschriebenen Umadressierungsoptionen eine Ziffer voran, wird durch diese Ziffer die entsprechende Dateideskriptornummer angegeben (statt des Standardwerts 0 oder 1). Im folgenden Beispiel öffnet die Shell Dateideskriptor 2 als Kopie von Dateideskriptor 1 zum Schreiben:

```
... 2>&1
```

Die Reihenfolge, in der Umadressierungen angegeben werden, ist entscheidend. Die Shell wertet jede Umadressierung bezüglich der Zuordnung (*Dateideskriptor*, *Datei*) zum Zeitpunkt der Auswertung aus. So ist zum Beispiel in der Anweisung:

```
... 1>Datei 2>&1
```

der Dateideskriptor 1 der Datei zugeordnet, die im Parameter *Datei* angegeben ist. Die Shell ordnet Dateideskriptor 2 der Datei zu, die Dateideskriptor 1 (*Datei*) zugeordnet ist. Bei umgekehrter Reihenfolge der Umadressierungen würde Dateideskriptor 2 der Datenstation zugeordnet (angenommen, Dateideskriptor 1 wäre dies vorher gewesen) und Dateideskriptor 1 würde der Datei zugeordnet, die im Parameter *Datei* angegeben ist.

Folgt auf einen Befehl ein & (Et-Zeichen) und ist die Jobsteuerung nicht aktiv, ist die Standardeingabe für den Befehl die leere Datei */dev/null*. Andernfalls enthält die Umgebung für die Ausführung eines Befehls die Dateideskriptoren der aufrufenden Shell, so wie sie von Ein- und Ausgabespezifikationen geändert wurden.

Jobsteuerung in der Korn-Shell

Die Korn-Shell verfügt über eine Einrichtung zur Steuerung der Reihenfolge von Befehlen oder *Jobs*. Wird der Sonderbefehl **set -m** ausgeführt, ordnet die Korn-Shell jeder Pipeline einen Job zu. Sie führt eine Tabelle der aktuellen Jobs, die mit dem Befehl **jobs** ausgegeben wird, und ordnet ihnen kleine, ganze Zahlen zu.

Wird ein Job mit einem **&** (Et-Zeichen) asynchron gestartet, gibt die Shell eine Zeile, ähnlich der folgenden aus:

```
[1] 1234
```

Dadurch wird angezeigt, daß der Job, der asynchron gestartet wurde, die Jobnummer 1 erhielt. Ebenso wird gezeigt, daß der Job einen Prozeß (der höchsten Stufe) mit der Prozeß-ID 1234 umfaßt.

Ein Job kann mit der Tastenfolge Strg-Z abgebrochen werden. Diese Tastenfolge sendet das Signal **STOP** zum aktuellen Job. Die Shell zeigt normalerweise an, daß der Job gestoppt wurde, und zeigt dann eine Shell-Eingabeaufforderung an. Der Benutzer kann dann den Status des Jobs ändern (ihn mit dem Befehl **bg** in den Hintergrund setzen), andere Befehle ausführen und am Ende den Job mit dem Befehl **fg** wieder in den Vordergrund bringen. Die Tastenfolge Strg-Z tritt sofort in Kraft und ist insofern eine Unterbrechung, als daß die Shell anstehende Ausgaben und ungelesene Eingaben löscht, wenn die Tastenfolge eingegeben wird.

Ein Job, der im Hintergrund ausgeführt wird, stoppt beim Versuch, von der Datenstation zu lesen. Hintergrundjobs dürfen normalerweise eine Ausgabe erzeugen. Diese Option kann inaktiviert werden, indem der Befehl **stty tostop** ausgegeben wird. Ist diese Datenstationsoption gesetzt, stoppen Hintergrundjobs beim Versuch, eine Ausgabe zu erstellen oder eine Eingabe zu lesen.

Auf einen Job kann in der Korn-Shell auf unterschiedliche Weise verwiesen werden: entweder über die Prozeß-ID eines seiner Prozesse oder auf eine der folgenden Arten:

%Nummer Gibt den Job mit der bestimmten Nummer an.

%Zeichenfolge Gibt jeden Job an, dessen Befehlszeile mit der Variablen *Zeichenfolge* beginnt.

%?Zeichenfolge Gibt jeden Job an, dessen Befehlszeile die Variable *Zeichenfolge* enthält.

%% Gibt den aktuellen Job an.

%+ Äquivalent zu %%.

%- Gibt den vorherigen Job an.

Die Shell ist immer sofort informiert, wenn ein Prozeß seinen Status ändert. Normalerweise gibt sie eine Nachricht aus, sobald ein Job blockiert ist, so daß kein weiteres Fortschreiten möglich ist. Diese Nachricht erfolgt kurz vor der Ausgabe einer Eingabeaufforderung, so daß die Arbeit des Benutzers nicht gestört wird.

Ist der Überwachungsmodus eingeschaltet, löst jeder beendete Hintergrundjob Hardwareprogrammverzweigungen aus, die für das Signal **CHLD** gesetzt wurden.

Versucht der Benutzer, die Shell zu verlassen, während Jobs gestoppt sind oder ausgeführt werden, gibt das System die Warnung 'You have stopped (running) jobs.' aus. Den Befehl **jobs** verwenden, um die betroffenen Jobs festzustellen. Hierbei oder wenn der Benutzer erneut versucht, die Shell zu verlassen, beendet die Shell die gestoppten Jobs ohne Warnung.

Befehlszeileneditierung in der Korn-Shell

Normalerweise wird jede Befehlszeile über eine Datenstationseinheit eingegeben und danach die Eingabetaste gedrückt. Werden die Optionen zur Befehlszeileneditierung `emacs`, `gmacs`, oder `vi` aktiviert, kann die Befehlszeile editiert werden.

Die folgenden Befehle führen zu Editiermodi:

set -o emacs Tritt in den Editiermodus `emacs` ein und leitet eine Variante des Zeileneditors `emacs` ein.

set -o gmacs Tritt in den Editiermodus `emacs` ein und leitet eine Variante des Zeileneditors `gmacs` ein.

set -o vi Tritt in den Editiermodus `vi` ein und leitet eine Variante des Zeileneditors `vi` ein.

Eine Editieroption wird jedes Mal automatisch ausgewählt, wenn der Variablen **VISUAL** oder **EDITOR** ein Wert zugeordnet ist, der mit einem dieser Optionsnamen endet.

Hinweis: Um die Editierfunktionen verwenden zu können, muß die Datenstation des Benutzers die Eingabetaste als Rücklauf ohne Zeilenvorschub akzeptieren. Das aktuelle Zeichen auf der Anzeige muß durch ein Leerzeichen überschrieben werden.

Jeder Editiermodus öffnet an der aktuellen Zeile ein Fenster. Das Fenster ist so breit, wie in der Variable **COLUMNS**, falls definiert, angegeben; andernfalls beträgt die Satzweite 80 Zeichen. Ist die Zeile länger als das Fenster breit ist (minus zwei), macht das System den Benutzer durch eine Markierung am Ende des Fensters darauf aufmerksam. Während sich der Cursor bewegt und den Fensterrand erreicht, wird das Fenster um den Cursor zentriert. Folgende Markierungen werden angezeigt:

- > Zeigt an, daß die Zeile über die rechte Seite des Fensters hinausgeht.
- < Zeigt an, daß die Zeile über die linke Seite des Fensters hinausgeht.
- * Zeigt an, daß die Zeile über beide Seiten des Fensters hinausgeht.

Die Suchbefehle in den jeweiligen Editiermodi liefern den Zugriff auf die Protokolldatei der Korn-Shell. Es werden nur Zeichenfolgen gesucht. Ist das führende Zeichen in einer Zeichenfolge ein `^` (Winkelzeichen), muß die Übereinstimmung beim ersten Zeichen in der Zeile beginnen.

Der Editiermodus `emacs`

Der Modus `emacs` wird entweder mit der Option `emacs` oder `gmacs` aktiviert. Der einzige Unterschied zwischen diesen Modi besteht in der Ausführung des Editierbefehls `Strg-T`. Zum Editieren den Cursor zur korrekturbedürftigen Position bewegen und je nach Bedarf Zeichen oder Wörter einfügen oder löschen. Alle Editierbefehle bestehen aus Steuerzeichen oder Escape-Zeichenfolgen.

Editierbefehle werden von jeder Position in einer Zeile aus ausgeführt (nicht nur, wenn sie am Anfang der Zeile stehen). Nach Editierbefehlen nicht die Eingabetaste oder die Taste Pfeil ab drücken. Ausnahmen werden aufgeführt.

Strg-F Bewegt den Cursor ein Zeichen weiter (nach rechts).

Esc-F Bewegt den Cursor ein Wort (eine Zeichenfolge, die nur aus Buchstaben, Ziffern und Unterstreichungszeichen besteht) weiter.

Strg-B Bewegt den Cursor ein Zeichen zurück (nach links).

Esc-B	Bewegt den Cursor ein Wort zurück.
Strg-A	Bewegt den Cursor an den Anfang der Zeile.
Strg-E	Bewegt den Cursor an das Ende der Zeile.
Strg-] c	Bewegt den Cursor in der aktuellen Zeile vorwärts bis zum angegebenen Zeichen.
Esc-Strg-] c	Bewegt den Cursor in der aktuellen Zeile zurück bis zum angegebenen Zeichen.
Strg-X Strg-X	Tauscht den Cursor gegen die Markierung aus.
ERASE	Löscht das vorherige Zeichen. (Benutzerdefiniertes Löschanzeigezeichen, durch den Befehl stty definiert, normalerweise die Tastenfolge Strg-H.)
Strg-D	Löscht das aktuelle Zeichen.
Esc-D	Löscht das aktuelle Wort.
Esc-Rückschrittaste	Löscht das vorherige Wort.
Esc-H	Löscht das vorherige Wort.
Esc-Löschtaste	Löscht das vorherige Wort. Ist die Löschtaste als Unterbrechungszeichen definiert, hat dieser Befehl keine Wirkung.
Strg-T	Setzt im Modus emacs das aktuelle Zeichen zusammen mit dem nächsten Zeichen um. Setzt im Modus gmacs die zwei vorherigen Zeichen um.
Strg-C	Setzt das aktuelle Zeichen in Großbuchstaben.
Esc-C	Setzt das aktuelle Wort in Großbuchstaben.
Esc-L	Setzt das aktuelle Wort in Kleinbuchstaben.
Strg-K	Löscht die Zeichen vom Cursor bis zum Ende der Zeile. Geht ein numerischer Parameter voran, dessen Wert kleiner als die aktuelle Cursorposition ist, löscht dieser Editierbefehl die Zeichen von der angegebenen Position bis zum Cursor. Geht ein numerischer Parameter voran, dessen Wert größer als die aktuelle Cursorposition ist, löscht dieser Editierbefehl die Zeichen vom Cursor bis zur angegebenen Position.
Strg-W	Löscht die Zeichen vom Cursor bis zur Markierung.
Esc-P	Stellt den Bereich vom Cursor bis zur Markierung in den Puffer.
KILL	(Benutzerdefiniertes Löschzeichen, durch den Befehl stty definiert, normalerweise die Tastenfolge Strg-G oder ein @ (Et-Zeichen).) Löscht die gesamte aktuelle Zeile. Werden zwei Löschzeichen nacheinander eingegeben, bewirken alle danach eingegebenen Löschzeichen einen Zeilenvorschub (nützlich bei der Verwendung von Fernschreibern).
Strg-Y	Stellt den letzten aus der Zeile gelöschten Eintrag wieder her. (Stellt den Eintrag aus dem Speicher zurück in die Zeile.)
Strg-L	Schaltet die Zeile und gibt die aktuelle Zeile aus.
Strg-@	(Nullzeichen) Setzt eine Markierung.
Esc-Leerschritt	Setzt eine Markierung.

Strg-J	Führt die aktuelle Zeile aus.
Strg-M	Führt die aktuelle Zeile aus.
EOF	Verarbeitet das Dateiendezeichen (normalerweise die Tastenfolge Strg-D) nur als Dateiende, wenn die aktuelle Zeile leer ist.
Strg-P	Ruft den vorherigen Befehl ab. Bei jeder Eingabe der Tastenfolge Strg-P wird auf den zeitlich vorherigen Befehl zugegriffen. Befindet sich der Cursor nicht auf der ersten Zeile eines Befehls mit mehreren Zeilen, wird er zurückbewegt.
Esc-<	Ruft die älteste Protokollzeile ab.
Esc->	Ruft die jüngste Protokollzeile ab.
Strg-N	Ruft die nächste Befehlszeile ab. Bei jeder Eingabe der Tastenfolge Strg-N wird auf die zeitlich nächste Befehlszeile zugegriffen.
Strg-R	<i>Zeichenfolge</i> Kehrt die Suchprotokollierung für eine vorherige Befehlszeile, die die im Parameter <i>Zeichenfolge</i> angegebene Zeichenfolge enthält, um. Ist der Wert Null angegeben, verläuft die Suche vorwärts. Die angegebene Zeichenfolge wird durch Drücken der Eingabetaste oder ein Zeilenvorschubzeichen beendet. Geht der Zeichenfolge ein ^ (Winkelzeichen) voran, muß die übereinstimmende Zeile mit dem Parameter <i>Zeichenfolge</i> beginnen. Wird der Parameter <i>Zeichenfolge</i> übergangen, wird auf die nächste Befehlszeile zugegriffen, die den jüngsten Parameter <i>Zeichenfolge</i> enthält. In diesem Fall dreht der Wert Null die Richtung der Suche um.
Strg-O	Führt die aktuelle Zeile aus und ruft die nächste mit der aktuellen Zeile zusammenhängende Zeile aus der Protokolldatei auf.
Esc Ziffern	Definiert den numerischen Parameter. Die Ziffern werden als Parameter für den nächsten Befehl übernommen. Die folgenden Befehle akzeptieren einen Parameter: Strg-F, Strg-B, ERASE, Strg-C, Strg-D, Strg-K, Strg-R, Strg-P, Strg-N, Strg-J, Esc-., Esc-Strg-J, Esc-_, Esc-B, Esc-C, Esc-D, Esc-F, Esc-H, Esc-L und Esc-Strg-H .
Esc Buchstabe	(Programmfunktionssymbol) Sucht in der Alias-Namenliste nach einem Alias-Namen <i>_Buchstabe</i> . Ist so ein Alias-Name definiert, wird sein Wert in die Eingabewarteschlange gestellt. Der Parameter <i>Buchstabe</i> darf keine Escape-Funktion angeben.
Esc-] Buchstabe	(Programmfunktionssymbol) Sucht in der Alias-Namenliste nach einem Alias-Namen <i>_Buchstabe</i> . Ist so ein Alias-Name definiert, wird sein Wert in die Eingabewarteschlange gestellt. Dieser Befehl kann auf vielen Datenstationen verwendet werden, um Funktionstasten zu programmieren.
Esc-.	Fügt in der Zeile das letzte Wort des vorherigen Befehls ein. Geht ein numerischer Parameter voran, legt der Wert dieses Parameters fest, welches Wort anstelle des letzten Wortes eingefügt werden soll.
Esc-_ Esc-*	Wie die Tastenfolge ESC-.. Startet die Dateinamenssubstitution für das aktuelle Wort. Ein Stern wird angehängt, wenn das Wort mit keiner Datei übereinstimmt oder besondere Musterzeichen enthält.
Esc-Esc	Dateinamensvervollständigung. Ersetzt das aktuelle Wort durch das längste gemeinsame Präfix von allen Dateinamen, die mit dem aktuellen Wort

übereinstimmen, und hängt einen Stern an. Ist die Übereinstimmung eindeutig, wird ein / (Schrägstrich) angehängt, wenn es sich bei der Datei um ein Verzeichnis handelt. Ein Leerzeichen wird angehängt, wenn es sich bei der Datei nicht um ein Verzeichnis handelt.

Esc= Listet die Dateinamen, die mit dem aktuellen Wort übereinstimmen, so auf, als wäre ein Stern angehängt.

Strg-U Multipliziert den Parameter des nächsten Befehls mit 4.

**** Hebt die Sonderbedeutung des nächsten Zeichens auf. Editierzeichen und die Zeichen ERASE, KILL und INTERRUPT (normalerweise die Löschtaste) können in eine Befehlszeile oder in einem Suchbegriff eingegeben werden, wenn ein \ (umgekehrter Schrägstrich) vorangeht. Der umgekehrte Schrägstrich löscht, falls vorhanden, die Editierfunktionen des nächsten Zeichens.

Strg-V Zeigt die Version der Shell an.

Esc-# Fügt am Anfang der Zeile ein # (Pfundzeichen) und führt die Zeile aus. Dadurch wird ein Kommentar in der Protokolldatei eingefügt.

Editiermodus vi

Der Editiermodus vi verfügt über zwei Eingabemodi. Wird ein Befehl eingegeben, ist der Eingabemodus aktiv. Um zu editieren, muß durch Drücken der Abbruchtaste der Steuermodus aktiviert werden.

Die meisten Steuerbefehle akzeptieren vor dem Befehl einen wahlfreien Wiederholungsparameter *Zahlenwert*. Auf den meisten Systemen wird bei Aufruf des Modus vi zu Beginn die kanonische Verarbeitung aktiviert. Der Befehl wird wieder zurückgemeldet, wenn eine der folgenden Bedingungen zutrifft:

- Die Geschwindigkeit beträgt 1200 Baud oder mehr.
- Der Befehl enthält Steuerzeichen.
- Weniger als eine Sekunde ist nach Ausgabe der Eingabeaufforderung vergangen.

Das Escape-Zeichen beendet die kanonische Verarbeitung für den Rest des Befehls; danach kann die Befehlszeile geändert werden. Dieses Schema bietet den Vorteil der kanonische Verarbeitung mit einem Eingabepufferecho für unformatierten Modus. Ist die Option **vi raw** ebenfalls gesetzt, ist die kanonische Verarbeitung grundsätzlich inaktiviert. Dieser Modus ist für Systeme, die nicht zwei alternative Zeilenendebegrenzer unterstützen, implizit und kann für bestimmte Datenstationen nützlich sein,

Eingabeeditierbefehle

Hinweis: Standardmäßig befindet sich der Editor im Eingabemodus.

ERASE (Benutzerdefiniertes Löschanzeigezeichen, durch den Befehl **stty** definiert, normalerweise die Tastenfolge Strg-H oder #.) Löscht das vorherige Zeichen.

Strg-W Löscht das vorherige, durch ein Leerzeichen abgetrennte Wort.

Strg-D Beendet die Shell.

Strg-V Hebt die Sonderbedeutung des nächsten Zeichens auf. Editierzeichen, wie z. B. die Zeichen ERASE oder KILL können in eine Befehlszeile oder in einem Suchbegriff eingegeben werden, wenn die Tastenfolge Strg-V vorangeht. Die Tastenfolge Strg-V löscht, falls vorhanden, die Editierfunktionen des nächsten Zeichens.

\ Hebt die Sonderbedeutung des nächsten Zeichens ERASE oder KILL auf.

Editierbefehle für Cursorbewegung

Folgende Editierbefehle bewegen den Cursor:

- [Zahlenwert]l Bewegt den Cursor ein Zeichen weiter (nach rechts).
- [Zahlenwert]w Bewegt den Cursor ein alphanumerisches Wort weiter.
- [Zahlenwert]W Bewegt den Cursor zum Anfang des nächsten Wortes, das auf ein Leerzeichen folgt.
- [Zahlenwert]e Bewegt den Cursor zum Ende des aktuellen Wortes.
- [Zahlenwert]E Bewegt den Cursor zum Ende des aktuellen, durch ein Leerzeichen abgetrennten Wortes.
- [Zahlenwert]h Bewegt den Cursor ein Zeichen zurück (nach links).
- [Zahlenwert]b Bewegt den Cursor ein Wort zurück.
- [Zahlenwert]B Bewegt den Cursor zum vorherigen, durch ein Leerzeichen abgetrennten Wort.
- [Zahlenwert]l Bewegt den Cursor zur im Parameter *Zahlenwert* angegebenen Spalte.
- [Zahlenwert]fc Sucht das nächste Zeichen *c* in der aktuellen Zeile.
- [Zahlenwert]Fc Sucht das vorherige Zeichen *c* in der aktuellen Zeile.
- [Zahlenwert]tc Äquivalent zu *f* gefolgt von *h*.
- [Zahlenwert]Tc Äquivalent zu *F* gefolgt von *l*.
- [Zahlenwert]; Wiederholt den letzten Einzelzeichensuchbefehl *f*, *F*, *t* oder *T* so oft, wie im Parameter *Zahlenwert* angegeben.
- [Zahlenwert], Dreht den letzten Einzelzeichensuchbefehl so oft um, wie im Parameter *Zahlenwert* angegeben.
- 0 Bewegt den Cursor an den Anfang einer Zeile.
- ^ Bewegt den Cursor zum ersten belegten Zeichen in einer Zeile.
- \$ Bewegt den Cursor an das Ende einer Zeile.

Sucheditierbefehle

Sucheditierbefehle greifen auf das Befehlsprotokoll zu:

- [Zahlenwert]k Ruft den vorherigen Befehl ab.
- [Zahlenwert]- Äquivalent zum Befehl *k*.
- [Zahlenwert]j Ruft den nächsten Befehl ab. Bei jeder Eingabe des Befehls *j* wird auf den nächsten Befehl zugegriffen.
- [Zahlenwert]+ Äquivalent zum Befehl *j*.
- [Zahlenwert]G Ruft den Befehl ab, dessen Nummer im Parameter *Zahlenwert* angegeben ist. Der Standardwert ist der älteste Protokollbefehl.
- /*Zeichenfolge* Sucht rückwärts im Protokoll nach einem vorherigen Befehl, der die angegebene Zeichenfolge enthält. Die Zeichenfolge wird durch Drücken der Eingabetaste oder ein Zeilenvorschubzeichen beendet. Steht der angegebenen Zeichenfolge ein ^ (Winkelzeichen) voran, muß die übereinstimmende Zeile mit dem Parameter *Zeichenfolge* beginnen. Ist der

Wert des Parameters *Zeichenfolge* Null, wird die vorherige Zeichenfolge verwendet.

?*Zeichenfolge* Wie */Zeichenfolge*, außer daß die Suchrichtung vorwärts ist.

n Sucht nach der nächsten Übereinstimmung mit dem letzten Muster des Befehls */Zeichenfolge* oder ?.

N Sucht nach der nächsten Übereinstimmung mit dem letzten Muster des Befehls */Zeichenfolge* oder ?, allerdings in umgekehrter Richtung. Sucht im Protokoll nach der Zeichenfolge, die im vorherigen Befehl */Zeichenfolge* eingegeben wurde.

Editierbefehle zur Textänderung

Editierbefehle zur Textänderung ändern die Zeile:

a Ruft den Eingabemodus auf. Die Texteingabe erfolgt hinter dem aktuellen Zeichen.

A Hängt den Text an das Ende der Zeile an. Äquivalent zum Befehl **\$a**.

[*Zahlenwert*]**c***Bewegung*

c[*Zahlenwert*]*Bewegung*

Löscht alle Zeichen vom aktuellen Zeichen bis zu dem Zeichen, das durch den Parameter *Bewegung* als Cursorposition markiert ist, und tritt in den Eingabemodus ein. Ist der Wert des Parameters *Bewegung c*, wird die gesamte Zeile gelöscht und der Eingabemodus wird aktiviert.

C Löscht alle Zeichen vom aktuellen Zeichen bis zum Ende der Zeile und tritt in den Eingabemodus ein. Äquivalent zum Befehl **c\$**.

S Äquivalent zum Befehl **cc**.

D Löscht alle Zeichen vom aktuellen Zeichen bis zum Ende der Zeile. Äquivalent zum Befehl **d\$**.

[*Zahlenwert*]**d***Bewegung*

d[*Zahlenwert*]*Bewegung*

Löscht alle Zeichen vom aktuellen Zeichen bis einschließlich dem im Parameter *Bewegung* angegebenen Zeichen. Ist der Wert des Parameters *Bewegung d*, wird die gesamte Zeile gelöscht.

i Ruft den Eingabemodus auf. Die Texteingabe erfolgt vor dem aktuellen Zeichen.

I Fügt vor dem ersten Zeichen in der Zeile Text ein. Äquivalent zum Befehl **Oi**.

[*Zahlenwert*]**P** Stellt die vorherige Textänderung vor den Cursor.

[*Zahlenwert*]**p** Stellt die vorherige Textänderung hinter den Cursor.

R Tritt in den Eingabemodus und überschreibt die Zeichen der Zeile.

[*Zahlenwert*]**rc** Ersetzt die im Parameter *Zahlenwert* angegebene Zahl von Zeichen, angefangen vom aktuellen Zeichen, durch die im Parameter *c* angegebenen Zeichen. Dieser Befehl bewegt außerdem den Cursor nach dem Ersetzen der Zeichen nach vorn.

[*Zahlenwert*]**x** Löscht das aktuelle Zeichen.

[*Zahlenwert*]**X** Löscht das vorherige Zeichen.

[*Zahlenwert*]**.** Wiederholt den vorherigen Befehl zur Textänderung.

- [Zahlenwert]~ Kehrt die Groß-/Kleinschreibung der im Parameter *Zahlenwert* angegebenen Zahl von Zeichen um, angefangen von der aktuellen Cursorposition, wobei der Cursor nach vorn bewegt wird.
- [Zahlenwert]_ Hängt das im Parameter *Zahlenwert* des vorherigen Befehls angegebene Wort an und tritt in den Eingabemodus ein. Wird der Parameter *Zahlenwert* übergangen, wird das letzte Wort verwendet.
- * Hängt einen * (Stern) an das aktuelle Wort an und startet die Dateinamenssubstitution. Wird keine Übereinstimmung gefunden, ertönt ein Signalton. Anderfalls wird das Wort durch das passende Muster ersetzt und der Eingabemodus aktiviert.
- \ Dateinamenvervollständigung. Ersetzt das aktuelle Wort durch das längste gemeinsame Präfix von allen Dateinamen, die mit dem aktuellen Wort übereinstimmen, und hängt einen Stern an. Ist die Übereinstimmung eindeutig, wird ein / (Schrägstrich) angehängt, wenn es sich bei der Datei um ein Verzeichnis handelt. Ein Leerzeichen wird angehängt, wenn es sich bei der Datei nicht um ein Verzeichnis handelt.

Sonstige Editierbefehle

Sonstige Editierbefehle sind:

[Zahlenwert]y*Bewegung*

y[Zahlenwert]*Bewegung*

Stellt alle Zeichen vom aktuellen Zeichen bis einschließlich dem Zeichen, das durch den Parameter *Bewegung* als Cursorposition markiert ist, in den Löschpuffer. Text und Cursor bleiben unverändert.

Y Stellt die Zeichen von der aktuellen Position bis zum Ende der Zeile in den Speicher. Äquivalent zum Befehl **y\$**.

u Hebt den letzten textverändernden Befehl auf.

U Hebt alle auf die betroffene Zeile ausgeführten textverändernden Befehle auf.

[Zahlenwert]v Gibt den Befehl `fc -e ${VISUAL:-${EDITOR:-vi}}` *Zahlenwert* in den Eingabepuffer zurück. Wird der Parameter *Zahlenwert* übergangen, wird die aktuelle Zeile verwendet.

Strg-L Schaltet die Zeile und gibt die aktuelle Zeile aus. Dieser Befehl ist nur im Steuermodus wirksam.

Strg-J Führt unabhängig vom Modus die aktuelle Zeile aus.

Strg-M Führt unabhängig vom Modus die aktuelle Zeile aus.

Sendet die Zeile, nachdem ein # (Pfundzeichen) am Anfang der Zeile eingefügt wurde. Dieser Befehl ist sinnvoll, wenn die aktuelle Zeile, ohne daß sie ausgeführt wird, in das Protokoll eingefügt werden soll.

= Listet die Dateinamen, die mit dem aktuellen Wort übereinstimmen, so auf, als wäre ein Stern angehängt.

@Buchstabe Sucht in der Alias-Namenliste nach einem Alias-Namen *_Buchstabe*. Ist ein solcher Alias-Name definiert, wird sein Wert zur Verarbeitung in die Eingabewarteschlange gesetzt.

Empfohlene Dokumentation

Zugehörige Informationen

Die Abschnitte zu den Befehlen **ksh** und **stty**.

Die Abschnitte zu den Sonderbefehlen der Korn-Shell **alias**, **cd**, **export**, **fc**, **getopts**, **read**, **set** und **typeset**.

Die Datei **/etc/passwd**.

Anhang B. Bourne-Shell

Bei der Bourne-Shell handelt es sich um einen interaktiven Befehls-Interpreter und eine befehlsorientierte Programmiersprache. Mit dem Befehl **bsh** wird die Bourne-Shell aufgerufen.

Verwendung der Bourne-Shell

Die Bourne-Shell kann entweder als Anmelde-Shell oder als Subshell unter der Anmelde-Shell ausgeführt werden. Nur mit dem Befehl **login** kann die Bourne-Shell als Anmelde-Shell aufgerufen werden. Dazu wird dazu ein spezielles Format des Befehlsnamens **bsh**: `-bsh` verwendet. Wird die Shell mit einem vorangestellten `-` (Minuszeichen) aufgerufen, liest die Shell erst Befehle aus der Systemdatei `/etc/profile` und der Datei `$HOME/.profile`, falls vorhanden, und führt diese aus. Danach akzeptiert die Shell Befehle, wie sie im Abschnitt über Optionen beschrieben werden. Ist der Benutzer angemeldet und arbeitet unter der Anmelde-Shell, kann er die Bourne-Shell mit dem Befehl `bsh` aufrufen. Von diesem Befehl wird eine Subshell, d.h. eine zweite Shell als Sohn der Anmelde-Shell, ausgeführt.

Ist beim Aufrufen der Bourne-Shell der Parameter *Datei* [*Parameter*] angegeben, führt die Shell die im Parameter *Datei* angegebene Prozedurdatei mit allen angegebenen Parametern aus. Die angegebene Prozedurdatei muß Leseberechtigung haben; die Einstellungen **setuid** und **setgid** werden ignoriert. Dann werden die Befehle gelesen. Eine Prozedurdatei sollte bei der Verwendung der Option `-c` oder `-s` nicht angegeben werden.

Bei der Anmeldung wird die Shell aufgerufen, um die Befehle des Benutzers zu lesen. Vorher prüft die Shell, ob die Datei `/etc/profile` im System existiert. Ist dies der Fall, werden zunächst aus dieser Datei Befehle gelesen (in dieser Datei werden die für alle Benutzer notwendigen Variablen gesetzt). Anschließend sucht die Shell nach der Datei `.profile` im Anmeldeverzeichnis des Benutzers. Ist diese Datei vorhanden, führt die Shell die Befehle von dort aus aus. Danach ist die Shell bereit, Befehle aus der Standardeingabe zu lesen.

Die Shell-Umgebung

Die *Umgebung* eines Befehls setzt sich aus allen Variablen (mit den zugeordneten Werten) zusammen, die dem Befehl zu Beginn der Ausführung bekannt sind. Zu der Umgebung gehören auch Variablen, die ein Befehl von seinem Vaterprozeß übernimmt, und Variablen, die als Schlüsselwortparameter in die Befehlszeile eingegeben werden, über die der Befehl aufgerufen wird.

Von der Shell werden die Variablen an die Sohnprozesse übergeben, die als Argumente für den integrierten Befehl **export** benannt werden. Dieser Befehl stellt die benannten Variablen in die Umgebungen der Shell und all ihrer zukünftigen Sohnprozesse.

Bei Schlüsselwortparametern handelt es sich um Zuordnungen von Variable/Wert-Paaren, die normalerweise vor dem Prozedurnamen in einer Befehlszeile stehen (siehe aber auch die Option für den Befehl **set**). Solche Variablen werden in die Umgebung der aufgerufenen Prozedur gestellt.

Es folgt als Beispiel eine einfache Prozedur, in der die Werte von zwei Variablen gezeigt werden (die in einer Befehlsdatei mit dem Namen `key_command` gespeichert sind):

```
# key_command
echo $a $b
```

Die folgenden Befehlszeilen erzeugen die dargestellte Ausgabe:

Eingabe

```
a=key1 b=key2 key_command  
a=tom b=john key_command
```

Ausgabe

```
key1 key2  
tom john
```

Die Schlüsselwortparameter einer Prozedur sind bei dem in \$# gespeicherten Parameteradditionswert nicht eingeschlossen.

Eine Prozedur kann auf die Werte beliebiger Variablen in ihrer Umgebung zugreifen. Verändert sie irgend einen dieser Werte, hat das jedoch keine Auswirkungen auf die Shell-Umgebung. Sie gelten nur für die betreffende Prozedur. Damit die Änderungen in die Umgebung gestellt werden, die von der Prozedur an die Sohnprozesse übergeben wird, müssen die neuen Werte von dieser Prozedur exportiert werden.

Um eine Liste der Variablen zu erhalten, die aus der aktuellen Shell exportiert werden können, folgendes eingeben:

```
export
```

Es kann auch eine Liste von schreibgeschützten Variablen aufgerufen werden. Um eine Liste der Name/Wert-Paare in der aktuellen Umgebung zu erhalten, folgendes eingeben:

```
env
```

Weitere Informationen über Benutzerumgebungen enthält die Umgebungsdatei **environment**.

Verwendung der eingeschränkten Shell

Mit Hilfe der eingeschränkten Shell werden Anmeldenamen und Ausführungsumgebungen installiert, deren Leistungsspektrum geringer ist als das der regulären Bourne-Shell. Mit dem Befehl **Rsh** wird die eingeschränkte Shell geöffnet. Die Aktionen des Befehls **Rsh** entsprechen denen des Befehls **bsh** mit der Ausnahme, daß die folgenden Aktionen nicht zulässig sind:

- das Verzeichnis wechseln (mit dem Befehl **cd**)
- Werte für die Variablen **PATH** oder **SHELL** setzen
- Pfad- oder Befehlsnamen angeben, die einen / (Schrägstrich) enthalten
- die Ausgabe umleiten.

Stellt die eingeschränkte Shell fest, daß es sich bei dem auszuführenden Befehl um eine Shell-Prozedur handelt, ruft sie zur Ausführung des Befehls die Bourne-Shell auf. Auf diese Weise ist es möglich, daß der Endbenutzer über Shell-Prozeduren verfügt, die auf das gesamte Leistungsspektrum der Bourne-Shell zugreifen, ohne daß ein umfassendes Befehlsmenü notwendig ist. Dieses Schema setzt voraus, daß der Endbenutzer in demselben Verzeichnis nicht über Schreib- und Ausführungsberechtigungen verfügt.

Ist beim Aufrufen der Bourne-Shell der Parameter *Datei* [*Parameter*] angegeben, führt die Shell die im Parameter *Datei* angegebene Prozedurdatei mit allen angegebenen Parametern aus. Die angegebene Prozedurdatei muß Leseberechtigung besitzen. Die Einstellungen für **setuid** und **setgid** (Definitionen für Benutzer- bzw. Gruppen-ID) werden ignoriert. Danach liest die Shell die Befehle.

Hinweis: Bei Verwendung der Option **-c** oder **-s** sollte keine Prozedurdatei angegeben werden.

Die Shell führt nach der Interpretation der Dateien **.profile** und **/etc/environment** Beschränkungen ein. So kann der Autor der Datei **.profile** Benutzeraktionen steuern, indem er Konfigurationsaktionen ausführt und den Benutzer in ein bestimmtes Verzeichnis stellt (möglicherweise nicht das Anmeldeverzeichnis). Ein Administrator kann ein Verzeichnis mit Befehlen im Verzeichnis **/usr/rbin** definieren, das vom Befehl **Rsh** aufgerufen wird.

Wird die eingeschränkte Shell unter dem Namen **-Rsh** aufgerufen, liest sie die Datei **.profile** des Benutzers (**\$HOME/.profile**). Sie verhält sich dabei wie die reguläre Bourne-Shell, mit der Ausnahme, daß durch eine Unterbrechung die Shell sofort verlassen wird, anstatt zur Befehlsebene zurückzukehren.

Beim Aufruf bestimmen Einzelbyteversionen der Anmelde-Shell, ob es sich um eine Einzelbyte- oder Mehrbyteumgebung handelt (wie durch die Umgebungsvariable **LANG** in der Datei **/etc/environment** definiert). Im Falle einer Mehrbyteumgebung ruft die Shell ihre Mehrbyteversion auf. Im Falle einer Einzelbyteumgebung fährt die Shell mit der Ausführung ihrer Einzelbyteversion fort.

Um die Mehrbyteversion der Anmelde-Shell in einer Einzelbyteumgebung auszuführen, muß die Shell mit dem vollständigen Pfadnamen der Mehrbyteversion aufgerufen werden. Diese Version befindet sich in der Datei **/etc/passwd**. (Nähere Informationen, wie dieser Wert geändert werden kann, siehe den Abschnitt zum Befehl **chuser**.) Eine weitere Möglichkeit besteht darin, den Wert der Umgebungsvariable **LANG** in der Einzelbyteversion der Anmelde-Shell zu ändern und die Shell erneut aufzurufen.

Verwendung von Bourne-Shell-Befehlen

Wird in der Bourne-Shell ein Befehl eingegeben, wertet die Shell den Befehl zunächst aus und führt alle angegebenen Substitutionen aus. Dann wird der Befehl ausgeführt, vorausgesetzt:

- der Befehlsname ist ein Sonderbefehl der Bourne-Shell

ODER

- der Befehlsname stimmt mit dem Namen einer definierten Funktion überein. In diesem Fall setzt die Shell für die Positionsparameter die Parameter der Funktion ein.

Stimmt ein Befehlsname weder mit einem integrierten Befehl noch dem Namen einer definierten Funktion überein, handelt es sich aber um eine ausführbare Datei, die ein kompiliertes (binäres) Programm ist, erzeugt die Shell (als *Vater*) einen neuen (*Sohn*-) Prozeß, der das Programm sofort ausführt. Ist die Datei zwar als ausführbar markiert, aber kein kompiliertes Programm, behandelt die Shell sie wie eine Shell-Prozedur. In diesem Fall erzeugt die Shell ein weiteres Exemplar (eine *Subshell*), um die Datei zu lesen und die darin enthaltenen Befehle auszuführen. Die Shell führt auch Klammerbefehle in einer Subshell aus. Für den Endbenutzer sieht es so aus, als ob ein kompiliertes Programm genauso ausgeführt wird wie eine Shell-Prozedur.

Die Shell sucht normalerweise an vier Stellen im Dateisystem nach Befehlen und zwar in folgender Reihenfolge:

1. Verzeichnis **/bin**
2. Verzeichnis **/usr/bin**
3. Verzeichnis **/etc**
4. Aktuelles Verzeichnis.

Die Shell durchsucht nacheinander jedes Verzeichnis, bis der gesuchte Befehl gefunden ist.

Hinweis: Die Reihenfolge, in der die Shell die Verzeichnisse durchsucht, ist durch die Variable **PATH** festgelegt. Die oben aufgeführte Reihenfolge wird eingehalten, wenn die Variable **PATH** auf **PATH=/bin:/usr/bin:/etc:** gesetzt ist. Diese Reihenfolge der Verzeichnisse kann durch Zurücksetzen der Variable **PATH** geändert werden.

Wenn beim Aufrufen eines Befehls ein bestimmter Pfadname angegeben wird (zum Beispiel **/usr/bin/sort**), sucht die Shell nur in diesem angegebenen Verzeichnis. Enthält der Befehlsname einen **/** (Schrägstrich), verwendet die Shell den Suchpfad nicht.

Der Benutzer kann einen vollständigen Pfadnamen angeben, der mit dem Stammverzeichnis beginnt (wie z. B. **/usr/bin/sort**). Er kann ebenso einen Pfad anlegen, der auf das aktuelle Verzeichnis Bezug nimmt. Wird beispielsweise:

```
bin/meindat
```

angegeben, sucht die Shell im aktuellen Verzeichnis nach einem Verzeichnis **bin** und in diesem Verzeichnis nach der Datei **meindat**.

Hinweis: Die eingeschränkte Shell führt keine Befehle aus, die einen **/** (Schrägstrich) enthalten.

Die Shell speichert den Standort im Suchpfad von jedem ausgeführten Befehl (um später unnötige **exec**-Befehle zu vermeiden). Findet sie den Befehl in einem relativen Verzeichnis (ein Verzeichnis, dessen Name nicht mit **/** beginnt), muß die Shell den Standort eines Befehls jedes Mal, wenn das aktuelle Verzeichnis wechselt, neu festlegen. Wenn die

Variable **PATH** geändert oder der Befehl **hash -r** ausgeführt wird, speichert die Shell die Befehlsstandorte nicht mehr.

Quotierungszeichen

Viele Zeichen haben für die Shell eine Sonderbedeutung. Es kommt allerdings vor, daß diese Bedeutung unterdrückt werden soll. Mit Anführungszeichen (") und halben Anführungszeichen ('), die um Zeichenfolgen gesetzt werden, oder einem umgekehrten Schrägstrich (\) vor einem einzelnen Zeichen können diese Zeichen ohne ihre Sonderbedeutung verwendet werden.

Im folgenden Beispiel werden alle Zeichen außer den einschließenden einfachen Anführungszeichen ohne Sonderbedeutung interpretiert. Der Befehl:

```
stuff='echo $? $*; ls * | wc'
```

ordnet die Literalzeichenfolge `echo $? $*; ls * | wc` der Variablen `stuff` zu. Die Shell führt weder die Befehle **echo**, **ls** und **wc** aus, noch erweitert sie die Variablen `?` und `*` und das Sonderzeichen `*` (Stern).

Zwischen Anführungszeichen bleibt die besondere Bedeutung des `$` (Dollarzeichen), `'` (Gravis) und `"` (Anführungszeichen) wirksam, wohingegen alle anderen Zeichen buchstäblich gesehen werden. Daher werden, wenn zwischen Anführungszeichen gesetzt, Befehls- und Variablensubstitutionen ausgeführt. Zudem haben Anführungszeichen auf Befehle innerhalb der Befehlssubstitution, die Teil der Zeichenfolge in Anführungszeichen sind, keine Auswirkungen, so daß bei diesen Zeichen die Sonderbedeutung erhalten bleibt.

Die Folge:

```
ls *
```

```
Datei1 Datei2 Datei3
```

```
message="Dieses Verzeichnis enthält 'ls * ' " echo $message
```

```
Dieses Verzeichnis enthält Datei1 Datei2 Datei3
```

zeigt, daß das Sonderzeichen `*` in der Befehlssubstitution erweitert wurde.

Um die Sonderbedeutung des `$` (Dollarzeichen), `'` (Gravis) und `"` (Anführungszeichen) innerhalb von Anführungszeichen aufzuheben, diesen Zeichen einen umgekehrten Schrägstrich (\) vorzustellen. Werden keine Anführungszeichen verwendet, hat das Vorstellen eines umgekehrten Schrägstrichs dieselbe Bedeutung, als wenn das Zeichen zwischen halbe Anführungszeichen gesetzt würde. Daher wird, wenn ein umgekehrter Schrägstrich unmittelbar vor ein Zeilenvorschubzeichen gestellt wird (d. h. ein umgekehrter Schrägstrich am Ende der Zeile), das Zeilenvorschubzeichen aufgehoben, und die Befehlszeile kann auf der nächsten physischen Zeile fortgesetzt werden.

Signalverarbeitung

Die Shell ignoriert die Signale **INTERRUPT** und **QUIT** für einen aufgerufenen Befehl, wenn der Befehl mit einem `&` (Et-Zeichen) abgeschlossen wurde; d. h., der Befehl im Hintergrund ausgeführt wird. Andernfalls haben Signale die Werte, die die Shell von ihrem Vater übernommen hat, mit Ausnahme des Signals 11. Weitere Informationen enthält der Abschnitt zum integrierten Bourne-Shell-Befehl **trap**.

Folgende zusammengesetzte Befehle gibt es in der Bourne-Shell:

- Pipeline (ein oder mehrere einfache Befehle, die durch das Symbol `|` getrennt sind)
- Liste mit einfachen Befehlen
- Befehl, der mit einem reservierten Wort beginnt

- Befehl, der mit dem Steueroperator (linke runde Klammer) beginnt.

Reservierte Wörter

Die folgenden reservierten Wörter werden nur erkannt, wenn sie ohne Anführungszeichen als erstes Wort eines Befehls erscheinen:

if	for
then	while
else	until
elif	do
fi	done
case	{
esac	}

Falls nicht anders angegeben, ist der Wert, der von einem zusammengesetzten Befehl zurückgegeben wird, der des letzten ausgeführten einfachen Befehls.

for *Kennung* [**in** *Wort* . . .] **do** *Liste* **done**

Setzt den Parameter *Kennung* auf das Wort (oder die Wörter), das im Parameter *Wort* angegeben ist und führt die im Parameter *Liste* angegebenen Befehle aus. Wird der Parameter **in** *Wort* . . . übergangen, führt der Befehl **for** den Parameter *Liste* für jeden gesetzten Positionsparameter aus. Die Ausführung ist beendet, wenn die Liste keine weiteren Wörter enthält.

case *Wort in* *Muster* [*Muster*] . . .) *Liste*;; [. . . *Muster* [*Muster*] . . .) *Liste*;;] **esac**

Führt die im Parameter *Liste* angegebenen Befehle aus, die dem ersten Parameter *Muster* zugeordnet sind, der mit dem Wert des Parameters *Wort* übereinstimmt. Dabei wird dieselbe zeichenabgleichende Schreibweise in Mustern verwendet wie bei der Dateinamenssubstitution, außer, daß keine unbedingte Übereinstimmung bei einem / (Schrägstrich), führenden (Punkt) oder einem Punkt, der einem Schrägstrich unmittelbar folgt, vorliegen muß.

if *Liste* **then** *Liste* [**elif** *Liste*] . . . [**else** *Liste*]**fi**

Führt den Parameter *Liste* aus, der auf das Schlüsselwort **if** folgt. Gibt der Befehl den Rückkehrcode Null zurück, führt die Shell den Parameter *Liste* aus, der dem ersten Befehl **then** folgt. Andernfalls führt sie, falls vorhanden, den Parameter *Liste* aus, der dem Befehl **elif** folgt. Ist dessen Rückkehrcode gleich Null, führt die Shell den nächsten Befehl **then** aus. Schlägt dessen Ausführung fehl, führt sie den Parameter *Liste* aus, der dem Befehl **else** folgt. Werden keine Befehle **else** *Liste* oder **then** *Liste* ausgeführt, gibt der Befehl **if** den Rückkehrcode Null zurück.

while *Liste* **do** *Liste* **done**

Führt den Parameter *Liste* aus, der dem Befehl **while** folgt. Ist der Rückkehrcode des letzten Befehls in der Liste Null, führt die Shell den Parameter *Liste* aus, der dem Befehl **do** folgt. Sie fährt mit dieser Schleife durch die Listen so lange fort, bis der Rückkehrcode des letzten Befehls im Parameter **while** *Liste* ungleich Null ist. Werden keine Befehle in der Liste des Befehls **do** *Liste* ausgeführt, gibt der Befehl **while** den Rückkehrcode Null zurück.

until *Liste* **do** *Liste* **done**

Führt den Parameter *Liste* aus, der dem Befehl **until** folgt. Ist der Rückkehrcode des letzten Befehls in der Liste ungleich Null, führt die Shell den Parameter *Liste* aus, der dem Befehl **do** folgt. Sie fährt mit dieser Schleife durch die Listen so lange fort, bis der Rückkehrcode des letzten Befehls im Parameter **until** *Liste* gleich Null ist. Werden keine Befehle in

der Liste des Befehls **do** *Liste* ausgeführt, gibt der Befehl **until** den Rückkehrcode Null zurück.

(*Liste*) Führt die Befehle im Parameter *Liste* in einer Subshell aus.

{ *Liste*; } Führt die Befehle im Parameter *Liste* innerhalb des aktuellen Shell-Prozesses aus, ohne eine Subshell zu starten.

Name () { *Liste*; }
Definiert eine Funktion, auf die durch den Parameter *Name* verwiesen wird. Der Hauptteil der Funktion ist die Liste von Befehlen, die zwischen geschweiften Klammern im Parameter *Liste* angegeben wird.

Sonderbefehle in der Bourne-Shell

Sonderbefehle sind in die Bourne-Shell integriert und werden im Shell-Prozeß ausgeführt. Wenn nicht anders angegeben, wird die Ausgabe in den Dateideskriptor 1 geschrieben und ist der Ausführungsstatus Null, es sei denn, der Befehl enthält Syntaxfehler. Ein- und Ausgabeumadressierung ist erlaubt.

Die folgenden Sonderbefehle werden etwas anders verarbeitet als andere Sonderbefehle:

:	(Doppelpunkt)	exec	shift
.	(Punkt)	exit	times
break		export	trap
continue		readonly	wait
eval		return	

Die Bourne-Shell verarbeitet diese Befehle wie folgt:

- Parameterzuordnungslisten, die dem Befehl vorangehen, bleiben gültig, wenn der Befehl beendet ist.
- E/A-Umadressierungen werden nach der Parameterzuordnung verarbeitet.
- Fehler in der Shell-Prozedur führen dazu, daß die Prozedur die Verarbeitung stoppt.

Zusätzlich werden Wörter, die im Format einer Parameterzuordnung den Sonderbefehlen **export** und **readonly** folgen, mit denselben Regeln erweitert wie Parameterzuordnungen.

Liste der Sonderbefehle

Die folgenden Befehle sind in die Bourne-Shell integriert:

: Gibt den Rückkehrcode Null zurück.

. *Datei* Liest Befehle aus dem Parameter *Datei* und gibt sie aus. Startet keine Subshell. Die Shell verwendet den in der Variablen **PATH** angegebenen Suchpfad, um das Verzeichnis zu finden, daß die angegebene Datei enthält.

break [*n*] Verläßt die Programmschleifen **for**, **while** oder **until**, falls vorhanden. Wird die Variable *n* angegeben, unterbricht der Befehl **break** die Anzahl der Ebenen, die durch die Variable *n* angegeben werden.

continue [*n*] Fährt mit der nächsten Iteration einer Programmschleife **for**, **while** oder **until** fort. Wird die Variable *n* angegeben, fährt der Befehl mit *n*ten Programmschleife fort.

cd [*Verzeichnis*]
Wechselt vom aktuellen Verzeichnis zum angegebenen Verzeichnis. Wird kein Verzeichnis angegeben, wird der Wert der Shell-Variablen **HOME**

verwendet. Die Shell-Variable **CDPATH** definiert den Suchpfad für das angegebene Verzeichnis. Alternative Verzeichnisnamen werden in einer durch einen Doppelpunkt abgetrennten Liste aufgeführt. Ein leerer Pfadname gibt das aktuelle Verzeichnis an (das der Standardpfad ist). Dieser leere Pfadname erscheint unmittelbar hinter dem Gleichheitszeichen in der Zuordnung oder zwischen begrenzenden Doppelpunkten an einer anderen Stelle in der Pfadliste. Beginnt das angegebene Verzeichnis mit einem / (Schrägstrich), verwendet die Shell den Suchpfad nicht. Andernfalls wird jedes Verzeichnis im Pfad durchsucht.

Hinweis: Die eingeschränkte Shell kann den Shell-Befehl **cd** nicht ausführen.

echo [*Argument* . . .]

Schreibt Argumente in die Standardausgabe. Der Abschnitt zum Befehl **echo** enthält nähere Informationen zur Verwendung und zu Parametern. Die Option **-n** wird nicht unterstützt.

eval [*Argument* . . .]

Liest die Argumente als Eingabe für die Shell und führt die daraus resultierenden Befehle aus.

exec [*Argument* . . .]

Führt anstelle der Shell den im Parameter *Argument* angegebenen Befehl aus, ohne einen neuen Prozeß zu erstellen. Ein- und Ausgabeargumente können auftreten, die, wenn keine anderen Argumente auftreten, die Shell-Ein- oder -Ausgabe ändern können. Dies ist im Falle der Anmelde-Shell nicht empfehlenswert.

exit [*n*]

Verläßt die Shell mit dem im Parameter *n* angegebenen Rückkehrcode. Wird dieser Parameter übergangen, ist der Rückkehrcode der des letzten ausgeführten Befehls (die Tastenfolge Strg-D führt ebenfalls zum Verlassen der Shell). Der Wert des Parameters *n* kann zwischen 0 und 255 einschließlich liegen.

export [*Name* . . .]

Markiert die angegebenen Namen für den automatischen Export in die Umgebung von nachfolgend ausgeführten Befehlen. Wird der Parameter *Name* nicht angegeben, zeigt der Befehl **export** eine Liste aller Namen an, die in dieser Shell exportiert wurden. Exportfunktionen können nicht exportiert werden.

hash [**-r**] [*Befehl* . . .]

Sucht und speichert den Standort im Suchpfad für jeden angegebenen Befehl. Die Option **-r** führt dazu, daß die Shell die gespeicherten Standorte wieder übergeht. Wird keine Option oder kein Befehl angegeben, zeigt die Shell Informationen über die gespeicherten Befehle im folgenden Format an:

Anzahl	Suche	Befehl
--------	-------	--------

Anzahl zeigt an, wie oft ein Befehl vom Shell-Prozeß ausgeführt wurde. Suche ist die Einheit für die Arbeit, die für das Lokalisieren eines Befehls im Suchpfad erforderlich war. Befehl zeigt die Pfadnamen der angegebenen Befehle an. Unter gewissen Umständen ist es erforderlich, daß der gespeicherte Standort eines Befehls neu ermittelt werden muß, beispielsweise der Standort eines relativen Pfadnamens, wenn das aktuelle Verzeichnis wechselt. In Frage kommende Befehle werden durch einen *

(Stern) neben der Angabe *Anzahl* gekennzeichnet. Der Wert *Suche* erhöht sich durch die erneute Ermittlung des Standortes.

pwd Zeigt das aktuelle Verzeichnis an. Weitere Informationen zu den Optionen enthält der Artikel zum Befehl **pwd**.

read [*Name* . . .]

Liest eine Zeile der Standardeingabe und ordnet das erste Wort dieser Zeile dem ersten Parameter *Name* zu, das zweite Wort dem zweiten Parameter *Name* usw. Verbleibende Wörter werden dem letzten Parameter *Name* zugeordnet. Dieser Befehl gibt den Wert Null zurück, wenn er nicht auf ein Dateieindezeichen trifft.

readonly [*Name* . . .]

Markiert die im Parameter *Name* angegebenen Namen als schreibgeschützt. Die Werte dieser Namen können nicht zurückgesetzt werden. Werden keine Namen angegeben, zeigt der Befehl **readonly** eine Liste aller schreibgeschützten Namen an.

return [*n*]

Führt dazu, daß eine Funktion mit dem Rückgabewert *n* verlassen wird. Wird keine Variable *n* angegeben, gibt die Funktion den Status des letzten innerhalb dieser Funktion ausgeführten Befehls zurück. Dieser Befehl ist nur gültig, wenn er innerhalb einer Shell-Funktion ausgeführt wird.

set [*Option* [*Argument*] . . .]

Setzt eine oder mehrere der folgenden Optionen:

-a

Markiert alle Variablen, für die eine Zuordnung ausgeführt wurde, für den Export. Steht die Zuordnung einem Befehlsnamen voran, ist das Exportattribut nur für die Ausführungsumgebung dieses Befehls gültig, es sei denn, die Zuordnung steht einem der integrierten Sonderbefehle voran. In diesem Fall bleibt das Exportattribut nach der Ausführung des integrierten Befehls bestehen. Steht die Zuordnung keinem Befehlsnamen voran oder ist sie das Verarbeitungsergebnis des Befehls **getopts** oder **read**, bleibt das Exportattribut bestehen, bis die Variable inaktiviert wird.

-C

Verhindert, daß vorhandene Dateien vom Shell-Umadressierungsoperator **>** überschrieben werden. Für einzelne Dateien kann diese Option durch Verwendung des Umadressierungsoperators **>|** überschrieben werden.

-e

Bricht die Ausführung sofort ab, wenn für einen Befehl alle der folgenden Bedingungen zutreffen:

- Der Befehl wird mit einem Rückgabewert größer als Null verlassen.
- Er ist nicht Teil der zusammengesetzten Listen für die Befehle **while**, **until** oder **if**.
- Er wurde nicht unter Verwendung der AND-Liste oder OR-Liste getestet.
- Es handelt sich bei dem Befehl nicht um eine Pipeline, der das reservierte Wort **!** (Ausrufezeichen) voransteht.

-f

Inaktiviert die Dateinamensubstitution.

-h

Sucht und speichert die Befehle, die innerhalb von Funktionen aufgerufen werden, während diese Funktionen definiert werden. (Normalerweise werden diese Befehle bei der Ausführung der Funktion lokalisiert, siehe Befehl **hash**.)

- k** Stellt alle Schlüsselwortparameter in die Umgebung eines Befehls, nicht nur die, die dem Befehlsnamen voranstehen.
- n** Liest Befehle, aber führt sie nicht aus. Die Option **-n** kann verwendet werden, um die Shell-Prozedur auf Syntaxfehler hin zu überprüfen. Möglicherweise wird diese Option von einer interaktiven Shell ignoriert.
- t** Bricht die Ausführung ab, nachdem ein Befehl gelesen und ausgeführt wurde.
- u** Behandelt eine nicht gesetzte Variable als Fehler und bricht die Ausführung der Variablensubstitution ab. Eine interaktive Shell bricht nicht ab.
- v** Zeigt Shell-Eingabezeilen so an, wie sie gelesen werden.
- x** Zeigt Befehle und ihre Argumente an, bevor sie ausgeführt werden.
- Ändert keine Option. Nützlich, wenn der Positionsparameter **\$1** für eine Zeichenfolge gesetzt werden soll, die mit einem **-** (Minuszeichen) beginnt.

Geht einer Option ein **+** (Pluszeichen) anstelle eines **-** (Minuszeichen) voran, wird diese inaktiviert. Die angeführten Optionen können auch in der Befehlszeile der Shell angegeben werden. Die Sondervariable **\$-** enthält den aktuellen Satz von Optionen.

Alle Argumente des Befehls **set** sind Positionsparameter und werden in Reihenfolge **\$1**, **\$2** usw. zugeordnet. Werden keine Optionen oder Argumente angegeben, zeigt der Befehl **set** alle Namen an.

shift [n]

Schiebt die Befehlszeilenargumente nach links, d. h., ordnet die Werte der Positionsparameter neu zu, indem der aktuelle Wert von **\$1** gelöscht wird und der Wert von **\$2** **\$1** zugeordnet wird, der Wert von **\$3** **\$2** usw. Sind mehr als neun Befehlszeilenargumente vorhanden, wird das zehnte **\$9** zugeordnet und die verbleibenden Argumente werden zunächst nicht zugeordnet (bis ein weiterer Befehl **shift** erfolgt). Sind neun oder weniger Argumente vorhanden, inaktiviert der Befehl **shift** den Positionsparameter mit der höchsten Nummer.

Der Positionsparameter **\$0** wird nie verschoben. Der Befehl **shift n** ist eine verkürzte Schreibweise, mit der die Anzahl **n** der aufeinanderfolgenden Verschiebungen angegeben wird. Der Standardwert des Parameters **n** ist 1.

test Ausdruck | [*Ausdruck*]

Wertet Bedingungsausdrücke aus. Weitere Informationen zu Optionen und Parametern enthält der Abschnitt zum Befehl **test**.

times

Zeigt die aufgelaufenen Benutzer- und Systemzeiten für Prozesse an, die von der Shell ausgeführt wurden.

trap [Befehl] [n] ...

Führt den im Parameter *Befehl* angegebenen Befehl aus, wenn die Shell das (oder die) in der Variablen *n* angegebene(n) Signal(e) empfängt. Die Befehle **trap** werden in der Reihenfolge der Signalnummern ausgeführt. Der Versuch, eine Hardwareprogrammverzweigung auf ein Signal zu setzen, das beim Eingang zur aktuellen Shell ignoriert wurde, ist wirkungslos.

Hinweis: Die Shell durchsucht den Parameter *Befehl* einmal, wenn die Hardwareprogrammverzweigung gesetzt wird und ein weiteres Mal, wenn die Hardwareprogrammverzweigung ausgeführt wird.

Wird kein Befehl angegeben, werden alle im Parameter *n* angegebenen Hardwareprogrammverzweigungen auf ihren aktuellen Wert zurückgesetzt. Wird eine leere Zeichenfolge angegeben, wird dieses Signal von der Shell und von den aufgerufenen Befehlen ignoriert. Ist der Parameter *n* gleich 0, (Null), wird der angegebene Befehl beim Verlassen der Shell ausgeführt. Wird weder ein Befehl noch ein Signal angegeben, zeigt der Befehl **trap** eine Liste mit Befehlen an, die jeder Signalnummer zugeordnet sind.

type [*Name* . . .]

Zeigt für jeden angegebenen Namen an, wie die Shell ihn als Befehlsnamen interpretieren würde.

ulimit [*Option* [*Variable*]]

Setzt Grenzwerte für die Prozeßgröße oder fragt diese ab. Dabei werden folgende Optionen verwendet:

-b [*m*] Setzt den aktuellen Gruppenwechselwert, der durch die Unterroutine **brk** oder **sbrk** definiert wird, auf den Wert der Variable *m*. Dadurch wird die Größe von Datensegmenten auf *m* Seiten begrenzt. Der Wert der Variable *m* wird zur nächsten Seitenbegrenzung gerundet und der neu zugeordnete Gruppenwechselwert wird zurückgegeben. Wird die Option **-b** ohne die Variable *m* angegeben, zeigt der Befehl **ulimit** den aktuellen Gruppenwechselwert des Prozesses an.

-f [*n*] Setzt den Grenzwert der Prozeßdateigröße auf den Wert von *n* Blöcken. Der Wert der Variable *n* legt einen Grenzwert für die Größe der größten Datei fest, die von einem Prozeß, der unter der aktuellen Shell ausgeführt wird, erstellt werden kann. Dateien jeglicher Größe können gelesen werden. Da die Shell der Wert der Variablen *n* abrundet auf die nächst untere Gruppengröße, ist es empfehlenswert, als Wert ein Vielfaches von 4 zu nehmen. Wird für die Variable *n* beispielsweise 1, 2 oder 3 angegeben, resultiert daraus ein Grenzwert für die Dateigröße von 0. Wird kein Wert für die Variable *n* angegeben, zeigt der Befehl **ulimit** den aktuellen Grenzwert (die Standardaktion) an.

Hinweis: Jeder Benutzer kann den Grenzwert für die Dateigröße verkleinern, aber nur ein Benutzer mit Root-Berechtigung kann den Grenzwert erhöhen.

-s [*m*] Setzt den Grenzwert der niedrigsten Stapelspeichergröße auf *m* Seiten. Der Wert der Variable *m* wird auf die nächst niedrigere Seitenbegrenzung abgerundet. Wird die Option **-s** ohne die Variable *m* angegeben, zeigt der Befehl **ulimit** den aktuellen Grenzwert für die Stapelspeichergröße an. Wenn der Stapelspeicher von den hohen auf die unteren Adreßbereiche ausgedehnt wird, wird die niedrigste gültige Stapelspeicheradresse zurückgegeben.

umask [*nnn*]

Setzt die Dateimoduserstellungsmaske des Benutzers auf den Wert der Variablen *nnn*. Wird die Variable *nnn* übergangen, zeigt der Befehl **umask** den aktuellen Wert der Maske an.

unset [*Name* . . .]

Entfernt die entsprechende Variable oder Funktion für jeden im Parameter *Name* angegebenen Namen. Die Shell-Variablen **PATH**, **PS1**, **PS2**, **MAILCHECK** und **IFS** können nicht inaktiviert werden.

wait [*n*]

Wartet auf die Beendigung des Sohnprozesses, dessen Prozeßnummer im Parameter *n* angegeben ist und meldet dessen Beendigungsstatus. Wird der Parameter *n* nicht angegeben, wartet die Shell auf den momentan aktiven Sohnprozeß und gibt den Rückgabewert 0 aus.

Befehlssubstitution in der Bourne-Shell

Die Befehlssubstitution ermöglicht dem Benutzer, die Ausgabe eines beliebigen Befehls als ein Argument eines weiteren Befehls zu erfassen. Wird eine Befehlszeile zwischen `' '` (Gravis) gesetzt, führt die Shell zunächst den Befehl bzw. die Befehle aus und ersetzt dann den gesamten Ausdruck, einschließlich der Gravis, durch die Ausgabe. Diese Funktion wird oft in Zuordnungsanweisungen verwendet. So wird zum Beispiel in der Anweisung:

```
heutiges Datum='date'
```

die Zeichenfolge, die das aktuelle Datum darstellt, der Variablen `heutiges Datum` zugeordnet. Die folgende Zuordnung speichert in der Variablen `Dateien` die Zahl der Dateien im aktuellen Verzeichnis:

```
Dateien='ls | wc -l'
```

Die Befehlssubstitution kann für jeden Befehl ausgeführt werden, der in die Standardausgabe schreibt.

Um die Befehlssubstitution zu verschachteln, jedem der verschachtelten Gravis einen `\` (umgekehrter Schrägstrich) voranzustellen, wie in:

```
logmsg='echo Benutzeranmeldeverzeichnis: `pwd`'
```

Shell-Variablen können indirekt auch Werte gegeben werden, indem der integrierte Befehl **read** verwendet wird. Dieser Befehl nimmt eine Zeile der Standardeingabe (normalerweise die Tastatur) und ordnet auf dieser Zeile den benannten Variablen nachfolgende Wörter zu. Zum Beispiel nimmt der Befehl:

```
read Vorname Name
```

die Eingabezeile im Format:

```
John Public
```

Das Ergebnis sieht aus, als wenn folgende Zeile eingegeben worden wäre:

```
Vorname=John Name=Public
```

Der Sonderbefehl **read** ordnet überschüssige Wörter der letzten Variablen zu.

Variablensubstitution in der Bourne-Shell

In der Bourne-Shell gibt es mehrere Möglichkeiten, Variablen zu erstellen (einen Zeichenfolgewart einem Namen zuzuordnen). Bestimmte Variablen, Positionsparameter und Schlüsselwortparameter werden normalerweise nur in einer Befehlszeile gesetzt. Andere Variablen sind einfach Namen, denen der Benutzer oder die Shell Zeichenfolgewart zuzuordnen kann.

Die folgenden Abschnitte enthalten Informationen über das Erstellen und Ersetzen von Variablen in der Bourne-Shell:

- Benutzerdefinierte Variablen
- Bedingte Substitution (auf Seite B-17)
- Positionsparameter (auf Seite B-18)

Benutzerdefinierte Variablen

Die Shell erkennt alphanumerische Variablen, denen Zeichenfolgewart zugeordnet werden können. Um einem Zeichenfolgewart einen Namen zuzuordnen, folgendes eingeben:

Name=Wert

Ein Name ist eine Folge von Buchstaben, Ziffern und Unterstreichungszeichen, die mit einem Unterstreichungszeichen oder Buchstaben beginnt. Um den einer Variablen zugeordneten Wert zu verwenden, am Anfang des Namens ein \$ (Dollarzeichen) hinzufügen. So ergibt die Variable *\$Name* den Wert, der in der Variablen *Zeichenfolge* angegeben ist. Zu beachten ist, daß in einer Zuordnungsanweisung links und rechts vom = (Gleichheitszeichen) keine Leerzeichen stehen. (Positionsparameter kommen in einer Zuordnungsanweisung nicht vor. Sie können nur, wie im Abschnitt "Positionsparameter" auf Seite B-18 beschrieben, gesetzt werden.) In eine Befehlszeile kann mehr als eine Zuordnung gestellt werden, wobei zu beachten ist, daß die Shell Zuordnungen von rechts nach links ausführt.

Wird die Variable *Zeichenfolge* in Anführungszeichen oder halbe Anführungszeichen eingeschlossen (", '), interpretiert die Shell Leerzeichen, Tabulatorzeichen, Semikolons und Zeilenvorschubzeichen innerhalb der Zeichenfolge nicht als Wortbegrenzer, sondern bettet sie buchstäblich in die Zeichenfolge ein.

Wird die Variable *Zeichenfolge* in Anführungszeichen (") eingeschlossen, erkennt die Shell noch immer Variablennamen in der Zeichenfolge und führt die Variablensubstitution aus, d. h., sie ersetzt Referenzdaten für Positionsparameter und andere Variablennamen, denen ein \$ (Dollarzeichen) voransteht, durch die entsprechenden Werte, falls vorhanden. Die Shell führt auch die Befehlssubstitution innerhalb von Zeichenfolgen aus, die in Anführungszeichen eingeschlossen sind.

Wird die Variable *Zeichenfolge* in halbe Anführungszeichen (') eingeschlossen, ersetzt die Shell keine Variablen oder Befehle innerhalb der Zeichenfolge. Das folgende Beispiel zeigt diesen Unterschied:

```
Benutzer:          stars=*****
                  Stern1="Plus $stars"
                  Stern2='Plus $stars'
                  echo $Stern1

System:           Plus *****

Benutzer:          echo $Stern2

System:           Plus $stars
```

Nach der Variablensubstitution interpretiert die Shell Leerzeichen in Zuordnungen nicht noch einmal. Daher haben die folgenden Zuordnungen mit dem Ergebnis `$first` und `$second` denselben Wert:

```
first='Eine Zeichenfolge mit eingegebenen Leerzeichen'  
second=$first
```

Wird auf eine Variable verwiesen, kann der Variablenname (oder die Ziffer, die einen Positionsparameter kennzeichnet) in `{ }` (geschweifte Klammern) eingeschlossen werden, um den Variablennamen von nachfolgenden Zeichenfolgen abzugrenzen. Dies gilt besonders, wenn es sich bei dem unmittelbar auf den Namen folgenden Zeichen um einen Buchstaben, eine Ziffer oder ein Unterstrichungszeichen handelt und die Variable kein Positionsparameter ist. In diesem Fall sind geschweifte Klammern erforderlich:

```
Benutzer:          a='Dies ist ein'  
                  echo "${a}e Zeichenfolge"  
  
Anzeige:          Dies ist eine Zeichenfolge  
  
Benutzer:          echo "$a Test"  
  
Anzeige:          Dies ist ein Test
```

Der Abschnitt "Bedingte Substitution" auf Seite B-17 enthält weitere Informationen zur Verwendung von geschweiften Klammern bei der Variablensubstitution.

Von der Shell verwendete Variablen

Die Shell verwendet die folgenden Variablen. Während die Shell nur einige von ihnen setzt, kann der Benutzer alle setzen und zurücksetzen:

- | | |
|-------------------|--|
| CDPATH | Gibt den Suchpfad für den Befehl <code>cd</code> (Verzeichnis wechseln) an. |
| HOME | Zeigt den Namen des <i>Anmeldeverzeichnisses</i> an, das nach Abschluß der Anmeldung zum aktuellen Verzeichnis wird. Das Anmeldeprogramm initialisiert diese Variable. Der Befehl <code>cd</code> verwendet den Wert der Variablen \$HOME als Standardwert. Durch die Verwendung dieser Variablen in einer Shell-Prozedur statt eines expliziten Pfadnamens ist es möglich, Prozeduren von unterschiedlichen Verzeichnissen aus ohne Veränderungen auszuführen. |
| IFS | Setzt die Zeichen, bei denen es sich um interne Feldtrennzeichen handelt (d. h. Zeichen, die die Shell bei der Interpretation von Leerzeichen verwendet; siehe Abschnitt "Interpretation von Leerzeichen"). Die Shell setzt beim Start die Variable IFS und schließt damit das Leerzeichen, das Tabulatorzeichen und das Zeilenvorschubzeichen ein. |
| LANG | Bestimmt die länderspezifischen Angaben, die verwendet werden sollen, wenn weder die Variable LC_ALL noch die entsprechende Umgebungsvariable (die mit LC_ beginnt) eine länderspezifische Angabe enthalten. |
| LC_ALL | Bestimmt die länderspezifischen Angaben, mit denen alle länderspezifischen Angaben überschrieben werden sollen, die durch die Einstellungen der Umgebungsvariablen LANG oder jeder anderen Umgebungsvariablen, die mit LC_ beginnt, festgelegt sind. |
| LC_COLLATE | Definiert die Sortierfolge, die verwendet wird, wenn Namen sortiert werden und wenn in Mustern Zeichenbereiche auftreten. Wird diese Variable nicht angegeben, wird die entsprechende Sortierfolge aus der Umgebungsvariablen LANG verwendet. Sind beide Variablen nicht angegeben, wird die ANSI-C-Sortierfolge verwendet. |

- LC_CTYPE** Bestimmt die länderspezifischen Angaben für die Interpretation von Bytefolgen von Textdaten als Zeichen (d. h. Einzelbytezeichen versus Mehrbytezeichen in Argumenten und Eingabedateien). Bestimmt weiterhin, welche Zeichen als Buchstaben (**Alphazeichenklasse**) definiert sind und wie sich Zeichenklassen bei der Verwendung als Platzhalterzeichen verhalten.
- LC_MESSAGES** Bestimmt, in welcher Sprache Nachrichten geschrieben werden.
- LIBPATH** Gibt den Suchpfad für gemeinsam benutzte Bibliotheken an.
- LOGNAME** Gibt den Anmeldenamen des Benutzer an, der als **schreibgeschützt** in der Datei **/etc/profile** markiert ist.
- MAIL** Gibt den Pfadnamen der Datei an, die vom Postsystem verwendet wird, um neue Post zu finden. Ist diese Variable gesetzt, überprüft die Shell in gewissen Abständen die Änderungszeit dieser Datei und zeigt den Wert der Variablen **\$MAILMSG** an, wenn sich die Zeit geändert hat und die Dateilänge größer als Null ist. Die Variable **MAIL** wird in der Datei **.profile** gesetzt. Benutzer des Befehls **mail** ordnen dieser Variablen normalerweise den Wert **/usr/mail/\$LOGNAME** zu.
- MAILCHECK** Gibt die Sekunden an, die die Shell vergehen läßt, bevor erneut geprüft wird, ob Post in den durch die Variablen **MAILPATH** oder **MAIL** angegebenen Dateien angekommen ist. Der Standardwert ist 600 Sekunden (10 Minuten). Wird die Variable **MAILCHECK** auf 0 gesetzt, führt die Shell vor jeder Eingabeaufforderung eine Überprüfung durch.
- MAILMSG** Gibt die Postbenachrichtigungsnachricht an. Wird die Variable **MAILMSG** explizit auf eine leere Zeichenfolge (**MAIL=""**) gesetzt, wird keine Nachricht angezeigt.
- MAILPATH** Gibt eine Liste von Dateinamen an, die durch Doppelpunkte getrennt werden. Ist diese Variable gesetzt, informiert die Shell den Benutzer, wenn in einer der in der Liste angegebenen Dateien Post angekommen ist. Nach jedem Dateinamen kann ein % (Prozentzeichen) und eine Nachricht eingegeben werden, die bei der Ankunft von Post angezeigt wird. Andernfalls verwendet die Shell den Wert der Variablen **MAILMSG** oder standardmäßig die Nachricht **you have mail**.
- Hinweis:** Ist die Variable **MAILPATH** gesetzt, werden die dort angegebenen Dateien überprüft und nicht die in der Variablen **MAIL** gesetzte Datei. Sollen sowohl die Dateien der Variablen **MAILPATH** als auch die der Variablen **MAIL** überprüft werden, die Datei **MAIL** in der Dateiliste **MAILPATH** angeben.
- PATH** Gibt den Suchpfad für Befehle an, wobei es sich um eine geordnete Liste von Verzeichnispfadnamen handelt, die durch Doppelpunkte getrennt sind. Die Shell durchsucht diese Verzeichnisse in der angegebenen Reihenfolge, wenn sie nach Befehlen sucht. Eine leere Zeichenfolge an einer beliebigen Stelle in der Liste stellt das aktuelle Verzeichnis dar.
- Die Variable **PATH** wird normalerweise von der Datei **/etc/profile** gesetzt, und zwar auf **/bin:/usr/bin:/etc::**. Der Benutzer kann diese Variable jedoch entsprechend seinen eigenen Bedürfnissen zurücksetzen. Soll beispielsweise als erstes (und nicht als letztes) das aktuelle Verzeichnis durchsucht werden, folgendes eingeben:
- ```
PATH=:/bin:/usr/bin:/etc
```

In diesem Beispiel wird per Definition eine leere Zeichenfolge vor dem führenden Doppelpunkt angenommen. Verfügt der Benutzer über ein persönliches Befehlsverzeichnis (beispielsweise **\$HOME/bin**), das vor dem Standardbenutzerverzeichnis durchsucht werden soll, muß die Variable **PATH** folgendermaßen gesetzt werden:

```
PATH=$HOME/bin:/bin:/usr/bin:/etc::
```

Soll die Variable **PATH** auf einen Wert gesetzt werden, der vom Standardwert abweicht, geschieht das am besten in der Datei **.profile**. Werden Befehle unter der eingeschränkten Shell ausgeführt, kann die Variable **PATH** nicht zurückgesetzt werden.

- PS1** Gibt die Zeichenfolge an, die als primäre Eingabeaufforderung verwendet werden soll. Eine interaktive Shell zeigt diese Eingabeaufforderungszeichenfolge an, wenn eine Eingabe erwartet wird. Der Standardwert der Variable **PS1** ist ein \$ (Dollarzeichen), gefolgt von einem Leerzeichen.
- PS2** Gibt den Wert der Zeichenfolge für die zweite Eingabeaufforderung an. Erwartet die Shell weitere Eingaben, wenn sie auf ein Zeilenvorschubzeichen trifft, wird als Eingabeaufforderung der Wert der Variablen **PS2** ausgegeben. Der Standardwert der Variable **PS2** ist ein > (Größer-als-Zeichen), gefolgt von einem Leerzeichen.
- SHACCT** Gibt den Namen einer Datei an, deren Eigner der Benutzer ist. Ist diese Variable gesetzt, schreibt die Shell für jede ausgeführte Shell-Prozedur einen Abrechnungssatz in die Datei. Um die gesammelten Daten zu analysieren, können Abrechnungsprogramme wie **acctcom** und **acctcms** verwendet werden.
- SHELL** Gibt den Pfadnamen der Shell an, die in der Umgebung existiert. Diese Variable sollte von der Datei **\$HOME/.profile** jeder eingeschränkten Anmeldung gesetzt und exportiert werden.
- TIMEOUT** Gibt die Zahl der Sekunden an, die die Shell vor dem Verlassen inaktiv bleibt. Ist diese Variable auf einen Wert größer als Null gesetzt, wird die Shell verlassen, wenn innerhalb der vorgeschriebenen Zahl von Sekunden nach Ausgabe der Eingabeaufforderung **PS1** kein Befehl eingegeben wurde. (Zu beachten ist, daß die Shell mit einem maximalen Grenzwert kompiliert werden kann, der für den hier beschriebenen Wert nicht überschritten werden darf.)

## Vordefinierte Sondervariablen

Einige Variablen haben eine Sonderbedeutung. Die folgenden Variablen werden nur von der Shell gesetzt:

- \$#** Gibt die Zahl der Positionsparameter an, die an die Shell übergeben wurden, wobei der Name der Shell-Prozedur selbst nicht gezählt wird. Die Variable **\$#** liefert demzufolge die Zahl des Positionsparameters mit der höchsten Nummer, der gesetzt wurde. Diese Variable dient hauptsächlich zur Überprüfung, ob die erforderliche Zahl von Argumenten vorhanden ist.
- \$?** Gibt den Rückkehrcode des letzten ausgeführten Befehls an. Sein Wert ist eine dezimale Zeichenfolge. Die meisten Befehle geben den Code 0 zurück, um den erfolgreichen Abschluß anzuzeigen. Die Shell selbst gibt als Rückkehrcode den aktuellen Wert der Variable **\$?** zurück.

**\$\$** Gibt die Prozeßnummer des aktuellen Prozesses an. Da Prozeßnummern für alle bestehenden Prozesse eindeutig sind, wird diese bis zu fünf Ziffern lange Zeichenfolge oft verwendet, um eindeutige Namen für temporäre Dateien zu generieren.

Das folgenden Beispiel zeigt die empfohlene Vorgehensweise zum Erstellen von temporären Dateien in einem eigens dafür angelegten Verzeichnis:

```
temp=$HOME/temp/$$
ls >$temp
.
.
rm $temp
```

**#!** Gibt die Prozeßnummer des letzten Prozesses an, der unter Verwendung des Abschlußzeichens & (Et-Zeichen) im Hintergrund ausgeführt wurde. Auch hierbei handelt es sich um eine bis zu fünf Ziffern lange Zeichenfolge.

**\$-** Gibt eine Zeichenfolge an, die aus den Namen der momentan in der Shell gesetzten Ausführungsoptionen besteht.

## Interpretation von Leerzeichen

Nach der Variablen- und Befehlssubstitution durchsucht die Shell das Ergebnis nach internen Feldtrennzeichen (d. h. denen, die in der Shell-Variablen **IFS** definiert sind). Dort, wo die Shell eines dieser Zeichen findet, wird die Zeile in bestimmte Argumente eingeteilt. Dann werden explizite Nullargumente ("") beibehalten, wohingegen implizite Nullargumente (die aus Parametern ohne Werte resultieren) gelöscht werden.

## Bedingte Substitution

Normalerweise ersetzt die Shell den Ausdruck **\$Variable** durch den Zeichenfolgewert, der der Variablen *Variable*, falls vorhanden, zugeordnet ist. Dennoch gibt es eine besondere Schreibweise, die die *bedingte Substitution* ermöglicht, abhängig davon, ob die Variable gesetzt oder/und nicht leer ist. Per Definition ist eine Variable gesetzt, wenn ihr irgendwann ein Wert zugeordnet wurde. Der Wert einer Variablen kann auch eine leere Zeichenfolge sein, die auf eine der folgenden Arten einer Variablen zugeordnet werden kann:

A=

bcd=""

Efg=''

Ordnet die leere Zeichenfolge jedem der entsprechenden Variablenamen zu.

set '' ''

Setzt den ersten und zweiten Positionsparameter auf die leere Zeichenfolge und inaktiviert alle anderen Positionsparameter.

Es folgt eine Liste von verfügbaren Ausdrücken, die zur Ausführung der bedingten Substitution verwendet werden können:

**`\${Variable}- Zeichenfolge`**

Ist die Variable gesetzt, wird der Wert *Variable* für diesen Ausdruck eingesetzt. Andernfalls wird dieser Ausdruck gegen den Wert *Zeichenfolge* ausgetauscht.

**`\${Variable:-Zeichenfolge`**

Ist die Variable gesetzt und nicht leer, wird der Wert *Variable* für diesen Ausdruck eingesetzt. Andernfalls wird dieser Ausdruck gegen den Wert *Zeichenfolge* ausgetauscht.

### **`${Variable=Zeichenfolge}`**

Ist die Variable gesetzt, wird der Wert *Variable* für diesen Ausdruck eingesetzt. Andernfalls wird der Wert *Variable* auf den Wert *Zeichenfolge* gesetzt und dann für den ursprünglichen Wert *Variable* eingesetzt. Positionsparametern können auf diese Weise keine Werte zugeordnet werden.

### **`${Variable:=Zeichenfolge}`**

Ist die Variable gesetzt und nicht leer, wird der Wert *Variable* für diesen Ausdruck eingesetzt. Andernfalls wird der Wert *Variable* auf den Wert *Zeichenfolge* gesetzt und dann für den ursprünglichen Wert *Variable* eingesetzt. Positionsparametern können auf diese Weise keine Werte zugeordnet werden.

### **`${Variable?Zeichenfolge}`**

Ist die Variable gesetzt, wird der Wert *Variable* für diesen Ausdruck eingesetzt. Andernfalls wird eine Nachricht im folgenden Format angezeigt:

*Variable: Zeichenfolge*

und die aktuelle Shell wird verlassen (es sei denn, es handelt sich um die Anmelde-Shell). Wird kein Wert für die Variable *Zeichenfolge* angegeben, zeigt die Shell folgende Nachricht an:

*Variable: parameter null or not set*

### **`${Variable:?Zeichenfolge}`**

Ist die Variable gesetzt und nicht leer, wird der Wert *Variable* für diesen Ausdruck eingesetzt. Andernfalls wird eine Nachricht im folgenden Format angezeigt:

*Variable: Zeichenfolge*

und die aktuelle Shell wird verlassen (es sei denn, es handelt sich um die Anmelde-Shell). Wird kein Wert für die Variable *Zeichenfolge* angegeben, zeigt die Shell folgende Nachricht an:

*Variable: parameter null or not set*

### **`${Variable+Zeichenfolge}`**

Ist die Variable gesetzt, wird der Wert *Zeichenfolge* für diesen Ausdruck eingesetzt. Andernfalls wird die leere Zeichenfolge eingesetzt.

### **`${Variable:+Zeichenfolge}`**

Ist die Variable gesetzt und nicht leer, wird der Wert *Zeichenfolge* für diesen Ausdruck eingesetzt. Andernfalls wird die leere Zeichenfolge eingesetzt.

Bei der bedingten Substitution wertet die Shell die Variable *Zeichenfolge* erst aus, wenn sie diese Variable als Ersatzzeichenfolge verwendet. So führt die Shell im folgenden Beispiel den Befehl `pwd` nur aus, wenn `d` nicht gesetzt oder leer ist:

```
echo ${d:-'pwd'}
```

## **Positionsparameter**

Wird eine Shell-Prozedur ausgeführt, erstellt die Shell automatisch Positionsparameter, die auf jedes Wort in der Befehlszeile aufgrund seiner dortigen Position verweisen. Das Wort auf Position 0 (der Prozedurname) wird `$0` genannt, das nächste Wort (der erste Parameter) wird `$1` genannt usw. bis `$9`. Um auf Befehlszeilenparameter zu verweisen, deren Nummer höher als 9 ist, den integrierten Befehl `shift` verwenden.

Diesen Positionsparametern können auch explizit Werte zugeordnet werden, indem der integrierte Befehl **set** verwendet wird.

**Hinweis:** Wird ein Argument für eine Position nicht angegeben, wird dessen Positionsparameter auf Null gesetzt. Positionsparameter sind global und können an verschachtelte Shell-Prozeduren übergeben werden.

---

## Dateinamenssubstitution in der Bourne-Shell

Befehlsparameter sind oft Dateinamen. Eine Liste mit Dateinamen als Parameter in der Befehlszeile kann automatisch erzeugt werden. Dazu ein Zeichen angeben, das die Shell als Platzhalterzeichen erkennt. Enthält ein Befehl ein solches Zeichen, ersetzt die Shell es durch Dateinamen in einem Verzeichnis.

**Hinweis:** In der Bourne-Shell wird die auf die Äquivalenzklassifikation von Zeichen basierende Dateinamenserweiterung nicht unterstützt.

Die meisten Zeichen in solchen Mustern entsprechen sich selbst, aber es können auch einige besondere Platzhalterzeichen in Mustern verwendet werden. Diese Sonderzeichen sind:

- \*            Entspricht jeder Zeichenfolge, einschließlich der leeren Zeichenfolge.
- ?            Entspricht jedem Zeichen.
- [... ]        Entspricht jedem in eckige Klammern eingeschlossenen Zeichen.
- [!... ]       Entspricht jedem Zeichen *mit Ausnahme* der Zeichen, die in der eckigen Klammer auf das Ausrufezeichen folgen.

Innerhalb von eckigen Klammern gibt ein durch einen – (Bindestrich) getrenntes Zeichenpaar alle Zeichen an, die aufgrund ihrer Position innerhalb der aktuellen Sortierfolge zwischen diesen beiden Zeichen liegen. Über die Umgebungsvariablen **LANG** und **LC\_COLLATE** wird die Sortierfolge gesteuert.

Die aktuelle Sortierfolge gruppiert möglicherweise Zeichen zu Äquivalenzklassen, um die Endpunkte des Zeichenbereichs zu definieren. Wird beispielsweise durch die Sortierfolge die lexikalische Reihenfolge **AaBbCc** ... definiert und werden Groß- und Kleinbuchstaben in Äquivalenzklassen gruppiert, haben alle folgenden Klammerausdrücke dieselbe Wirkung: **[a-c]**, **[A-C]**, **[a-C]** und **[A-c]**.

**Hinweis:** Durch eine Sortierfolge in länderspezifischen Angaben für Mehrbytesprachen können keine Äquivalenzklassen zur Verwendung in Bereichsausdrücken definiert werden. Um unvorhersehbare Ergebnisse beim Abgleich eines Bereichsausdrucks mit einer Klasse von Zeichen zu vermeiden, einen *Zeichenklassenausdruck* statt eines Standardbereichsausdrucks verwenden.

Die Verwendung von Platzhalterzeichen unterliegt einigen Beschränkungen. Ist das erste Zeichen in einem Dateinamen ein **.** (Punkt), kann es nur mit einem Muster übereinstimmen, das auch mit einem Punkt beginnt. Beispielsweise stimmt **\*** (Stern) mit den Dateinamen **meindat** und **deindat** überein, aber nicht mit den Dateinamen **.meindat** und **.deindat**. Um diese Dateien abzugleichen, ein ähnliches Muster wie das folgende verwenden:

```
.*Datei
```

Stimmt das Muster mit keinem Dateinamen überein, wird das Muster selbst als Ergebnis des versuchten Abgleichs zurückgegeben.

Datei- und Verzeichnisnamen sollten die Zeichen \* (Stern), ? (Fragezeichen), [ (linke eckige Klammer) oder ] (rechte eckige Klammer) nicht enthalten, da dadurch eine endlose Rekursion (d. h. Endlosschleifen) während der Mustersuche verursacht werden kann.

## Zeichenklassen

Um Dateinamen in einem Bereich abzugleichen, können auch Zeichenklassen verwendet werden:

[ :Zeichenklasse: ]

Mit diesem Format gleicht das System jedes einzelne Zeichen, das zu der angegebenen Klasse gehört, ab. Die definierten Klassen entsprechen den Unterroutinen **ctype**. Folgende Zeichenklassen sind vorhanden:

### Zeichenklasse Definition

|               |                                                                                                                                      |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------|
| <b>alnum</b>  | alphanumerische Zeichen                                                                                                              |
| <b>alpha</b>  | Groß- und Kleinbuchstaben                                                                                                            |
| <b>digit</b>  | Ziffern                                                                                                                              |
| <b>lower</b>  | Kleinbuchstaben                                                                                                                      |
| <b>print</b>  | druckbare Zeichen                                                                                                                    |
| <b>punct</b>  | Interpunktionszeichen                                                                                                                |
| <b>space</b>  | Leerzeichen, Tabulatorzeichen, Zeilenschaltungszeichen, Zeilenvorschubzeichen, Vertikaltabulatorzeichen oder Formularvorschubzeichen |
| <b>upper</b>  | Großbuchstaben                                                                                                                       |
| <b>xdigit</b> | Hexadezimalzahlen                                                                                                                    |



---

## Ein- und Ausgabe in der Bourne-Shell umleiten

Im allgemeinen ist es bei den meisten Befehlen unerheblich, ob die Ein- oder Ausgabe über die Tastatur, den Bildschirm oder eine Datei erfolgt. Demzufolge kann ein Befehl einfach entweder an der Tastatur oder in einer Pipeline verwendet werden.

Dieses Kapitel enthält Informationen zu folgenden Bereichen der Ein- und Ausgabeumadressierung:

- Standardein- und -ausgabe
- Diagnose- und sonstige Ausgabe
- Befehlszeileneingabedokument ("bin da"-Dokument)
- Ein- und Ausgabeumadressierung mit Dateideskriptoren
- Zusammenfassung der Umadressierungsoptionen

### Standardein- und -ausgabe

Beginnt ein Befehl mit der Ausführung, sollten normalerweise drei Dateien bereits geöffnet sein: Standardeingabe, Standardausgabe und die Diagnoseausgabe (auch Fehlerausgabe oder Standarddatei für Fehlerausgabe genannt). Jeder dieser Dateien ist eine Zahl, der sogenannte Dateideskriptor, zugeordnet:

**Dateideskriptor 0** Standardeingabe.  
**Dateideskriptor 1** Standardausgabe.  
**Dateideskriptor 2** Diagnoseausgabe (Fehlerausgabe)

Ein Sohnprozeß übernimmt diese Dateien normalerweise von seinem Vaterprozeß. Alle drei Dateien sind standardmäßig der Datenstation zugeordnet (0 der Tastatur, 1 und 2 dem Bildschirm). In der Shell können sie jedoch umgeleitet werden, bevor die Steuerung an einen Befehl übergeben wird.

Die folgenden Shell-Argumente bewirken eine Ein- und Ausgabeumadressierung:

**<Datei** Öffnet die angegebene Datei als Standardeingabe.  
**>Datei** Öffnet die angegebene Datei als Standardausgabe. Dadurch wird der vorherige Inhalt der Datei, falls vorhanden, gelöscht.  
**>>Datei** Leitet die Standardausgabe zum Ende der angegebenen Datei. Durch diese Option können Daten der Datei hinzugefügt werden, ohne den bestehenden Inhalt zu löschen.

**Hinweis:** Für das Argument **>Datei** oder das Argument **>>Datei** erstellt die Shell die angegebene Datei, wenn sie noch nicht existiert.

Diese Umadressierungsargumente betreffen nur die Variablen- und Befehlssubstitution. Nach der Substitution wird weder eine Interpretation von Leerzeichen ausgeführt, noch hat die Verwendung von Platzhalterzeichen bei Dateinamen Gültigkeit. Daher erzeugt die Eingabe:

```
echo 'Dies ist ein Test' > *.temp
```

eine einzeilige Datei mit dem Namen `*.temp` und die Eingabe:

```
cat < ?
```

eine Fehlermeldung, es sei denn, es gibt eine Datei mit dem Namen `?`.

## Diagnose- und sonstige Ausgabe

Die Diagnoseausgabe von Befehlen wird normalerweise an die dem Dateideskriptor 2 zugeordnete Datei geleitet. Diese Fehlerausgabe kann in eine andere Datei umgeleitet werden, indem entweder dem Ausgabeumadressierungssymbol > oder >> eine 2 (die Nummer des Dateideskriptors) unmittelbar vorangestellt wird. Beispielsweise fügt die folgende Zeile Fehlermeldungen des Befehls **make** an die Datei Fehler an:

```
make testdat 2>> Fehler
```

**Hinweis:** Zwischen dem Dateideskriptor und dem Umadressierungssymbol dürfen keine Leerzeichen stehen. Andernfalls interpretiert die Shell die Zahl als separates Argument zum Befehl.

Mit dieser Methode kann auch die Ausgabe, die einem der ersten 10 Dateideskriptoren (mit den Nummern 0 bis 9) zugeordnet ist, umgeleitet werden. Schreibt beispielsweise ein Befehl in den Dateideskriptor 9, kann diese Ausgabe durch die folgende Eingabe in einer Datei mit dem Namen `sichdat` gespeichert werden:

```
Befehl 9> sichdat
```

Schreibt ein Befehl in mehr als eine Ausgabe, können diese unabhängig voneinander umgeleitet werden. Angenommen, ein Befehl leitet seine Standardausgabe in den Dateideskriptor 1, seine Fehlerausgabe in den Dateideskriptor 2 und erstellt eine Datendatei in Dateideskriptor 9. Dann kann mit der folgenden Befehlszeile jede dieser Ausgaben in eine andere Datei umgeleitet werden:

```
Befehl > Standard 2> Fehler 9> Daten
```

## Befehlszeileneingabedokument ("bin da"-Dokument)

Wird eine Befehlszeile im Format:

```
Befehl << eofstring
```

einggegeben, wobei *eofstring* eine beliebige Zeichenfolge ohne Platzhalterzeichen ist, übernimmt die Shell die nachfolgenden Zeilen als Standardeingabe von *Befehl*, bis sie eine Zeile liest, die nur aus *eofstring* besteht (möglicherweise gehen ein oder mehrere Tabulatorzeichen voran). Die Zeilen zwischen dem ersten und zweiten *eofstring* werden oft als *Befehlszeileneingabe*- oder "bin da"-Dokument bezeichnet. Folgt unmittelbar auf die Umadressierungszeichen << ein – (Minuszeichen), entfernt die Shell führende Tabulatorzeichen aus jeder Zeile des Eingabedokuments, bevor die Zeile an den Befehl übergeben wird.

Die Shell erstellt eine temporäre Datei, die das Eingabedokument enthält, und führt dort die Variablen- und Befehlssubstitution durch, bevor die Datei an den Befehl übergeben wird. Sie führt einen Musterabgleich für Dateinamen aus, die Teil von Befehlszeilen bei der Befehlssubstitution sind. Um Substitutionen zu verhindern, ein beliebiges Zeichen von *eofstring* in Anführungszeichen setzen oder einen umgekehrten Schrägstrich voranstellen:

```
Befehl << \eofstring
```

Das "bin da"-Dokument ist besonders nützlich bei geringen Mengen von Eingabedaten, die eher in die Shell-Prozedur gestellt werden sollten statt in eine separate Datei (wie z. B. Editorprozeduren). Beispielsweise kann eingegeben werden:

```
cat <<- xyz
 Diese Nachricht erscheint in der Anzeige
 ohne führende Tabulatorzeichen.
xyz
```

Diese Funktion wird oft in Shell-Prozeduren verwendet.

**Hinweis:** Befehlszeileneingabedokumente dürfen nicht innerhalb von ' ' (Gravis) stehen, da diese Zeichen die Befehlssubstitution anzeigen.

## Ein- und Ausgabeumadressierung mit Dateideskriptoren

Wie bereits im Abschnitt "Diagnose- und sonstige Ausgabe" beschrieben, kann ein Befehl die Ausgabe zu einer Datei leiten, die einem anderen Dateideskriptor zugeordnet ist als dem Dateideskriptor 1 oder 2. Die Shell bietet außerdem die Möglichkeit, eine Ausgabedatei zu erstellen, die einem bestimmten Dateideskriptor zugeordnet ist. Durch die Eingabe von:

```
Datei1>&Datei2
```

wobei *Datei1* und *Datei2* gültige Dateideskriptoren sind, kann die Ausgabe, die normalerweise dem Dateideskriptor *Datei1* zugeordnet ist, in die Datei geleitet werden, die *Datei2* zugeordnet ist. Der Standardwert für *Datei1* und *Datei2* ist 1 (Standardausgabe). Ist zur Ausführungszeit *Datei2* keine Datei zugeordnet, wird die Umadressierung ungültig. Diese Möglichkeit wird meistens dazu verwendet, um die Standardfehlerausgabe in dieselbe Datei zu leiten wie die Standardausgabe. Dazu folgendes eingeben:

```
Befehl 2 >&1
```

Sollen beide Ausgaben, Standardausgabe und Standardfehlerausgabe, in dieselbe Datei umgeleitet werden, folgendes eingeben:

```
Befehl > Datei2>&1
```

Hierbei ist die Reihenfolge wichtig. Als erstes ordnet die Shell Dateideskriptor 1 der *Datei* zu. Dann ordnet sie Dateideskriptor 2 der *Datei* zu, die momentan dem Dateideskriptor 1 zugeordnet ist. Wird die Reihenfolge der Umadressierungen umgekehrt, erfolgt die Standardfehlerausgabe am Bildschirm und die Standardausgabe in einer Datei, da zum Zeitpunkt der Umadressierung der Fehlerausgabe der Dateideskriptor 1 noch immer dem Bildschirm zugeordnet war.

Auf diese Weise kann auch die Standardeingabe umgeleitet werden. Zum Beispiel können mit der Eingabe:

```
Datei1<&Datei2
```

beide Dateideskriptoren derselben Eingabedatei zugeordnet werden. Für Befehle, die nacheinander ausgeführt werden, ist der Standardwert von *Datei1* und *Datei2* 0 (Standardeingabe). Für Befehle, die asynchron ausgeführt werden (d. h. Befehle, die durch ein & (Et-Zeichen) beendet werden), ist der Standardwert von *Datei1* und *Datei2* die Datei */dev/null*. Diese Eingabeumadressierung ist bei Befehlen nützlich, die zwei oder mehr Eingabequellen verwenden.

## Zusammenfassung der Umadressierungsoptionen

Folgende Umadressierungsoptionen können an beliebiger Stelle in einfachen Befehlen auftreten. Sie können auch einem Befehl voranstehen oder folgen, werden aber nicht an den Befehl übergeben:

- |                 |                                                                                                                                                                          |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| < <i>Datei</i>  | Verwendet die angegebene Datei als Standardeingabe.                                                                                                                      |
| > <i>Datei</i>  | Verwendet die angegebene Datei als Standardausgabe. Falls die Datei noch nicht existiert, wird sie erstellt; andernfalls wird sie auf die Feldlänge Null abgeschnitten.  |
| >> <i>Datei</i> | Verwendet die angegebene Datei als Standardausgabe. Falls die Datei noch nicht existiert, wird sie erstellt; andernfalls wird die Ausgabe am Ende der Datei hinzugefügt. |

**<<[-]eofstr** Liest von der Variablen *eofstr* an alle Zeilen als Standardeingabe bis zu einer Zeile, die nur aus *eofstr* besteht oder bis zu einem Dateiendezeichen. Ist ein beliebiges Zeichen innerhalb der Variablen *eofstr* in Quotierungszeichen gesetzt, werden von der Shell keine Zeichen in den Eingabezeilen erweitert oder interpretiert. Andernfalls führt die Shell die Variablen- und Befehlssubstitution aus und ignoriert quotierte Zeilenvorschubzeichen (**neue Zeile**). Als Quotierungszeichen innerhalb der Variablen *eofstr* oder innerhalb von Eingabezeilen einen \ (umgekehrter Schrägstrich) verwenden.

Wird der Umadressierungsoption << ein - (Minuszeichen) hinzugefügt, werden alle führenden Tabulatorzeichen aus der Variablen *eofstr* und aus den Eingabezeilen entfernt.

**<&Ziffer** Ordnet die Standardeingabe dem in der Variablen *Ziffer* angegebenen Dateideskriptor zu.

**>&Ziffer** Ordnet die Standardausgabe dem in der Variablen *Ziffer* angegebenen Dateideskriptor zu.

**<&-** Schließt die Standardeingabe.

**>&-** Schließt die Standardausgabe.

**Hinweis:** In der eingeschränkten Shell ist eine Ausgabeumadressierung nicht möglich.

## Empfohlene Dokumentation

### Zugehörige Informationen

Die Abschnitte zu den Befehlen **bsh** oder **Rsh**, **chmod**, **csch**, **echo**, **ed** oder **red**, **ksh**, **login**, **make**, **pwd**, **rsh**, **sh**, **test**, **tsh**, **vi** oder **vedit**.

Die Unterroutinen **setuid** und **setgid**.

Die Umgebungsdatei **environment** und die Datei **/etc/passwd**.

Die Gerätedatei **null**.

---

## Anhang C. C-Shell

Bei der C-Shell handelt es sich um einen interaktiven Befehls-Interpreter und eine befehlsorientierte Programmiersprache. Sie verwendet eine ähnliche Syntax wie die Programmiersprache C. Mit dem Befehl **cs** wird die C-Shell aufgerufen.

---

### Verwendung der C-Shell

Beim Aufrufen des Befehls **cs** wird im Benutzerverzeichnis nach der Datei **.cshrc** gesucht. Falls vorhanden, enthält die Datei **.cshrc** angepaßte Benutzerdaten zur Ausführung der C-Shell. Wird die C-Shell als Anmelde-Shell ausgeführt, werden sowohl aus der Datei **.cshrc** als auch aus der Datei **.login** Befehle ausgeführt.

### Beschränkungen der C-Shell

Die C-Shell unterliegt folgenden Beschränkungen:

- Wörter dürfen maximal aus 1024 Zeichen bestehen.
- Argumentenlisten dürfen aus maximal 5120 Zeichen bestehen.
- Die Anzahl von Argumenten für einen Befehl, der eine Dateinamenerweiterung beinhaltet, ist auf ein Sechstel der Anzahl an zulässigen Zeichen in einer Argumentenliste begrenzt.
- Bei Befehlssubstitutionen können nicht mehr Zeichen ersetzt werden, als in einer Argumentenliste zulässig sind.
- Damit Schleifen festgestellt werden können, schränkt die Shell die Anzahl an Alias-Namenssubstitutionen in einer einzigen Zeile auf 20 ein.
- Der Befehl **cs** unterstützt die Dateinamenerweiterung, die auf der Äquivalenzklassifikation von Zeichen basiert, nicht.

### Signalverarbeitung

Normalerweise ignoriert die C-Shell Signale. Jobs, die abgehängt ausgeführt werden, reagieren nicht auf Signale, die über die Tastatur generiert werden (**INTERRUPT**, **QUIT** und **HANGUP**). Andere Signale haben die Werte, die die Shell von der Vater-Shell übernommen hat. Die Art und Weise, wie die Shell Unterbrechungs- und Beendigungssignale in Shell-Prozeduren bearbeitet, kann mit **onintr** gesteuert werden. Anmelde-Shells nehmen Beendigungssignale an oder ignorieren sie. Alle anderen Shell-Arten übergeben Beendigungssignale an die Sohnprozesse. Wenn eine Anmelde-Shell die Datei **.logout** liest, sind in keinem Fall Unterbrechungen erlaubt.

---

## Verwendung von C-Shell-Befehlen

In den folgenden Abschnitten wird beschrieben, wie die C-Shell Befehle interpretiert und ausführt:

- Einführung in Shell-Befehle
- Ausführung nicht integrierter Befehle

### Einführung in Shell-Befehle

Ein einfacher Befehl besteht aus einer Folge von Wörtern, die durch Leerzeichen oder Tabulatorzeichen getrennt sind.

Ein Wort ist eine Folge von Zeichen oder/und Numeralen, die keine Leerzeichen ohne Anführungszeichen enthält. Zusätzlich bilden auch die folgenden Zeichen und doppelten Zeichen Einzelwörter, wenn sie als Befehlstrennzeichen oder Abschlußzeichen verwendet werden:

```
& | ;
&& || << >>
< > ()
```

Diese Sonderzeichen können auch Teil von anderen Wörtern sein. Wenn ihnen jedoch ein \ (umgekehrter Schrägstrich) vorangeht, interpretiert die Shell sie nicht als Sonderzeichen. Zeichenfolgen, die in ' ' oder " " (gleiche Paare von Anführungszeichen) oder Gravis eingeschlossen sind, können auch Teile von Wörtern bilden. Leerzeichen, Tabulatorzeichen und Sonderzeichen bilden keine eigenen Wörter, wenn sie in diese Markierungszeichen gesetzt sind. Zusätzlich kann auch ein Zeilenvorschubzeichen in diese Markierungszeichen eingeschlossen werden, wenn ihm ein \ (umgekehrter Schrägstrich) vorangestellt wird.

Das erste Wort in der Folge eines einfachen Befehls (mit der Nummer 0) gibt normalerweise den Namen des Befehls an. Alle folgenden Wörter werden, mit wenigen Ausnahmen, an diesen Befehl übergeben. Gibt der Befehl eine ausführbare Datei an, bei der es sich um ein kompiliertes Programm handelt, führt die Shell dieses Programm sofort aus. Ist die Datei zwar als ausführbar markiert, aber kein kompiliertes Programm, behandelt die Shell sie wie eine Shell-Prozedur. In diesem Fall erzeugt die Shell eine Subshell, um die Datei zu lesen und die darin enthaltenen Befehle auszuführen.

### Ausführung nicht integrierter Befehle

Stellt die C-Shell fest, daß es sich bei einem Befehl nicht um einen integrierten Shell-Befehl handelt, versucht sie, den Befehl mit dem Befehl **execve** auszuführen. Jedes Wort in der Shell-Variablen **path** benennt ein Verzeichnis, von dem aus die Shell versucht, den Befehl auszuführen. Wird weder die Option **-c** noch die Option **-t** angegeben, setzt die Shell die Namen in diesen Verzeichnissen in eine interne Hash-Tabelle. Die Shell startet den Befehl **exec** nur in Verzeichnissen, in denen der Befehl wahrscheinlich steht. Wird dieser Mechanismus mit dem Befehl **unhash** ausgeschaltet oder werden die Optionen **-c** oder **-t** angegeben, verknüpft die Shell sie mit dem gegebenen Befehlsnamen, um einen Pfadnamen einer Datei zu bilden. Diesen Schritt führt die Shell auch für jede Verzeichniskomponente der Variablen **path** aus, die nicht mit einem / (Schrägstrich) beginnt. Danach versucht die Shell, den Befehl auszuführen.

Klammerbefehle werden immer in einer Subshell ausgeführt. Zum Beispiel zeigt der Befehl:

```
(cd ; pwd) ; pwd
```

das Benutzerverzeichnis an, ohne das aktuelle Verzeichnis zu verlassen. Dagegen wechselt der Befehl:

cd ; pwd

vom aktuellen Verzeichnis zum Benutzerverzeichnis. Klammerbefehle werden meistens verwendet, um Auswirkungen des Befehls **chdir** auf die aktuelle Shell zu verhindern.

Verfügt die Datei über die Ausführungsberechtigung, handelt es sich aber nicht um eine ausführbare Binärdatei, interpretiert die Shell sie als Datei mit Shell-Befehlen und startet zum Lesen eine neue Shell.

Sind Alias-Namen für die Shell vorhanden, werden die Wörter der Alias-Namen den Argumentenlisten vorangestellt, um den Shell-Befehl zu bilden. Das erste Wort in dem Alias-Namen sollte der vollständige Pfadname der Shell sein.

**Hinweis:** Hierbei handelt es sich um einen besonderen, spät auftretenden Fall von Alias-Namenssubstitution, bei dem der Argumentenliste nur Wörter vorangestellt werden dürfen, die nicht verändert wurden.

---

## Integrierte Befehle in der C-Shell

Integrierte Befehle werden innerhalb der Shell ausgeführt. Tritt ein integrierter Befehl als Komponente einer Pipeline an einer anderen Stelle als an der letzten auf, wird der Befehl in einer Subshell ausgeführt.

**Hinweis:** Wird an der C-Shell-Eingabeaufforderung ein Befehl eingegeben, sucht das System zunächst nach einem integrierten Befehl. Existiert kein integrierter Befehl, sucht das System in den in der Shell-Variablen **path** angegebenen Verzeichnissen nach einem Befehl auf Systemebene. Einige integrierte C-Shell-Befehle haben denselben Namen wie Betriebssystembefehle. Dennoch arbeiten diese Befehle nicht unbedingt auf dieselbe Weise. In der jeweiligen Befehlsbeschreibung überprüfen, wie der Befehl arbeitet.

Wird eine Shell-Prozedur von der Shell aus ausgeführt und sind die ersten Zeichen der Shell-Prozedur `#!/ShellPfadname`, führt die C-Shell die Shell aus, die im Kommentar zur Verarbeitung der Prozedur angegeben ist. Andernfalls führt sie die Standard-Shell aus (d. h. die Shell, die mit `/usr/bin/sh` verbunden ist). Erfolgt die Ausführung von der Standard-Shell, werden integrierte C-Shell-Befehle nicht erkannt. Um C-Shell-Befehl vom System ausführen zu lassen, muß die erste Zeile der Prozedur `#!/usr/bin/csh` lauten.

## Liste der integrierten C-Shell-Befehle

Die folgenden Befehle sind in der C-Shell integriert:

**alias**

**alias Name**

**alias [Name [Wortliste ]]**

Zeigt, wenn keine Parameter angegeben werden, alle Alias-Namen an. Anderfalls wird der Alias-Name für den angegebenen Namen angezeigt. Wird in der Variablen *Wortliste* eine Wortliste angegeben, ordnet der Befehl sie als Alias-Namen der Variablen *Name* zu. Der angegebene Name darf nicht **alias** oder **unalias** sein.

**alloc**

Zeigt, wieviel dynamischer Speicherplatz angefordert ist, aufgeteilt in belegten und freien Speicherplatz. Wird ein Argument angegeben, zeigt der Befehl **alloc** die Anzahl der freien und belegten Blöcke in jeder Größenkategorie an. Die Kategorien beginnen bei der Größe 0 und verdoppeln sich bei jedem Schritt.

|                                                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-----------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>bg</b><br><b>bg %Job ...</b>                           | Stellt den aktuellen Job oder die angegebenen Jobs in den Hintergrund und führt sie weiter aus, wenn sie gestoppt werden.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>break</b>                                              | Nimmt die Ausführung nach dem Ende der nächsten Programmschleife <b>foreach</b> oder <b>while</b> wieder auf. Der Befehl <b>break</b> führt die in der aktuellen Zeile verbleibenden Befehle aus. Unterbrechungen auf mehreren Ebenen sind demnach möglich, wenn alle Befehle in eine Zeile geschrieben werden.                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>breaksw</b>                                            | Unterbricht bei einem Befehl <b>switch</b> und nimmt die Ausführung nach dem Befehl <b>endsw</b> wieder auf.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>case Kennsatz:</b>                                     | Definiert einen Kennsatz in einem Befehl <b>switch</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>cd</b><br><b>cd Name</b><br><b>chdir [Name]</b>        | Wechselt vom aktuellen Verzeichnis zu dem in der Variablen <i>Name</i> angegebenen. Wird kein Name angegeben, wechselt der Befehl zum Benutzerverzeichnis. Ist der Wert der Variablen kein Unterverzeichnis des aktuellen Verzeichnisses und beginnt er nicht mit einem / (Schrägstrich), ./ (Punkt, Schrägstrich) oder ../ (zwei Punkte, Schrägstrich), überprüft die Shell jede Komponente der Shell-Variablen <i>cdpath</i> , ob sie ein Unterverzeichnis enthält, das mit der Variablen <i>Name</i> übereinstimmt. Wird keine Übereinstimmung gefunden, ist die Variable <i>Name</i> aber eine Shell-Variable mit einem Wert, der mit einem / (Schrägstrich) beginnt, probiert die Shell aus, ob es sich dabei um ein Verzeichnis handelt. |
| <b>continue</b>                                           | Setzt die Ausführung der nächsten Programmschleife <b>while</b> oder <b>foreach</b> fort. Die restlichen Befehle in der aktuellen Zeile werden ausgeführt.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>default:</b>                                           | Kennzeichnet die Standardschreibweise in einer Anweisung <b>switch</b> . Der Standardwert sollte nach allen Kennsätzen <b>case</b> folgen.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>dirs</b>                                               | Zeigt den Verzeichnisstapel an. Der Anfang des Stapels ist links, das erste Verzeichnis im Stapel ist das aktuelle Verzeichnis.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>echo [-n] Zeichenfolge ...</b>                         | Schreibt die aufgelisten Zeichenfolgen in die Standardausgabe der Shell. Die Ausgabe wird durch Leerzeichen getrennt und endet mit einem Zeilenvorschubzeichen, es sei denn, die Option <b>-n</b> wurde angegeben.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>else</b><br><b>end</b><br><b>endif</b><br><b>endsw</b> | Beschreibungen zu den Anweisungen <b>else</b> , <b>end</b> , <b>endif</b> und <b>endsw</b> befinden sich bei den Beschreibungen zu den Anweisungen <b>foreach</b> , <b>if</b> , <b>switch</b> und <b>while</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <b>eval Parameter ...</b>                                 | Liest den Wert der Variablen <i>Parameter</i> als Eingabe für die Shell und führt die resultierenden Befehle in der aktuellen Shell aus. Dieser Befehl sollte verwendet werden, um Befehle auszuführen, die als das Ergebnis von Befehls- oder Variablensubstitution generiert wurden, da die Syntaxanalyse vor diesen Substitutionen erfolgt.                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>exec Befehl</b>                                        | Führt anstelle der aktuellen Shell den angegebenen Befehl aus.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |



**exit [(Ausdruck) ]**

Verläßt die Shell entweder mit dem Wert der Shell-Variablen **status** (falls kein Ausdruck angegeben wurde) oder mit dem Wert des angegebenen Ausdrucks.

**fg**

**fg %Job ...**

Bringt den aktuellen Job oder die angegebenen Jobs in den Vordergrund und führt sie weiter aus, wenn sie gestoppt wurden.

**foreach Name (Liste)**

**end**

Setzt nacheinander die Variable *Name* auf jedes in der Variablen *Liste* angegebene Member und führt die Befehlsfolge zwischen der Anweisung **foreach** und der übereinstimmenden Anweisung **end** aus. Beide Anweisungen, **foreach** und **end**, müssen getrennt auf separaten Zeilen stehen.

Der Befehl verwendet die Anweisung **continue**, um mit der Schleife fortzufahren, und die Anweisung **break**, um die Schleife vorzeitig zu beenden. Wird dieser Befehl an der Datenstation eingegeben, wird die Schleife einmal gelesen und dann die Eingabeaufforderung ? (Fragezeichen) ausgegeben, bevor eine Anweisung in der Schleife ausgeführt wird. Wurde bei der Eingabe der Schleife ein Fehler gemacht, kann er vor Ausführung der Schleife korrigiert werden. Befehle innerhalb von Schleifen, die mit der Eingabeaufforderung ? bestätigt werden, werden nicht in die Protokolliste gestellt.

**glob Liste**

Arbeitet wie die Anweisung **echo**, erkennt aber das Escape-Zeichen \ (umgekehrter Schrägstrich) nicht; grenzt außerdem Wörter in der Ausgabe durch Nullzeichen ab. Dieser Befehl ist nützlich, wenn die Shell zur Ausführung der Dateinamenssubstitution zur Erweiterung von Wortlisten verwendet wird.

**goto Wort**

Fährt mit der Ausführung nach der in der Variablen *Wort* angegebenen Zeile fort. Das angegebene Wort ist ein erweiterter Dateiname und Befehl, die eine Zeichenfolge in dem Format bilden, das in der Variablen *Kennsatz* angegeben ist. Die Shell rekonstruiert ihre Eingabe so weit wie möglich und sucht nach einer Zeile im Format *Kennsatz*, der möglicherweise Leerzeichen oder Tabulatorzeichen voranstehen.

**history [-r | -h [n]]**

Zeigt die Protokollereignisliste an. Die ältesten Ereignisse werden zuerst angezeigt. Wird eine Zahl *n* angegeben, wird nur die angegebene Anzahl der jüngsten Ereignisse angezeigt. Die Option **-r** dreht die Reihenfolge, in der die Ereignisse angezeigt werden, um, so daß das jüngste Ereignis zuerst angezeigt wird. Mit der Option **-h** wird die Protokolliste ohne führende Nummern angezeigt. Diese Option sollte verwendet werden, um Dateien zu erzeugen, die für die Verwendung der Option **-h** des Befehls **source** geeignet sind.

**if (Ausdruck) Befehl**

Führt den angegebenen Befehl (mit seinen Argumenten) aus, wenn der angegebene Ausdruck wahr ist. Die Variablensubstitution für die Variable *Befehl* findet zu einem frühen Zeitpunkt statt, zur gleichen Zeit wie für den Rest der Anweisung **if**. Der angegebene Befehl muß ein einfacher Befehl sein (und keine Pipeline, Befehlsliste oder Liste mit Klammerbefehlen).

**Hinweis:** Die Ein- und Ausgabeadressierung erfolgt auch, wenn die Variable *Ausdruck* falsch ist und der Befehl nicht ausgeführt wird.

**if** (*Ausdruck*) **then**

...

**else if** (*Ausdruck2*) **then**

...

**else**

...

**endif**

Führt, wenn die Variable *Ausdruck* wahr ist, die Befehle aus, die auf die erste Anweisung **then** folgen. Ist der **else if** *Ausdruck2* wahr, führt der Befehl **endif** die Befehle aus, die auf die zweite Anweisung **then** folgen. Die Anweisung **else** führt die Befehle aus, die auf das zweite **else** folgen. Eine beliebige Anzahl von **else-if**-Paaren ist möglich. Es ist nur eine Anweisung **endif** erforderlich. Das Segment **else** ist wahlfrei. Die Wörter **else** und **endif** können nur am Anfang von Eingabezeilen verwendet werden. Das Segment **if** darf nur allein auf einer Eingabezeile stehen oder nach einem Befehl **else**.

**jobs** [-l]

Listet die aktiven Jobs auf. Ist die Option **-l** gesetzt, listet der Befehl **jobs** zusätzlich zur Jobnummer auch Prozeß-IDs auf.

**kill** -l | [--Signal] %Job... | PID...

Sendet entweder das Signal **TERM** (beenden) oder das in der Variablen *Signal* angegebene Signal zu den angegebenen Jobs oder Prozessen. Signale werden entweder durch eine Nummer oder einen Namen angegeben (wie in der Datei */usr/include/sys/signal.h* definiert, wobei das Präfix **SIG** übergangen wird). Mit der Option **-l** werden die Signalnamen aufgelistet.

**limit** [-h] [*Ressource* [*maxBel*]]

Begrenzt die Belegung der angegebenen Ressource durch den aktuellen Prozeß und jeden von ihm erstellten Prozeß. Prozeßressourcengrenzen werden in der Datei */etc/security/limits* definiert. Kontrollierbare Ressourcen sind CPU-Zeit (Zeit der Zentraleinheit), Dateigröße, Datengröße, Kernspeicherauszugsgröße und Speicherbelegung. Die maximal zulässigen Werte für diese Ressourcen werden mit dem Befehl **mkuser** gesetzt, wenn der Benutzer dem System hinzugefügt wird. Mit dem Befehl **chuser** werden sie geändert.

Grenzwerte können entweder veränderlich oder fest sein. Der Benutzer kann die veränderlichen Grenzwerte bis zur oberen, von den festen Grenzwerten festgelegten Grenze erhöhen. Um den veränderlichen Grenzwert über den festen Grenzwert hinaus zu erhöhen oder um den festen Grenzwert zu ändern, muß die Root-Berechtigung vorliegen. Mit der Option **-h** werden anstelle der veränderlichen Grenzwerte die festen Grenzwerte angezeigt.

Wird der Parameter *maxBel* nicht angegeben, zeigt der Befehl **limit** den aktuellen Grenzwert für die angegebene Ressource an. Wird der Parameter *Ressource* nicht angegeben, zeigt der Befehl **limit** die aktuellen Grenzwerte für alle Ressourcen an. Weitere Informationen über Ressourcen, die von dem Unterbefehl **limit** gesteuert werden, enthalten die Abschnitte zu den Unterroutinen **getrlimit**, **setrlimit** oder **vlimit**.

Der Parameter *maxBel* für die CPU-Zeit wird im Format hh:mm:ss angegeben. Der Parameter *maxBel* für andere Ressourcen wird als Gleitkommazahl oder als ganze Zahl angegeben, auf die wahlweise ein Maßstabsfaktor folgt. Der Maßstabsfaktor ist k (Kilobyte = 1024 Byte), m (Megabyte) oder b (Blöcke), d. h. Einheiten, die von der Unterroutine **ulimit** verwendet werden. Wird kein Maßstabsfaktor angegeben, wird b für die Dateigröße angenommen und k für die anderen Ressourcen. Sowohl für Ressourcennamen als auch für Maßstabsfaktoren genügen eindeutige Präfixe zu den Namen.

**Hinweis:** Dieser Befehl begrenzt den physischen Speicher (die Speicherbelegung), der für einen Prozeß zur Verfügung steht, nur, wenn durch andere aktive Prozesse im Systemspeicher eine Konkurrenzsituation besteht.

**login** Beendet die Anmelde-Shell und ersetzt sie durch ein Exemplar des Befehls **/usr/sbin/login**. Das ist eine Art der Abmeldung (aus Gründen der Kompatibilität mit den Befehlen **ksh** und **bsh** eingefügt).

**logout** Beendet die Anmelde-Shell. Dieser Befehl ist besonders nützlich, wenn die Option **ignoreeof** gesetzt ist.

**newgrp** Führt den Befehl **newgrp** im aktuellen Shell-Prozeß aus. Der Befehl **newgrp** ändert die Primärgruppen-ID des Benutzers in die angegebene Gruppen-ID.

**Hinweis:** Jede benutzererstellte aktive Shell wird beendet, wenn der Befehl **newgrp** eingegeben wird.

**nice**

**nice +n**

**nice Befehl**

**nice +n Befehl** Setzt die Priorität, mit der Befehle in dieser Shell ausgeführt werden, auf 24 (erstes Format). Mit dem zweiten Format wird die Priorität auf die angegebene Zahl gesetzt. Die letzten zwei Formate setzen den angegebenen Befehl auf die Priorität 24 bzw. auf die entsprechende Zahl (*n*). Besteht Root-Berechtigung, kann die Anweisung **nice** mit einer negativen Zahl angegeben werden. Der Befehl wird immer in einer Subshell ausgeführt und es gelten die Beschränkungen, die Befehlen in einfachen **if**-Anweisungen auferlegt werden.

**nohup**

**nohup Befehl** Führt dazu, daß nichtprogrammierte Stopps für den Rest der Prozedur ignoriert werden (erstes Format). Mit dem zweiten Format wird der angegebene Befehl ausgeführt, wobei nichtprogrammierte Stopps ignoriert werden. Um eine Pipeline oder Liste von Befehlen mit diesem Format auszuführen, die Pipeline oder Liste in eine Shell-Prozedur stellen, der Prozedur Ausführungsberechtigung erteilen und sie als Wert der Variablen *Befehl* verwenden. Alle Prozesse, die mit einem & (Et-Zeichen) im Hintergrund ausgeführt werden, sind beim Abmelden des Benutzers wirksam gegen ein Signal für nichtprogrammierten Stopp geschützt. Sie können jedoch von explizit gesendeten Signalen für nichtprogrammierten Stopp betroffen sein, es sei denn, die Anweisung **nohup** wird verwendet.

**notify**

**notify %Job ...**

Führt dazu, daß der Benutzer von der Shell asynchron beachrichtigt wird, wenn sich der Status des aktuellen oder des angegebenen Jobs ändert.

Normalerweise gibt die Shell Hinweise aus, kurz bevor sie die Shell-Eingabeaufforderung anzeigt. Diese Funktion ist automatisch, wenn die Shell-Variable **notify** gesetzt ist.

#### **onintr**

**onintr -**

**onintr** *Kennsatz*

Steuert bei Unterbrechungen die Aktionen der Shell. Mit dem ersten Format wird die Standardaktion der Shell bei Unterbrechungen wiederhergestellt, die die Shell-Prozedur beendet oder zur Befehlseingabeebene der Datenstation zurückkehrt. Mit dem zweiten Format werden alle Unterbrechungen ignoriert. Mit dem dritten Format führt die Shell die Anweisung **goto** *Kennsatz* aus, wenn sie eine Unterbrechung empfängt oder wenn ein Sohnprozeß aufgrund einer Unterbrechung endet. Auf jeden Fall verlieren alle Formate der Anweisung **onintr** ihre Bedeutung, wenn die Shell im Hintergrund ausgeführt wird und Unterbrechungen ignoriert werden. Dann werden Unterbrechungen weiterhin von der Shell und allen aufgerufenen Befehlen ignoriert.

#### **popd**

**popd** *+n*

Zeigt den Verzeichnisstapel an und kehrt zum neuen Hauptverzeichnis zurück. Wird die Variable *a +n* angegeben, löscht der Befehl den *n*ten Eintrag im Stapel. Die Elemente im Verzeichnisstapel sind von oben nach unten, angefangen mit 0, durchnummeriert.

#### **pushd**

**pushd** *Name*

**pushd** *+n*

Tauscht, wenn keine Argumente angegeben werden, die zwei obersten Elemente im Verzeichnisstapel aus. Wird die Variable *Name* angegeben, wechselt der Befehl zum neuen Verzeichnis und legt das alte aktuelle Verzeichnis (wie in der Shell-Variablen **cwd** angegeben) im Verzeichnisstapel ab. Wird die Variable *a +n* angegeben, stellt der Befehl die *n*te Komponente des Verzeichnisstapels an den Anfang und wechselt dorthin. Die Elemente im Verzeichnisstapel sind von oben nach unten, angefangen mit 0 durchnummeriert.

#### **rehash**

Führt dazu, daß die interne Hash-Tabelle mit dem Inhalt der in der Shell-Variablen **path** angegebenen Verzeichnisse neu berechnet wird. Diese Aktion ist erforderlich, wenn den Verzeichnissen in der Shell-Variablen **path** neue Befehle hinzugefügt werden, während der Benutzer angemeldet ist. Der Befehl **rehash** ist nur erforderlich, wenn Befehle einem der benutzereigenen Verzeichnisse hinzugefügt werden oder wenn der Inhalt eines der Benutzerverzeichnisse geändert wird.

**repeat** *Anzahl* *Befehl*

Führt den angegebenen Befehl die angegebene Zahl von Malen aus, und zwar unter denselben Beschränkungen wie die Anweisung **if**.

**Hinweis:** Die E/A-Umadressierung wird genau einmal ausgeführt, auch wenn die Variable *Anzahl* mit 0 (Null) angegeben wurde.

#### **set**

**set** *Name*

**set** *Name=Wort*

**set** *Name[n]=Wort*

**set** *Name=(Liste)*

Zeigt die Werte aller Shell-Variablen (erstes Format). Variablen, die als Wert mehr als ein Einzelwort enthalten, werden als Wortliste in Klammern

angezeigt. Mit dem zweiten Format wird der Wert der Variablen *Name* auf die leere Zeichenfolge gesetzt. Mit dem vierten Format wird die *rte* Komponente der Variablen *Name* auf den Wert der Variablen *Wort* gesetzt; diese Komponente muß bereits existieren. Mit dem letzten Format wird die Variable *Name* auf die Liste der Wörter gesetzt, die in der Variablen *Liste* angegeben wird. In allen Fällen besteht der Wert aus erweiterten Befehlen und Dateinamen. Diese Argumente werden möglicherweise wiederholt, um Mehrfachwerte in einem einzelnen Befehl **set** zu setzen. Dennoch werden für alle Argumente die Variablen erweitert, bevor jegliche Einstellung erfolgt.

#### **setenv** *Name Wert*

Setzt den Wert der in der Variablen *Name* angegebenen Umgebungsvariablen auf *Wert*, eine einfache Zeichenfolge. Die am häufigsten verwendeten Umgebungsvariablen, **USER**, **TERM**, **HOME** und **PATH**, werden automatisch von und zu den C-Shell-Variablen **user**, **term**, **home** und **path** ex- und importiert. Für diese Variablen ist keine Anweisung **setenv** erforderlich.

Werden die Umgebungsvariablen **LANG** oder **LC\_COLLATE** geändert, werden die aktuelle Umgebung für die internationale Zeichenunterstützung und die Sortierfolge für nachfolgende Befehle, die von der Shell ausgeführt werden, wie angegeben geändert.

**shift** [*Variable*] Verschiebt die Member der Shell-Variable **argv** oder der angegebenen Variablen nach links. Ist die Shell-Variablen **argv** nicht gesetzt oder hat sie als Wert weniger als ein Wort, tritt ein Fehler auf.

#### **source** [-h] *Name*

Liest die in der Variablen *Name* angegebenen Befehle. Die Befehle **source** können verschachtelt werden. Wenn sie zu sehr verschachtelt sind, kann es allerdings vorkommen, daß die Shell nicht mehr über genügend Dateideskriptoren verfügt. Ein Fehler in einem Befehl **source** auf einer beliebigen Ebene beendet alle verschachtelten Befehle **source**. Normalerweise wird die Eingabe während der Befehle **source** nicht in die Protokolliste gestellt. Mit der Option **-h** können die Befehle in die Protokolliste gestellt werden, ohne ausgeführt zu sein.

#### **stop**

**stop** %*Job* ... Stoppt die aktuellen oder angegebenen Jobs im Hintergrund.

**suspend** Stoppt die Shell, als wenn das Signal **STOP** empfangen worden wäre.

#### **switch** (*Zeichenfolge*)

**case** *Zeichenfolge*:

...

**breaksw**

**default**:

...

**breaksw**

**endsw**

Gleicht nacheinander alle Groß-/Kleinschreibungskennsätze gegen den Wert der Variablen *Zeichenfolge* ab. Die Zeichenfolge besteht aus zuerst erweiterten Befehlen und Dateinamen. Die Platzhalterzeichen \* (Stern), ? (Fragezeichen) und [ . . . ] (Auslassungen umschließende eckige Klammern) in den Kennsätzen **case** verwenden, die variablen erweitert sind. Stimmen keine Kennsätze überein, bevor der Kennsatz **default** gefunden wird, beginnt die Ausführung nach dem Kennsatz **default**. Die Kennsätze **case** und **default** müssen jeweils am Anfang der Zeile stehen. Mit dem Befehl **breaksw** wird die Ausführung nach dem Befehl **endsw** fortgesetzt.

Andernfalls werden alle Kennsätze **case** und **default** weiter ausgeführt. Stimmt kein Kennsatz überein und ist kein Wert für **default** gesetzt, wird die Ausführung nach dem Befehl **endsw** fortgesetzt.

**time**  
**time Befehl**

Wird keine Befehlsvariable angegeben, zeigt der Befehl **time** die Gesamtzeit an, die von der Shell und ihren Söhnen verbraucht wurde. Wird ein Befehl mit der Variablen *Befehl* angegeben, ist er zeitverzögert. Die Shell zeigt dann eine Zeitübersicht an, wie unter der Shell-Variablen **time** beschrieben. Falls erforderlich, wird eine zusätzliche Shell erstellt, um die Zeitstatistik anzuzeigen, wenn der Befehl beendet ist.

Der Befehl **time** steuert die Ablaufsteuerung von Befehlen. Er kann mit einem oder zwei Werten verwendet werden. Der erste Wert zeigt die Schwelle in CPU-Sekunden an. Der zweite Wert ist eine Zeichenfolge von Kennungen und Text, die kennzeichnet, zu welchen Ressourcen Informationen angezeigt werden sollen.

Eine Kennung ist ein Prozentzeichen (%), gefolgt von einem einzelnen Großbuchstaben. Nicht als solche erkannte Kennungen werden als Text ausgegeben. Falls gesetzt, führt jede Anzahl an CPU-Sekunden dazu, daß am Ende eine Berichtszeile angezeigt wird.

Die Standardübersicht zeigt die Ausgabe der folgenden Kennungen an und zwar in dieser Reihenfolge: %U , %S, %E, %P, %X, %D, %I, %O, %F und %W.

Die Shell erkennt die folgenden Kennungen:

|    |                                                                                                    |
|----|----------------------------------------------------------------------------------------------------|
| %D | Durchschnittsgröße des nicht gemeinsam benutzten Datenraums in Kilobyte                            |
| %E | für den Befehl abgelaufene Zeit                                                                    |
| %F | Fehlseifenbedingungen                                                                              |
| %I | Anzahl der Blockeingabeoperationen                                                                 |
| %K | Durchschnittsgröße des nicht gemeinsam benutzten Stapelspeicherbereichs in Kilobyte                |
| %M | maximaler Realspeicher, der während der Ausführung des Prozesses verwendet wurde                   |
| %O | Anzahl der Blockausgabeoperationen                                                                 |
| %P | gesamte CPU-Zeit – U (Benutzerzeit) plus S (Systemzeit) – als Prozentsatz von E (abgelaufene Zeit) |
| %S | Anzahl der Sekunden, die vom Kernel aufgrund des Benutzerprozesses an CPU-Zeit verbraucht wurden   |
| %U | Anzahl an Sekunden, die an CPU-Zeit für den Benutzerprozeß verbraucht wurden                       |
| %W | Anzahl der Austauschungen                                                                          |
| %X | Durchschnittsgröße des gemeinsam benutzten Speichers in Kilobyte                                   |

**umask**  
**umask Wert** Zeigt die Dateierstellungsmaske an oder setzt sie auf den in der Variablen *Wert* angegebenen Wert. Die Maske wird als oktaler Wert angegeben. Gebräuchliche Werte für die Maske sind 002, mit dem der Eigner und die Gruppe Gesamtzugriff und alle anderen Schreib- und Ausführungszugriff erhalten, oder 022, mit dem der Eigner Gesamtzugriff erhält und die Benutzer in der Gruppe oder andere Gesamtzugriff, aber ohne Schreibzugriff, erhalten.

**unalias Muster** Löscht alle Alias-Namen, deren Name mit der Variablen *Muster* übereinstimmt. Mit dem Befehl **unalias \*** werden alle Alias-Namen gelöscht. Sind keine Alias-Namen vorhanden, führt das nicht zu einem Fehler.

**unhash** Inaktiviert die Verwendung der internen Hash-Tabelle bei der Suche nach aktiven Programmen.

**unlimit [-h][Ressource ]**  
 Löscht die Grenzwerte der Variablen *Ressource*. Ist keine Variable *Ressource* angegeben, werden alle Grenzwerte *Ressource* entfernt. Die Beschreibung des Befehls **limit** enthält eine Liste der Ressourcennamen. Die Option **-h** löscht die entsprechenden festen Grenzwerte. Nur ein Benutzer mit Root-Berechtigung kann feste Grenzwerte ändern.

**unset Muster** Löscht alle Variablen, die mit der Variablen *Muster* übereinstimmen. Mit dem Befehl **unset \*** werden alle Variablen gelöscht. Werden keine Variablen gelöscht, tritt dennoch kein Fehler auf.

**unsetenv Muster**  
 Löscht alle Variablen aus der Umgebung, deren Name mit dem angegebenen *Muster* übereinstimmt. (Siehe den integrierten Befehl **setenv**.)

**wait** Wartet auf alle Hintergrundjobs. Ist die Shell interaktiv, wird der Wartestatus durch ein INTERRUPT (die Tastenfolge Strg-C) unterbrochen. Die Shell zeigt dann die Namen und Jobnummern von allen anstehenden Jobs an.

**while (Ausdruck)**  
 ...  
**end** Wertet die Befehle zwischen der Anweisung **while** und der übereinstimmenden Anweisung **end** aus, solange das Ergebnis der Auswertung des in der Variablen *Ausdruck* angegebenen Ausdrucks ungleich Null ist. Mit der Anweisung **break** kann die Schleife vorzeitig beendet und mit der Anweisung **continue** fortgesetzt werden. Die Anweisungen **while** und **end** müssen separat in ihrer Eingabezeile stehen. Erfolgt die Eingabe über die Datenstation, erscheinen beim ersten Lauf durch die Schleife Eingabeaufforderungen, wie bei der Anweisung **foreach**.

**@**  
**@ Name = Ausdruck**  
**@ Name[n]= Ausdruck**  
 Zeigt die Werte aller Shell-Variablen an (erstes Format). Mit dem zweiten Format wird der in der Variablen *Name* angegebene Name auf den Wert der Variablen *Ausdruck* gesetzt. Enthält der Ausdruck die Zeichen < (kleiner als), > (größer als), & (Et-Zeichen) oder | (Balken), muß dieser Teil des Ausdrucks in Klammern gesetzt werden. Mit dem dritten Format wird der Wert der Variablen *Ausdruck* der *n*ten Komponente der Variablen *Name* zugeordnet. Sowohl die Variable *Name* als auch ihre *n*te Komponente müssen bereits existieren.

Operatoren der Programmiersprache C, wie \*= (Stern, Gleichheitszeichen) und += (Pluszeichen, Gleichheitszeichen) sind verfügbar. Das Leerzeichen zwischen der Variablen *Name* und dem Zuordnungsoperator ist wahlfrei. Leerzeichen sind allerdings erforderlich, um die Komponenten der Variablen *Ausdruck* voneinander zu trennen, die andernfalls als ein Einzelwort gelesen würden. Besondere Suffixoperatoren, ++ (doppeltes Pluszeichen) und -- (doppelter Bindestrich) erhöhen oder vermindern jeweils den Wert der Variablen *Name*.

## Ausdrücke

Der integrierte Befehl @ und die Anweisungen **exit**, **if** und **while** akzeptieren Ausdrücke, die Operatoren beinhalten, die denen der Programmiersprache C ähnlich sind, mit derselben Vorrangstellung. Folgende Operatoren stehen zur Verfügung:

|    |      |    |    |
|----|------|----|----|
| *  | /    | %  |    |
| +  | --<< | >> |    |
| <= | >=   | >  |    |
| == | !=   | =~ | !~ |

In der vorangegangenen Liste erscheinen Operatoren mit der gleichen Vorrangstellung in derselben Zeile. Die Zeilen mit Operatoren sind von oben nach unten mit abnehmender Vorrangstellung aufgelistet. Die Operatoren ==, !=, =~, und !~ vergleichen ihre Argumente als Zeichenfolgen; alle anderen arbeiten mit Zahlen. Die Operatoren =~ und !~ sind den Operatoren == und != ähnlich, außer es sich bei der ganz rechten Seite um ein *Muster* handelt, gegen das der ganz linke Operand abgeglichen wird. Damit wird die Anweisung **switch** in Shell-Prozeduren nicht mehr so oft erforderlich, wenn nur die Verwendung von Platzhalterzeichen nötig ist.

Die logischen Operatoren || (oder) und && (und) sind ebenfalls verfügbar. Sie können beispielsweise dazu verwendet werden, um Zahlenbereiche zu überprüfen:

```
if ($#argv > 2 && $#argv < 7) then
```

Im vorangegangenen Beispiel muß die Anzahl der Argumente größer als 2 und kleiner als 7 sein.

Zeichenfolgen, die mit 0 (Null) beginnen, werden als Oktalzahlen interpretiert. Leere oder fehlenden Argumente werden als 0 interpretiert. Alle Ausdrücke liefern als Ergebnis Zeichenfolgen, die Dezimalzahlen darstellen. Zu beachten ist, daß zwei Komponenten eines Ausdrucks in demselben Wort stehen können. Ausdruckskomponenten sollten von Leerzeichen umgeben sein, außer wenn sie neben Ausdruckskomponenten stehen, die syntaktisch signifikant für den Parser (& | < > ( ) ) sind.

Als Basisoperanden in Ausdrücken sind außerdem Befehlsausführungen, die in { } (geschweifte Klammern) eingeschlossen sind, vorhanden und Dateiabfragen im Format *-/Name*, wobei / eine der folgenden Optionen ist:

|   |                        |
|---|------------------------|
| r | Lesezugriff            |
| w | Schreibzugriff         |
| x | Ausführungszugriff     |
| e | Bestehen               |
| o | Eignerstatus           |
| z | Größe Null             |
| f | unverschlüsselte Datei |
| d | Verzeichnis            |



Bei dem angegebenen Namen werden Befehle und Dateinamen erweitert und dann überprüft, ob die angegebene Abhängigkeit zum realen Benutzer besteht. Existiert die Datei nicht oder besteht keine Zugriffsmöglichkeit, werden alle Abfragen als falsch (0) zurückgegeben. Die Befehlsausführung ist erfolgreich (der Wert wahr (1) wird zurückgegeben), wenn der Befehl mit dem Status 0 endet. Andernfalls wird der Wert falsch (0) zurückgegeben. Sind zusätzliche Statusinformationen erforderlich, den Befehl außerhalb eines Ausdrucks ausführen und dann die Shell-Variable **status** überprüfen.

---

## Befehlssubstitution in der C-Shell

Bei der *Befehlssubstitution* führt die Shell einen bestimmten Befehl aus und ersetzt diesen Befehl durch die Ausgabe. Zur Befehlsausführung in der C-Shell muß der Befehl oder die Befehlsfolge in ' ' (Gravis) eingeschlossen sein. Die Shell unterteilt die Ausgabe des Befehls normalerweise bei Leerzeichen, Tabulatorzeichen und Zeilenvorschubzeichen in einzelne Wörter. Danach wird der ursprüngliche Befehl durch diese Ausgabe ersetzt.

Im folgenden Beispiel zeigen die ' ' (Gravis) um den Befehl **date**, daß die Ausgabe des Befehls ersetzt werden wird:

```
echo Aktuelles Datum, aktuelle Zeit: `date`
```

Die Ausgabe dieses Befehls könnte folgendermaßen aussehen:

```
Aktuelles Datum, aktuelle Zeit: Wed Apr 8 13:52:14 CDT 1992
```

Die C-Shell führt eine selektive Befehlssubstitution mit den Argumenten von integrierten Shell-Befehlen durch. Das bedeutet, daß Komponenten von Ausdrücken, die nicht ausgewertet werden, nicht erweitert werden. Bei nicht integrierten Befehlen ersetzt die Shell den Befehlsnamen getrennt von der Argumentenliste. Bei einem Sohn der Haupt-Shell erfolgt die Substitution erst, nachdem die Shell die Ein- oder Ausgabeumadressierung ausgeführt hat.

Ist eine Befehlsfolge von " " (Anführungszeichen) umgeben, interpretiert die Shell nur Zeilenvorschubzeichen als Worttrennzeichen, womit Leerzeichen und Tabulatorzeichen innerhalb eines Wortes erhalten bleiben. In jedem Fall erzwingt ein einzelnes abschließendes Zeilenvorschubzeichen kein neues Wort.

**Hinweis:** Bei der Befehlssubstitution ist es möglich, daß nur ein Teil eines Wortes betroffen ist, auch wenn der Befehl eine ganze Zeile ausgibt.

---

## Protokollsubstitution in der C-Shell

Mit der Protokollsubstitution können einzelne Wörter von vorangegangenen Befehlen geändert werden, um somit neue Befehle zu erstellen. Die Protokollsubstitution erleichtert das Wiederholen von Befehlen, das Wiederholen von Argumenten aus vorangegangenen Befehlen im aktuellen Befehl oder die Berichtigung von Rechtschreibfehlern im vorangegangenen Befehl mit geringem Schreibaufwand.

Protokollsubstitutionen beginnen mit dem ! (Ausrufezeichen) und können überall in der Befehlszeile erscheinen, vorausgesetzt, sie werden nicht verschachtelt (d. h., eine Protokollsubstitution kann keine weiteren Protokollsubstitutionen enthalten). Wird dem ! (Ausrufezeichen) ein \ (umgekehrter Schrägstrich) vorangestellt, wird damit die Sonderbedeutung des Ausrufezeichens aufgehoben. Wenn das ! vor ein Leerzeichen, Tabulatorzeichen, Zeilenvorschubzeichen, = (Gleichheitszeichen) oder ( (linke Klammer) gesetzt wird, erfolgt ebenfalls keine Protokollsubstitution.

Protokollsubstitutionen treten auch auf, wenn eine Eingabezeile mit einem ^ (Zirkumflex) beginnt. Die Shell meldet jede Eingabezeile, die Protokollsubstitutionen enthält, zur Datenstation zurück, bevor diese Zeile ausgeführt wird.

## Protokollisten

In der Protokolliste werden Befehle, die die Shell an der Datenstation liest und die aus einem oder mehreren Wörtern bestehen, gespeichert. Aus diesen gespeicherten Wörtern führt die Protokollsubstitution Wortfolgen wieder in den Eingabedatenstrom ein.

Über die Shell-Variable **history** wird die Größe der Protokolliste gesteuert. Die Shell-Variable **history** muß entweder in der Datei **.cshrc** gesetzt werden oder in der Befehlszeile mit dem integrierten Befehl **set**. Der vorherige Befehl wird unabhängig vom Wert der Variablen **history** immer beibehalten. Befehle in der Protokolliste werden, beginnend mit 1, durchnummeriert. Der integrierte Befehl **history** erzeugt eine Ausgabe, ähnlich der folgenden:

```
9 write michael
10 ed write.c
11 cat oldwrite.c
12 diff *write.c
```

Die Shell zeigt die Befehlsfolgen mit ihren Ereignisnummern an. Normalerweise ist es nicht erforderlich, mit Ereignisnummern auf Ereignisse zu verweisen. Dennoch kann die aktuelle Ereignisnummer als Teil der Eingabeaufforderung angezeigt werden, wenn ein ! (Ausrufezeichen) in der Eingabeaufforderungszeichenfolge gesetzt wird, die der Umgebungsvariablen **prompt** zugeordnet ist.

Ein vollständiger Protokollverweis enthält eine Ereignisspezifikation, eine Wortkennung und einen oder mehrere Änderungswerte im folgenden allgemeinen Format:

*Ereignis[.]Wort:Änderungswert[:Änderungswert] . . .*

**Hinweis:** Es kann nur ein Wort geändert werden. Eine Zeichenfolge, die Leerzeichen enthält, ist nicht zulässig.

Im vorherigen Beispiel für die Ausgabe des Befehls **history** ist die aktuelle Ereignisnummer 13. Bei Verwendung dieses Beispiels verweisen die folgenden Formate auf vorherige Ereignisse:

|       |                                                                               |
|-------|-------------------------------------------------------------------------------|
| !10   | Ereignisnummer 10                                                             |
| !-2   | Ereignisnummer 11 (das aktuelle Ereignis minus 2)                             |
| !d    | Befehlsword, das mit <code>d</code> beginnt (Ereignisnummer 12)               |
| !mic? | Befehlsword, das die Zeichenfolge <code>mic</code> enthält (Ereignisnummer 9) |

Mit diesen Formaten werden die Wörter der angegebenen Ereignisse, durch einzelne Leerzeichen voneinander getrennt, ohne weitere Änderungen einfach wieder eingeführt. Als Sonderfall verweist das Zeichen `!!` (doppeltes Ausrufezeichen) auf den vorherigen Befehl. Der Befehl `!!` allein auf einer Eingabezeile führt den vorherigen Befehl erneut aus.

## Ereignisspezifikation

Um Wörter eines Ereignisses auszuwählen, hinter die Ereignisspezifikation einen : (Doppelpunkt) und eine der folgenden Wortkennungen stellen (die Wörter einer Eingabezeile sind, beginnend mit 0, durchnummeriert):

|     |                                                                                                                   |
|-----|-------------------------------------------------------------------------------------------------------------------|
| 0   | erstes Wort (der Befehlsname)                                                                                     |
| n   | das <i>n</i> te Argument                                                                                          |
| ^   | erstes Argument                                                                                                   |
| \$  | letztes Argument                                                                                                  |
| %   | Wort, das mit einer unmittelbar voranstehenden Suche <i>?Zeichenfolge?</i> übereinstimmt                          |
| x-y | Wortbereich vom <i>x</i> ten Wort bis zum <i>y</i> ten Wort                                                       |
| -y  | Wortbereich vom ersten Wort (0) bis zum <i>y</i> ten Wort                                                         |
| *   | vom ersten bis zum letzten Argument, bzw. keine Auswahl, wenn im Ereignis nur ein Wort vorkommt (der Befehlsname) |
| x*  | vom <i>x</i> ten bis zum letzten Argument                                                                         |
| x-  | wie <i>x*</i> , wobei allerdings das letzte Wort übergangen wird                                                  |

Der Doppelpunkt, der die Ereignisspezifikation von der Wortkennung trennt, kann ausgelassen werden, wenn die Wortkennung mit einem `^` (Zirkumflex), `$` (Dollarzeichen), `*` (Stern), `-` (Bindestrich) oder `%` (Prozentzeichen) beginnt. Außerdem kann einer oder mehrere der folgenden Änderungswerte nach der wahlweise zu verwendenden Wortkennung eingegeben werden, jeder mit einem vorangestellten Doppelpunkt:

|   |                                                                                             |
|---|---------------------------------------------------------------------------------------------|
| h | Entfernt abschließende Pfadnamenerweiterungen, wobei der Kopf bestehen bleibt.              |
| r | Entfernt abschließende <code>.xxx</code> -Komponenten, wobei der Root-Name bestehen bleibt. |
| e | Entfernt alles außer der abschließenden Erweiterung <code>.xxx</code> .                     |

### **s/AltWort/NeuWort/**

Ersetzt den Wert der Variablen *AltWort* durch den Wert der Variablen *NeuWort*.

Die linke Seite der Substitution ist weniger ein Muster im Sinne einer Zeichenfolge, die von einem Editor erkannt wird; eher eine Einzeleinheit ohne Leerzeichen, ein Wort. Normalerweise wird das ursprüngliche Wort (*AltWort*) durch einen / (Schrägstrich) von seinem Ersatzwort (*NeuWort*) abgetrennt. Es kann jedoch ein beliebiges Zeichen als Begrenzer verwendet werden, wenn diesem Zeichen ein \ (umgekehrter Schrägstrich) voransteht. Im folgenden Beispiel ist es durch die Verwendung von `\%` (umgekehrter

Schrägstrich, Prozentzeichen) als Begrenzer möglich, daß die Wörter einen / (Schrägstrich) enthalten:

```
s\%/home/meindat\%/home/deindat\%
```

Die Shell ersetzt ein & (Et-Zeichen) in der Variablen *NeuWort* durch den Text von *AltWort*. Im folgenden Beispiel wird /home/meindat zu /temp/home/meindat.

```
s\%/home/meindat\%/temp&\%
```

Die Shell ersetzt ein leeres Wort in einer Substitution entweder mit der letzten Substitution oder mit der letzten Zeichenfolge, die in der kontextbezogenen Suche *!?Zeichenfolge?* verwendet wurde. Das abschließende Begrenzungszeichen /) kann übergangen werden, wenn unmittelbar ein Zeilenvorschubzeichen folgt. Die folgenden Änderungswerte verwenden, um die Protokolliste zu begrenzen:

- t** Entfernt alle führenden Pfadnamekomponenten, wobei der Nachsatz bestehen bleibt.
- &** Wiederholt die vorherige Substitution.
- g** Wendet die Änderung global an, d. h. mit **g&**.
- p** Zeigt den neuen Befehl an, ohne ihn auszuführen.
- q** Setzt die ersetzten Wörter in Anführungszeichen, um weitere Substitutionen zu verhindern.
- x** Wie der Änderungswert **q**, trennt aber bei Leerzeichen, Tabulatorzeichen und Zeilenvorschubzeichen Wörter ab.

Außer, wenn der Änderungswert **g** den beschriebenen Änderungswerten vorangeht, wird jeweils nur das erste modifizierbare Wort verändert.

Wird ein Protokollverweis ohne Ereignisspezifikation angegeben (zum Beispiel !\$), verwendet die Shell den vorherigen Befehl als Ereignis. Tritt ein vorheriger Protokollverweis in derselben Zeile auf, wiederholt die Shell diesen vorherigen Verweis. Demzufolge ergibt die folgende Anweisung das erste und letzte Argument des Befehls, der mit ?foo? übereinstimmt.

```
!?foo?^ !$
```

Eine besondere Abkürzung eines Protokollverweises tritt auf, wenn das erste belegte Zeichen einer Eingabezeile ein ^ (Zirkumflex) ist. Es ist äquivalent zu !:s^ und liefert so eine passende Kurzform für Substitutionen, die den Text der vorherigen Zeile betreffen. Mit dem Befehl ^ 1b^ 1ib wird die Rechtschreibung von "lib" im vorherigen Befehl korrigiert.

Falls erforderlich, kann eine Protokollsubstitution in { } (geschweifte Klammern) eingeschlossen werden, um sie von den folgenden Zeichen abzutrennen. Soll beispielsweise ein Verweis auf den Befehl:

```
ls -ld ~paul
```

verwendet werden, um den Befehl:

```
ls -ld ~paula
```

auszuführen, folgende Schreibweise verwenden:

```
!{1}a
```

In diesem Beispiel sucht !1a nach einem Befehl, der mit 1a beginnt.

## **Die Verwendung von Anführungszeichen und halben Anführungszeichen**

Zeichenfolgen in Anführungszeichen oder halbe Anführungszeichen einschließen, wenn alle oder einige der verbleibenden Substitutionen verhindert werden sollen. Das Einschließen von Zeichenfolgen in ' ' (halbe Anführungszeichen) verhindert eine weitere Interpretation, während das Einschließen von Zeichenfolgen in " " (Anführungszeichen) Erweiterungen ermöglicht. In beiden Fällen wird der resultierende Text (ganz oder teilweise) zu einem einzelnen Wort. Nur in einem speziellen Fall ergibt eine Zeichenfolge, die in Anführungszeichen eingeschlossen ist, mehr als ein Wort; bei Zeichenfolgen in halben Anführungszeichen tritt dieser Fall nie auf.

---

## Alias-Namensubstitution in der C-Shell

Ein *Alias-Name* ist ein Name, der einem Befehl oder einer Befehlsfolge zugeordnet ist. Mit der C-Shell ist es möglich, Alias-Namen zuzuordnen und sie wie Befehle zu verwenden. Die Shell pflegt eine Liste der definierten Alias-Namen.

Nachdem die Shell die Befehlszeile durchsucht hat, teilt sie sie in bestimmte Befehle auf und prüft das erste Wort in jedem Befehl, von links nach rechts, ob es einen Alias-Namen enthält. Ist das der Fall, verwendet die Shell die Protokollfunktion, um den Text des Alias-Namen durch den Text des Befehls zu ersetzen, auf den durch den Alias-Namen verwiesen wird. Die resultierenden Wörter ersetzen den Befehl und die Argumentenliste. Ohne einen Verweis auf die Protokolliste bleibt die Argumentenliste unverändert.

Weitere Informationen über die Protokollfunktion der C-Shell enthält der Abschnitt "Protokollsubstitution in der C-Shell" auf Seite C-14.

Die integrierten Befehle **alias** und **unalias** richten die Alias-Namenliste ein, zeigen sie an und ändern sie. Der Befehl **alias** wird im folgenden Format verwendet:

```
alias [Name [Wortliste]]
```

Die wahlfreie Variable *Name* gibt den Alias-Namen für den angegebenen Namen an. Wird mit der Variablen *Wortliste* eine Wortliste angegeben, ordnet der Befehl sie der Variablen *Name* als Alias-Name zu. Wird der Befehl **alias** ohne eine der wahlfreien Variablen eingegeben, werden alle Alias-Namen der C-Shell angezeigt.

Ist `ls -l` der Alias-Name für den Befehl `ls`, wird der Befehl:

```
ls /usr
```

ersetzt durch den Befehl:

```
ls -l /usr
```

Die Argumentenliste bleibt unverändert, da im Befehl mit dem Alias-Namen kein Verweis auf die Protokolliste enthalten ist. Ist der Alias-Name für den Befehl **lookup**:

```
grep \!^ /etc/passwd
```

ersetzt die Shell `lookup bill` durch:

```
grep bill /etc/passwd
```

In diesem Beispiel bezieht sich `!^` auf die Protokolliste und die Shell ersetzt diese Zeichen durch das erste Argument in der Eingabezeile, in diesem Fall durch `bill`.

In Alias-Namen können besondere Platzhalterzeichen verwendet werden. Der Befehl:

```
alias lprint 'pr &bslash2.!* >> print'
```

erstellt einen Befehl, der seine Argumente auf dem Zeilendrucker ausgibt. Das `!` wird innerhalb des Alias-Namens vor der Shell geschützt, so daß es nicht erweitert wird, bis der Befehl `pr` ausgeführt wird.

Trifft die Shell auf einen Alias-Namen, ersetzt sie den Eingabetext und führt mit der damit gebildeten Eingabezeile eine erneute Alias-Namensubstitution durch. Stimmt das erste Wort des neuen Textes mit dem des alten Textes überein, wird eine Schleife dadurch verhindert, daß dort eine Markierung zur Beendigung der Alias-Namensubstitution gesetzt wird. Andere nachfolgende Schleifen werden festgestellt und führen zu einem Fehler.

---

## Variablensubstitution in der C-Shell

Die C-Shell pflegt einen Satz von Variablen, von denen jede als Wert eine Liste von null oder mehr Wörtern hat. Einige dieser Variablen werden von der Shell gesetzt oder es wird auf sie verwiesen. Beispielsweise ist die Variable `argv` ein Abbild der Shell-Variablenliste, und auf Wörter, die den Wert dieser Variablen enthalten, wird auf besondere Weise verwiesen.

Die Werte von Variablen können mit den `set` und `unset` geändert und angezeigt werden. Bei einer Reihe von Variablen, auf die die Shell verweist, handelt es sich um Schalter (Variablen, die etwas ein- und ausschalten). Der Wert dieser Variablen ist für die Shell unerheblich, entscheidend ist, ob sie aktiviert oder inaktiviert sind. Beispielsweise ist die Shell-Variable `verbose` ein Schalter, der dafür sorgt, daß die Befehlseingabe zurückgemeldet wird. Zum Setzen dieser Variablen die Option `-v` in der Befehlszeile eingeben.

Andere Operationen verarbeiten die Variable numerisch. Der Befehl `@` führt numerische Berechnungen aus und ordnet das Ergebnis einer Variablen zu. Die Werte von Variablen werden dennoch immer als (leere oder nicht leere) Zeichenfolgen dargestellt. Bei numerischen Operationen ist eine leere Zeichenfolge gleichbedeutend mit Null und bei Werten mit mehreren Wörtern werden das zweite Wort und alle nachfolgenden ignoriert.

Wird ein Befehl eingegeben, analysiert die Shell die Eingabezeile syntaktisch und führt die Alias-Namenssubstitution durch. Danach, vor Ausführung des Befehls, wird die Variablensubstitution durchgeführt. Das `$` (Dollarzeichen) ist das Schlüsselwort für die Substitution. Es wird allerdings unverändert übergeben, wenn ein Leerzeichen, Tabulatorzeichen oder Zeilenvorschubzeichen folgt. Steht dem Zeichen `$` ein `\` (umgekehrter Schrägstrich) voran, wird diese Erweiterung verhindert, mit Ausnahme von zwei Fällen:

- Der Befehl steht in `"` (Anführungszeichen). In diesem Fall führt die Shell immer die Substitution durch.
- Der Befehl steht in `'` (halben Anführungszeichen). In diesem Fall führt die Shell nie die Substitution durch. Zeichenfolgen in `'` werden für die Befehlssubstitution interpretiert. (Siehe Abschnitt "Befehlssubstitution in der C-Shell" auf Seite C-13.)

Die Shell erkennt die Ein- und Ausgabeadressierung vor der Variablenerweiterung und führt sie unabhängig voneinander durch. Andernfalls werden der Befehlsname und die vollständige Argumentenliste zusammen erweitert. Es ist demzufolge möglich, daß das erste (Befehls-) Wort mehr als ein Wort generiert, von denen das erste zum Befehlsnamen wird und die übrigen zu Parametern.

Wenn es nicht in `"` (Anführungszeichen) steht oder mit dem Änderungswert `:q` versehen ist, kann mit dem Ergebnis der Variablensubstitution möglicherweise noch die Befehls- und Dateinamenssubstitution durchgeführt werden. In Anführungszeichen gesetzt, wird eine Variable, die aus mehreren Wörtern besteht, zu einem einzelnen Wort oder Teil eines einzelnen Wortes erweitert, wobei die Wörter des Variablenwertes durch Leerzeichen voneinander getrennt sind. Wird der Änderungswert `:q` auf eine Substitution angewandt, wird die Variable auf mehrere Wörter erweitert. Jedes Wort ist durch ein Leerzeichen abgetrennt und in Anführungszeichen gesetzt, um eine spätere Befehls- und Dateinamenssubstitution zu verhindern.

Mit den folgenden Schreibweisen können Variablenwerte in die Shell-Eingabe eingeführt werden. Wenn nicht anders angegeben, ist es ein Fehler, auf eine Variable zu verweisen, die nicht mit dem Befehl `set` gesetzt wurde.

**`$Name`**  
**`${Name}`** Ersetzt die der Variablen *Name* zugeordneten Wörter, durch ein Leerzeichen voneinander getrennt. Durch geschweifte Klammern wird die Variable *Name* von nachfolgenden Zeichen abgetrennt, die ansonsten Teil von ihr wären. Shell-Variablenamen fangen mit einem Buchstaben an und bestehen aus bis zu 20 Buchstaben und Ziffern, einschließlich dem `_` (Unterstreichungszeichen). Gibt die Variable *Name* keine Shell-Variable an, sondern wurde sie in der Umgebung gesetzt, wird ihr Wert zurückgegeben. Änderungswerte mit führenden Doppelpunkten sowie andere hier beschriebene Formate sind in diesem Fall ungültig.

**`$Name[Selektor]`**  
**`${Name[Selektor]}`** Wählt nur einige der Wörter vom Wert der Variablen *Name* aus. Der Selektor ist von der Variablensubstitution betroffen und besteht möglicherweise aus einer einzelnen Zahl oder zwei Zahlen, die durch einen `-` (Bindestrich) getrennt sind. Das erste Wort des Zeichenfolgewertes einer Variablen ist mit 1 numeriert. Wird die erste Zahl in einem Bereich ausgelassen, wird als Standardwert 1 eingestellt. Wird die letzte Zahl in einem Bereich ausgelassen, wird als Standardwert  `$#Name` eingestellt. Mit dem Symbol `*` (Stern) werden alle Wörter ausgewählt. Es ergibt keinen Fehler, wenn ein Bereich leer ist, solange das zweite Argument ausgelassen wird oder sich in einem Bereich befindet.

**`$#Name`**  
 **`${#Name}`** Gibt die Anzahl der Wörter in der Variablen an. Diese Angabe ist nützlich bei einer späteren Verwendung in einem `[Selektor]`.

**`$0`** Ersetzt den Namen der Datei, aus der die Befehlseingabe gelesen wird. Ein Fehler tritt auf, wenn der Name nicht bekannt ist.

**`$Nummer`**  
 **`${Nummer}`** Äquivalent zu  `$argv[Nummer]`.  
 **`$*`** Äquivalent zu  `$argv[*]`.

Auf die obengenannten Substitutionen können die Änderungswerte `:gh`, `:gt`, `:gr`, `:h`, `:r`, `:q` und `:x` angewandt werden. Kommen im Befehlsformat `{ }` (geschweifte Klammern) vor, müssen die Änderungswerte innerhalb der geschweiften Klammern stehen. Mit der aktuellen Implementierung ist nur ein Änderungswert (mit führendem `:` (Doppelpunkt)) pro Variablenerweiterung zugelassen.

Die folgenden Substitutionen können möglicherweise durch Änderungswerte (mit `:`) nicht geändert werden:

**`$?Name`**  
 **`${?Name}`** Ersetzt die Zeichenfolge 1, wenn die Variable *Name* gesetzt ist; die Zeichenfolge 0 (Null), wenn diese Variable nicht gesetzt ist.

**`$?0`** Ersetzt 1, wenn der aktuelle Eingabedateiname bekannt ist; 0 (Null), wenn der Dateiname nicht bekannt ist.

**`$$`** Ersetzt die (dezimale) Prozeßnummer der Vater-Shell.

**`$<`** Ersetzt eine Zeile der Standardeingabe ohne weitere Interpretation. Diese Substitution verwenden, um in einer Shell-Prozedur die Eingabe von der Tastatur zu lesen.



---

## Vordefinierte Variablen und Umgebungsvariablen in der C-Shell

Bestimmte Variablen haben in der Shell eine Sonderbedeutung. Von diesen werden die Variablen **argv**, **cwd**, **home**, **path**, **prompt**, **shell** und **status** immer von der Shell eingestellt. Bis auf die Variablen **cwd** und **status** erfolgen die Einstellungen nur bei der Initialisierung. Sie werden beibehalten, wenn der Benutzer sie nicht explizit zurücksetzt.

Der Befehl **csch** kopiert die Umgebungsvariablen **USER**, **TERM**, **HOME** und **PATH** in die entsprechenden **csch**-Variablen **user**, **term**, **home** und **path**. Die Werte werden immer, wenn die normalen Shell-Variablen zurückgesetzt werden, in die Umgebung zurückkopiert. Der Benutzer braucht sich, außer, wenn es um die Datei **.cschrc** geht, nicht um die Einstellung der Variablen **path** zu kümmern, da die **csch**-Unterprozesse die Pfaddefinition von der Umgebung importieren und nach einer Änderung wieder exportieren.

Folgende Variablen haben eine Sonderbedeutung:

|                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>argv</b>      | Enthält die Argumente, die an Shell-Prozeduren übergeben wurden. Positionsparameter werden von dieser Variablen ersetzt.                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>cdpath</b>    | Gibt eine Liste mit alternativen Verzeichnissen an, die von dem Befehl <b>chdir</b> durchsucht werden sollen, um Unterverzeichnisse zu finden.                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>cwd</b>       | Gibt den vollständigen Pfadnamen des aktuellen Verzeichnisses an.                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>echo</b>      | (Gesetzt, wenn die Befehlszeilenoption <b>-x</b> verwendet wird.) Führt, falls gesetzt, dazu, daß jeder Befehl und seine Argumente kurz vor der Ausführung eine Meldung zurückgeben. Bei nicht integrierten Befehlen werden alle Erweiterungen vor der Rückmeldung durchgeführt. Integrierte Befehle werden vor der Befehls- und Dateinamensubstitution zurückgemeldet, da diese Substitutionen dann selektiv durchgeführt werden.                                                                                               |
| <b>histchars</b> | Gibt einen Zeichenfolgewart an, der die in der Protokollsubstitution verwendeten Zeichen ändern soll. Das erste Zeichen dieses Wertes wird als Protokollsubstitutionszeichen verwendet, es ersetzt das Standardzeichen ! (Ausrufezeichen). Das zweite Zeichen dieses Wertes ersetzt den ^ (Zirkumflex) in schnellen Substitutionen.<br><br><b>Hinweis:</b> Wird der Wert <b>histchars</b> auf ein Zeichen gesetzt, das in Befehl- oder Dateiname verwendet wird, kann es zu einer unbeabsichtigten Protokollsubstitution kommen. |
| <b>history</b>   | Enthält einen numerischen Wert zur Steuerung der Größe der Protokolliste. Jeder Befehl, auf den in dieser Anzahl von Ereignissen verwiesen wird, wird nicht gelöscht. Sehr große Werte für die Variable <b>history</b> können dazu führen, daß die Shell nicht mehr genügend Speicher besitzt. Unabhängig davon, ob diese Variable gesetzt ist, speichert sie den letzten Befehl, der mit der Protokolliste ausgeführt wurde.                                                                                                    |
| <b>home</b>      | Zeigt das Benutzerverzeichnis, von der Umgebung initialisiert. Die Dateinamenerweiterung der <b>~</b> (Tilde) verweist auf diese Variable.                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>ignoreeof</b> | Gibt an, daß die Shell ein Dateiendezeichen, das von einer Datenstation aus eingegeben wurde, ignoriert. Dadurch kann die Shell nicht unbeabsichtigt gelöscht werden, wenn sie ein Dateiendezeichen (Strg-D) liest.                                                                                                                                                                                                                                                                                                              |

- mail** Gibt die Dateien an, in denen die Shell nach elektronischer Post sucht. Das erfolgt nach jedem Befehlsabschluß und führt zu einer Eingabeaufforderung, wenn eine angegebene Zeitspanne vorüber ist. Die Shell zeigt die Nachricht `You have new mail an`, wenn die Datei existiert und die Zeit des letzten Zugriffs nicht später ist als die Änderungszeit.
- Ist das erste Wort des Wertes der Variablen **mail** numerisch, gibt es (in Sekunden) ein anderes Intervall für die Suche nach Post an; der Standardwert ist 10 Minuten. Werden mehrere Postdateien angegeben, zeigt die Shell die Nachricht `New mail in Datei an`, wenn sich elektronische Post in der angegebenen Datei befindet.
- noclobber** Legt der Ausgabeumadressierung Beschränkungen auf, um sicherzustellen, daß Dateien nicht unbeabsichtigt zerstört werden und daß sich >>-Umadressierungen auf vorhandene Dateien beziehen.
- noglob** Verhindert die Dateinamenerweiterung. Diese Variable ist äußerst nützlich in Shell-Prozeduren, die sich nicht mit Dateinamen befassen, oder wenn eine Liste mit Dateinamen erhalten wurde und eine weitere Erweiterung nicht erwünscht ist.
- nomatch** Gibt an, daß kein Fehler auftritt, wenn eine Dateinamenerweiterung mit keiner der vorhandenen Dateien übereinstimmt; stattdessen wird das Elementarmuster zurückgegeben. Es ist dennoch ein Fehler, wenn das Elementarmuster nicht einwandfrei ist.
- notify** Gibt an, daß die Shell asynchrone Hinweise von Änderungen im Jobstatus sendet. Mit dem Standardwert werden Statusänderungen kurz vor der Shell-Eingabeaufforderung angezeigt.
- path** Gibt die Verzeichnisse an, in denen Befehle für die Ausführung gesucht werden. Ein leeres Wort gibt das aktuelle Verzeichnis an. Ist keine Variable **path** gesetzt, werden nur vollständige Pfadnamen ausgeführt. Der normale Suchpfad ist das aktuelle Verzeichnis `/usr/bin`. Eine Shell, in der weder die Option `-c` noch die Option `-t` angegeben wurde, schreibt normalerweise den Inhalt der in der Variablen **path** angegebenen Verzeichnisse in eine interne Tabelle und zwar nach dem Lesen der Datei `.cshrc` und jedes Mal, wenn die Variable **path** zurückgesetzt wird. Werden diesen Verzeichnissen neue Befehle hinzugefügt, während die Shell aktiv ist, muß der Befehl **rehash** eingegeben werden, sonst werden die Befehle nicht gefunden.
- prompt** Gibt die Zeichenfolge an, die angezeigt wird, bevor einzelne Befehle von einer interaktiven Datenstationseingabe gelesen werden. Erscheint ein ! (Ausrufezeichen) in der Zeichenfolge, wird es durch die aktuelle Ereignisnummer ersetzt. Befindet sich das ! innerhalb einer Zeichenfolge in Anführungszeichen (oder halben Anführungszeichen), muß dem ! ein \ (umgekehrter Schrägstrich) voranstehen. Die Standardeingabeaufforderung für Benutzer mit Root-Berechtigung ist `%`. Die Standardeingabeaufforderung für den Root ist `#`.
- savehist** Gibt einen numerischen Wert an, um die Anzahl der Einträge in der Protokolliste zu steuern, die beim Abmelden in der Datei `~/.history` gespeichert werden. Jeder Befehl, auf den in dieser Anzahl an Ereignissen verwiesen wird, wird gespeichert. Während des Systemanlaufs liest die Shell die Datei `~/.history` in die Protokolliste ein, womit das Protokoll über die Anmeldung hinweg gespeichert ist. Durch sehr große Werte für die Variable **savehist** wird der Shell-Systemanlauf verlangsamt.

- shell** Gibt die Datei an, in der sich die C-Shell befindet. Wird verwendet bei der Aufspaltung von Shells, um Dateien zu interpretieren, in denen Ausführungsbit gesetzt sind, die aber vom System nicht ausgeführt werden können. Der Wert der Variablen **shell** wird auf den Pfadnamen der C-Shell gesetzt (normalerweise /bin/csh).
- status** Gibt den Status an, der von dem letzten Befehl zurückgegeben wurde. Wird der Befehl abnormal beendet, wird zu der Statusangabe 0200 hinzugefügt. Bei integrierten Befehlen, die nicht erfolgreich ausgeführt wurden, wird der Ausführungsstatus 1 zurückgegeben. Bei allen anderen integrierten Befehlen wird der Status auf 0 (Null) gesetzt.
- time** Steuert die automatische Ablaufsteuerung von Befehlen. Ist diese Variable gesetzt, wird jeder Befehl, der mehr als die angegebene Zahl an CPU-Sekunden benötigt, nach seiner Ausführung neun Standardausgaben anzeigen, die von verschiedenen Befehls- und Textzeichenfolgen generiert wurden. Weitere Informationen über Standardausgaben enthält der Abschnitt zu dem integrierten Befehl **time**.
- verbose** Diese Variable wird durch die Befehlszeilenoption **-v** gesetzt und führt dazu, daß die Wörter in jedem Befehl nach der Protokollsubstitution angezeigt werden.

---

## Dateinamensubstitution in der C-Shell

Die C-Shell verfügt über verschiedene Verkürzungsfunktionen, die Zeit und Tastenanschläge sparen. Enthält ein Wort eines der folgenden Zeichen: \* (Stern), ? (Fragezeichen), [] (eckige Klammern) oder { } (geschweifte Klammern) oder beginnt ein Wort mit einer ~ (Tilde), wird mit diesem Wort die Dateinamensubstitution durchgeführt. Die C-Shell betrachtet das Wort als Muster und ersetzt es mit einer alphabetisch sortierten Liste von Dateinamen, die mit dem Muster übereinstimmen.

Die aktuelle Sortierfolge wird verwendet, wie in der Umgebungsvariablen **LC\_COLLATE** oder **LANG** angegeben. Bei einer Liste von Wörtern, die die Dateinamensubstitution angeben, tritt ein Fehler auf, wenn kein Muster mit einem vorhandenen Dateinamen übereinstimmt. Es ist jedoch nicht erforderlich, daß für alle Muster eine Übereinstimmung gefunden wird. Nur die zeichenabgleichenden Symbole \* (Stern), ? (Fragezeichen) und [ ] (eckige Klammern) weisen auf die Verwendung von Platzhalterzeichen oder die Dateinamenerweiterung hin. Die ~ (Tilde) und { } (geschweifte Klammern) weisen auf die Dateinamenabkürzung hin.

### Dateinamenerweiterung

Der \* (Stern) entspricht jeder Zeichenfolge, einschließlich der leeren Zeichenfolge. Zum Beispiel werden in einem Verzeichnis mit den Dateien:

```
a aa aax alice b bb c cc
```

mit dem Befehl `print a*` alle Dateinamen ausgegeben, die mit dem Zeichen `a` beginnen:

```
a aa aax alice
```

**Hinweis:** Beim Abgleich der Dateinamen müssen die Zeichen `.` (Punkt) und `/` (Schrägstrich) explizit übereinstimmen.

Das ? (Fragezeichen) entspricht jedem einzelnen Zeichen. Der Befehl:

```
ls a?x
```

listet alle Dateinamen auf, die mit dem Buchstaben `a` beginnen, gefolgt von einem einzelnen Zeichen, und mit dem Buchstaben `x` enden:

```
aax
```

Um ein einzelnes Zeichen oder einen Zeichenbereich abzugleichen, das oder die Zeichen in [ ] (eckige Klammern) einschließen. Der Befehl:

```
ls [abc]
```

listet alle Dateinamen auf, die mit einem der eingeschlossenen Zeichen exakt übereinstimmen:

```
a b c
```

In eckigen Klammern ist ein lexikalischer Zeichenbereich durch `[a-z]` gekennzeichnet. Die Zeichen, die mit diesem Muster übereinstimmen, sind durch die aktuelle Sortierfolge definiert.

## Dateinamenabkürzung

Die ~ (Tilde) und { (linke geschweifte Klammer) weisen auf die Dateinamenabkürzung hin. Eine ~ am Anfang eines Dateinamens wird verwendet, um Benutzerverzeichnisse anzuzeigen. Steht es allein, wird das Zeichen ~ zum Benutzerverzeichnis erweitert, wie im Wert der Shell-Variablen **home** definiert. Zum Beispiel listet der Befehl:

```
ls ~
```

alle Dateien und Verzeichnis im Benutzerverzeichnis **\$HOME** auf.

Folgt auf die Tilde ein Name, der aus Buchstaben, Ziffern und - (Gedankenstrich) besteht, sucht die Shell nach einem Benutzer mit diesem Namen und ersetzt das Benutzerverzeichnis **\$HOME**.

**Hinweis:** Folgt auf die ~ ein anderes Zeichen als ein Buchstabe oder ein / (Schrägstrich), oder erscheint eine Tilde woanders als am Anfang eines Wortes, wird sie nicht erweitert.

Um Zeichen in Dateinamen abzugleichen, ohne den vollständigen Dateinamen einzugeben, { } (geschweifte Klammern) um die Dateinamen verwenden. Das Muster a{b,c,d}e ist die Kurzform für abe ace ade. Die Shell hält an der Reihenfolge von links nach rechts fest, und speichert deswegen die Ergebnisse von Abgleichen separat auf einer niedrigen Stufe. Der Aufbau kann verschachtelt werden. Daher wird:

```
~source/sl/{oldls,ls}.c
```

zu:

```
/usr/source/sl/oldls.c /usr/source/sl/ls.c
```

erweitert, wenn /usr/source das Benutzerverzeichnis für source ist. Ähnlich könnte:

```
../{memo,*box}
```

erweitert werden zu:

```
../memo ../box ../mbox
```

## Zeichenklassen

Um Dateinamen in einem Bereich abzugleichen, können auch Zeichenklassen verwendet werden:

```
[:Zeichenklasse:]
```

Mit diesem Format gleicht das System jedes einzelne Zeichen, das zu der angegebenen Klasse gehört, ab. Die definierten Klassen entsprechen den Unterroutinen **ctype**.

### Zeichenklasse Definition

|               |                                                                                                                                      |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------|
| <b>alnum</b>  | alphanumerische Zeichen                                                                                                              |
| <b>alpha</b>  | Groß- und Kleinbuchstaben                                                                                                            |
| <b>digit</b>  | Ziffern                                                                                                                              |
| <b>lower</b>  | Kleinbuchstaben                                                                                                                      |
| <b>print</b>  | druckbare Zeichen                                                                                                                    |
| <b>punct</b>  | Interpunktionszeichen                                                                                                                |
| <b>space</b>  | Leerzeichen, Tabulatorzeichen, Zeilenschaltungszeichen, Zeilenvorschubzeichen, Vertikaltabulatorzeichen oder Formularvorschubzeichen |
| <b>upper</b>  | Großbuchstaben                                                                                                                       |
| <b>xdigit</b> | Hexadezimalzahlen                                                                                                                    |

Angenommen, der Benutzer befindet sich in einem Verzeichnis, das die folgenden Dateien enthält:

```
a aa aax Alice b bb c cc
```

Nach der Eingabe des folgenden Befehls an der C-Shell-Eingabeaufforderung:

```
ls [:lower:]
```

listet die C-Shell alle Dateinamen auf, die mit einem Kleinbuchstaben beginnen:

```
a aa aax b bb c cc
```

Weitere Informationen über Ausdrücke mit Zeichenklassen enthält der Abschnitt über den Befehl `ed`.

---

## Ein- und Ausgabe in der C-Shell umleiten

Bevor die C-Shell einen Befehl ausführt, sucht sie in der Befehlszeile nach Umadräsentationszeichen. Diese Sonderzeichen weisen die Shell an, die Ein- und Ausgabe umzuleiten.

Mit den folgenden Syntaxanweisungen kann die Standardein- und -ausgabe eines Befehls umgeleitet werden:

- < Datei** Öffnet die angegebene Datei (bei der zuerst Variable, Befehl und Dateiname erweitert werden) als Standardeingabe.
- << Wort** Liest die Shell-Eingabe bis zu der Zeile, die mit dem Wert der Variablen *Wort* übereinstimmt. Bei der Variablen *Wort* werden Variable, Dateiname oder Befehl nicht erweitert. Jede Eingabezeile wird mit der Variablen *Wort* verglichen, bevor Substitutionen durchgeführt werden. Wenn kein Quotierungszeichen (\, ", ' oder '.') in der Variablen *Wort* auftritt, führt die Shell Variablen- und Befehlssubstitution für die dazwischenliegenden Zeilen aus, wobei der \ (umgekehrter Schrägstrich) als Quotierungszeichen für \$ (Dollarzeichen), \ (umgekehrter Schrägstrich) und ' (Gravis) zugelassen ist. Ersetzte Befehle verfügen noch über alle Leerzeichen, Tabulatorzeichen und Zeilenvorschubzeichen; lediglich das letzte Zeilenvorschubzeichen wird gelöscht. Der resultierende Text wird in eine namenlose temporäre Datei gestellt, die dem Befehl als Standardeingabe übergeben wird.
- > Datei**  
**>! Datei**  
**>& Datei**  
**>&! Datei** Verwendet die angegebene Datei als Standardausgabe. Existiert die Datei noch nicht, wird sie von der Shell erstellt. Existiert die Datei bereits, wird sie abgeschnitten und der ursprüngliche Inhalt geht verloren. Ist die Shell-Variablen **noclobber** gesetzt, darf die Datei nicht existieren oder es muß sich um eine zeichenorientierte Gerätedatei handeln oder ein Fehler tritt auf. Damit wird verhindert, daß Dateien unbeabsichtigt zerstört werden. In diesem Fall die Formate mit ! (Ausrufezeichen) verwenden, um diese Prüfung zu umgehen. Die Formate mit & (Et-Zeichen) leiten die Diagnoseausgabe in die angegebene Datei weiter sowie an die Standardausgabe. Die Variable *Datei* wird ebenso erweitert wie < Eingabedateinamen.
- >> Datei**  
**>>& Datei**  
**>>! Datei**  
**>>&! Datei** Verwendet wie > die angegebene Datei als Standardausgabe, stellt aber die Ausgabe an das Ende der Datei. Ist die Shell-Variablen **noclobber** gesetzt, tritt ein Fehler auf, wenn die Datei nicht existiert, es sie denn, eines der Formate mit ! (Ausrufezeichen) wurde angegeben. Ansonsten ähnelt diese Anweisung der Anweisung >.

Ein Befehl übernimmt die Umgebung, in der die Shell aufgerufen wurde, so wie sie von Ein-/Ausgabeparametern und der Anwesenheit des Befehls als Pipeline verändert wurde. Deswegen haben, anders als bei einigen vorherigen Shells, Befehle, die von einer Datei mit Shell-Befehlen aus ausgeführt werden, standardmäßig keinen Zugriff auf den Text der Befehle. Stattdessen empfangen sie die Originalstandardeingabe der Shell. << verwenden, um Daten der Befehlszeile auszugeben. Damit arbeiten Shell-Befehlsdateien wie Pipelinekomponenten und der Shell-Block liest die Eingabe. Zu beachten ist, daß die

Standardeingabe für einen Befehl, der abgehängt ausgeführt wird, in der leeren Datei **/dev/null** nicht geändert wird. Stattdessen bleibt die Originalstandardeingabe der Shell der Standard.

Um die Diagnoseausgabe durch eine Pipe mit der Standardausgabe umzuleiten, das Format **|&** (senkrechter Balken, Et-Zeichen) statt **|** (senkrechter Balken) verwenden.

## Steuerungsfluß

Die Shell enthält Befehle, mit denen der Steuerungsfluß in Befehlsdateien (Shell-Prozeduren) und (begrenzt, aber nützlich) von der Datenstationseingabe reguliert werden kann. Diese Befehle führen die Shell dazu, ihre Eingabe erneut zu lesen oder zu überspringen und, wegen der Implementierung, die Plazierung von einigen Befehlen zu beschränken.

Die Anweisungen **foreach**, **switch** und **while** sowie das Format **if-then-else** jeder Anweisung **if** machen es erforderlich, daß die Hauptschlüsselwörter in einem einzelnen einfachen Befehl auf einer Eingabezeile erscheinen.

Kann eine Shell-Eingabe nicht durchsucht werden, stellt die Shell die Eingabe, wenn eine Schleife gelesen wird, in den Puffer und durchsucht den internen Puffer, wenn von der Schleife aus ein erneutes Lesen erforderlich ist. In dem hier möglichen Ausmaß sind rückwärtige **goto**-Befehle bei Eingaben, die nicht durchsucht werden können, erfolgreich.



---

## Jobsteuerung in der C-Shell

Die Shell ordnet jedem Prozeß eine Jobnummer zu. Sie führt eine Tabelle der aktuellen Jobs und ordnet ihnen kleine, ganze Zahlen zu. Wird ein Job asynchron gestartet, indem der Befehl mit einem & (Et-Zeichen) abgeschlossen wird, zeigt die Shell eine Zeile, ähnlich der folgenden an:

```
[1] 1234
```

Diese Zeile zeigt, daß die Jobnummer 1 ist und daß der Job aus einem einzelnen Prozeß mit der Prozeß-ID 1234 besteht. Den integrierten Befehl **jobs** verwenden, um eine Tabelle der aktuellen Jobs aufzurufen.

Ein Job, der im Hintergrund ausgeführt wird, konkurriert um die Eingabe, wenn er versucht, von der Datenstation zu lesen. Hintergrundjobs können auch Ausgaben erzeugen, die um die Datenstation konkurrieren und dort mit der Ausgabe von anderen Jobs verzahnt werden.

Auf Jobs in der Shell kann auf mehrere Arten verwiesen werden. Mit dem % (Prozentzeichen) wird der Jobname eingeführt. Dabei kann es sich entweder um die Jobnummer handeln oder um den Befehlsnamen, der den Job gestartet hat, wenn dieser Name eindeutig ist. Wird beispielsweise ein Prozeß **make** als Job 1 ausgeführt, kann auf ihn mit %1 verwiesen werden. Gibt es nur einen ausgesetzten Job, dessen Name mit der Zeichenfolge **make** beginnt, kann auch mit %**make** darauf verwiesen werden. Ebenso kann:

*%?Zeichenfolge*

verwendet werden, um einen Job anzugeben, dessen Name die Variable Zeichenfolge enthält, wenn nur ein solcher Job vorhanden ist.

Die Shell stellt sofort fest, wenn ein Prozeß seinen Status ändert. Wird ein Job blockiert, so daß keine weitere Ausführung möglich ist, sendet die Shell eine Nachricht an die Datenstation. Diese Nachricht wird erst nach Drücken der Eingabetaste angezeigt. Ist jedoch die Shell-Variable **notify** gesetzt, gibt die Shell sofort eine Nachricht aus, die die Statusänderungen von Hintergrundjobs anzeigt. Den integrierten Befehl **notify** verwenden, um einen Einzelprozeß zu markieren, so daß dessen Statusänderungen sofort gemeldet werden. Standardmäßig markiert der Befehl **notify** den aktuellen Prozeß.

## Empfohlene Dokumentation

### Zugehörige Informationen

Die Abschnitte zu den Befehlen **cs**h und **ed**.

Die Abschnitte zu den integrierten C-Shell-Befehlen **alias**, **jobs**, **notify**, **set** und **unalias**.



# Index

## Zahlen

1/4 Zoll Band, Arbeitsoberflächenmenü, 5-28  
8-mm-Band, Arbeitsoberflächenmenü, 5-28

## A

- Ablezen der dreistelligen Fehlernummer, 1-3
- Abmelden, Vorgehensweise, 1-10
- Abreißknopf, 4-28
- Abreißmenü, 4-28
- Add\_Program, Arbeitsoberfläche, 5-3
- Admin Desktop, 5-51
  - sichtbare Dateien, 5-55
  - Symbole, 5-52
- Admin, Verzeichnis, Symbole, 5-55
- Adressen
  - Netz, allgemein, 6-4
  - Post, 7-3-7-4
- AIX-Editoren
  - edit, 3
  - ex, 3
- AIXwindows Desktop
  - Siehe auch* Desktop
  - Arbeitsoberflächentypen, 5-38
- AIXwindows Window Manager (MWM). *Siehe* Window Manager und MWM
- Aktionsfenstermenüs, 4-24
- Aktives Fenster, Beschreibung, 4-13
- Aktivieren von Symbolen, Arbeitsoberfläche, 5-20
- Alias-Namenprüfung, Befehle, Korn-Shell, A-26
- Alias-Namenssubstitution, C-Shell, C-18
- Alias-Namen für Post
  - Liste anzeigen, 7-51
  - Liste erstellen, 7-50
- Alle abwählen (Option), Arbeitsoberflächenmenü, 5-27
- Alle abwählen (Option), Arbeitsoberflächenverzeichnismenü Editieren (Menü), 5-31
- Alle auswählen (Option), Arbeitsoberflächenmenü, 5-27
- Alle auswählen (Option), Arbeitsoberflächenverzeichnismenü, 5-31
- Alle Verzeichnisse schließen (Option), Arbeitsoberflächenmenü Datei, 5-27
- Alternativdrucker, Arbeitsoberflächenmenü, 5-28
- Anmeldedateien
  - Datei .profile, 2-31
  - Datei /etc/profile, 2-31
- Anmelden
  - an mehreren Systemen, 2-33
  - Beispiel für Anmeldedialog, 1-3
  - den Namen anzeigen, mit dem Befehl whoami, 3-57
  - derzeit angemeldete Systembenutzer anzeigen, 3-57
  - ferner Host, Befehl ftp, 6-16
  - mehrfach am selben System, 2-33
  - Vorgehensweise, 1-4
  - Anmeldung, Zweck, 2-33
- Anpassen, Arbeitsoberfläche, Hilfsprogramme, 5-3
- Anwendungssymbole, Arbeitsoberfläche
  - ausführen, 5-20
  - Beschreibung, 5-13
- Anzeige, Anzeige in eine Datei kopieren, 3-20
- Anzeigekomponenten, Arbeitsoberfläche, 5-2
- Anzeigekomponenten, Desktop, 5-7
- Anzeigen, Dateien, 3-19
- anzeigen
  - Dateien, angegebene Abschnitte, 3-24
  - Dateien, teilweise, 3-24
  - Text seitenweise anzeigen, 3-19-3-25
- Anzeigen (Menü), Arbeitsoberfläche
  - Bereinigen (Option), 5-28
  - Neu anordnen (Option), 5-28
  - Optionen, 5-28
- Anzeigen (Menü), Arbeitsoberflächenverzeichnis
  - Details (Option), 5-32
  - Namen (Option), 5-32
  - Optionen, 5-32
  - Sortieren (Option), 5-32
    - Nach Zeit, 5-32
    - Nach Größe, 5-32
    - Nach Name, 5-32
    - Nach Verzeichnis, 5-32
  - Symbole (Option), 5-32
- Anzeigen-Server, Definition, 4-3
- Anzeigeoptionen bei Nachrichten, 7-19-7-22
- Arbeitsdialog, 4-31
- Arbeitsoberfläche
  - Anzeigekomponenten, 5-2, 5-7
  - Arbeitsoberflächenfenster, 5-7
  - Arbeitsoberflächentypen, 5-2
  - automatisch starten, 5-6
  - Dialogfenster, 5-8
  - Einführung, 5-1
  - Interaktion mit dem Window Manager, 4-4
  - interaktive Hilfsprogramme zum Anpassen, 5-3
  - Maus, 5-9-5-11
  - Mauszeiger, 5-8
  - mehrere Arbeitsoberflächen, 5-2
  - Menüleiste, 5-8
  - Menüoptionen und Tastenäquivalente, 5-34
  - Menüs, 5-24
  - Merkmale, 5-2
  - starten, 5-5
  - Starten in der Anmelde-Shell, 5-5
  - Starten mit Befehl, 5-6
  - Symbole, 5-2, 5-8, 5-12
  - Symbolmenüs, 5-2
  - Verwaltung von Arbeitsoberflächen und Dateien, 5-2
- Arbeitsoberfläche - Grundlagen, Arbeitsoberfläche starten, 5-5

- Arbeitsoberflächen- und Dateiverwaltung, 5-2
- Arbeitsoberflächenfenster, 5-7
- Arbeitsoberflächenmenü, *Siehe*, 5-29
- Arbeitsoberflächenprogrammierung, 5-3
- Arbeitsoberflächensymbole, Arbeitsoberfläche, Beschreibung, 5-14
- Arbeitsoberflächentypen, 5-2, 5-38
  - admin, 5-51
  - Power Desktop, 5-45
- Arbeitsoberflächenverzeichnis (Menüoptionen), 5-30
- ASCII-Schnittstelle
  - erste Schritte, 8-14
  - neue Funktionen, 8-9
- Aufgabenindex, Einführung, 8-5
- Ausführen von Anwendungssymbolen, Arbeitsoberfläche, 5-20
- Ausführen von Programmsymbolen, Arbeitsoberfläche, 5-20
- Ausführungsberechtigung, Arbeitsoberfläche, 5-16
- Ausführungsstatus, Korn-Shell, A-3
- Auswahl, Text, 4-37
- Auswahldialog, 4-30
- Auswahlmenü, 4-28

## B

- backup, Befehl, 3-44–3-45
  - Funktion, 2-4
- Basisfenster, 4-6
- Bedingte Substitution, Bourne-Shell, B-17
- Befehl cc, Arbeitsoberflächenmenü, 5-29
- Befehl ps, Arbeitsoberflächenmenü, 5-29
- Befehl talk, Echtzeitdialog, 6-21
- Befehle
  - abbrechen, 2-45
  - Arbeiten mit Dateien, 3-5
  - ausführen, 2-44
  - backup, 3-44
  - Bourne-Shell, Verwendung, B-4
  - C-Shell
    - integrierte, C-3
    - Verwendung, C-2
  - cal, 3-61
  - cat, 1-8, 3-19, 3-26
  - cd, 3-32
  - chown, 3-36
  - clear, 3-62
  - colrm, 3-28
  - compress, 3-28
  - cp, 3-17, 3-33
  - cut, 3-26
  - del, 3-16
  - df, 3-25
  - dircmp, 3-37
  - eingeben, 1-7, 2-44–2-45
  - env, 3-59
  - file, 3-24
  - find, 3-21
  - grep, 3-22
  - head, 3-24
  - id, 3-57
  - Korn-Shell, zusammengesetzt, A-7

- li, 3-33
- Liste, Einführung, 8-6
- ls, 3-34
- mkdir, 3-31
- more, 3-19
- mv, 3-17, 3-19, 3-31
- nl, 3-28
- pack, 3-29
- passwd, 1-6, 3-59
- paste, 3-27
- pg, 3-19
- Pipes, Beschreibung, 2-41
- rm, 3-17
- rmdir, 3-36
- Shell-Prozeduren, Beschreibung, 2-41
- su, 3-59
- Syntaxdiagramme, Erklärung, 2-41
- tee, 3-20
- uname, 3-58
- wc, 3-23
- who, 1-8, 3-57
- whoami, 1-7, 3-57
- write, 3-60
  - zum Arbeiten mit Verzeichnissen, 3-11
- Befehls-Alias-Namenprüfung, Korn-Shell, A-26
  - Tildensubstitution, A-27
- Befehlsdialog, 4-30
- Befehloptionen für ... (Option),
  - Arbeitsoberflächenmenü, 5-29
  - , 5-29
- Befehloptionen für ... (Option),
  - Arbeitsoberflächenmenü Optionen,
  - Namensanzeige, 5-29
- Befehlsprotokoll, Korn-Shell, A-22
- Befehlssubstitution
  - Bourne-Shell, B-12
  - C-Shell, C-13
  - Korn-Shell, A-24
- Befehlszeileneditierung, Korn-Shell, A-38
  - Editiermodus emacs, A-38
  - Editiermodus vi, A-41
- Befehlszeilenschnittstelle, Beschreibung, 1-4
- Benutzer
  - Ändern eines Kennwortes, 1-6–1-10
  - Benutzer-ID ändern, mit dem Befehl su, 3-59–3-62
  - Kennwort für einen Benutzer ändern, mit dem Befehl passwd, 3-59–3-62
  - System-ID anzeigen, 3-57–3-62
- Benutzerumgebungen, anpassen, mit Hilfe von Profilen, 2-31
- Benutzerverzeichnissymbole, Arbeitsoberfläche, Beschreibung, 5-13
- Benutzung von Menüs, 4-24
- Berechtigungen. *Siehe* Zugriffsmodi
- Bereinigen (Option), Arbeitsoberflächenmenü, 5-28
- Betriebssystem
  - aktuellen Namen anzeigen, 3-58–3-62
  - allgemeine Aufgaben, Beispiele, 3-56
  - Tastensequenzen, 2-38
- Betriebssysteme, andere, 6-6
- Bildschirm

- einschalten, 1-3
- löschen, 3-62
- Blätterfenster, 4-33
- Blättern
  - Arbeitsoberflächenmenü, 5-29
  - Text, 8-29
- BNU (Netz-Basisdienstprogramm), Post
  - Adressieren, 7-7-7-8
  - Beschreibung, 7-7
- Bourne-Shell
  - Aufruf, B-1
  - bedingte Substitution, B-17
  - Befehle
    - integrierte, B-7
    - Verwendung, B-4
  - Befehlssubstitution, B-12
  - Dateinamensubstitution, B-19
  - Diagnoseausgabe, B-22
  - Ein- und Ausgabe umleiten, B-21
    - Verwendung von Dateideskriptoren, B-23
  - Platzhalterzeichen, B-19
  - Positionsparameter, B-18
  - Quotierungszeichen, B-5
  - Signalverarbeitung, B-5
  - Sonderbefehle, B-7
  - Standardein- und -ausgabe, B-21
  - Umgebung, B-1
  - Variablen, B-14
    - Substitution, B-13
    - benutzerdefiniert, B-13
    - vordefinierte Sondervariablen, B-16
- Brille auf Symbol, Arbeitsoberfläche, 5-15
- Brücken, Netz, 6-5
- Bücherliste, Einführung, 8-7
- Integrierte Befehle, Bourne-Shell, B-7

## C

- C-Shell
  - Alias-Namenssubstitution, C-18
  - Aufruf, C-1
  - Befehle
    - integriert, C-3
    - Verwendung, C-2
  - Befehlssubstitution, C-13
  - Beschränkungen, C-1
  - Ein- und Ausgabe umleiten, C-27
  - Dateinamensubstitution, C-24
  - Jobsteuerung, C-29
  - Protokollsubstitution, C-14
  - Signalverarbeitung, C-1
  - Variablensubstitution, C-19
    - vordefinierte und Umgebungsvariablen, C-21
- cal, Befehl, 3-61
- cat, Befehl, 3-19
- cd, Befehl, 3-32
- chfs, Befehl, Funktion, 2-3
- chown, Befehl, 3-36
- clear, Befehl, 3-62
- Klienten-Server-Modell, Definition, 4-3
- colrm, Befehl, 3-28-3-30
- compress, Befehl, 3-28
- cp, Befehl, 3-17, 3-33
- cpio, Befehl, Funktion, 3-40, 3-41

- crfs, Befehl, Funktion, 2-3

## D

- Dämonprozesse, Beschreibung, 2-14
- Datei
  - Eignerstatus, 3-12
    - nach einem Muster suchen, 3-22-3-25
  - Verknüpfen von Zeilen in einer Datei, mit dem Befehl paste, 3-27-3-30
  - Zugriffsmodus, festlegen, 3-12
- Datei .profile, 2-31
- Datei (Menü), Arbeitsoberfläche, Verzeichnis, Öffnen (Option), 5-30
- Datei (Menü), Arbeitsoberflächenverzeichnis
  - Datei editieren (Option), 5-30
  - Duplizieren (Option), 5-31
  - Kopieren in... (Option), 5-31
  - Löschen (Option), 5-31
  - Merkmale... (Option), 5-31
  - Neue Datei (Option), 5-30
  - Neues Verzeichnis (Option), 5-30
  - Optionen, 5-30
  - Umbenennen (Option), 5-31
  - Verknüpfen mit... (Option), 5-31
  - Verzeichnis schließen (Option), 5-31
- Datei /etc/profile, 2-31
- Datei dead.letter, 7-14
- Datei editieren (Option),
  - Arbeitsoberflächenverzeichnismenü, 5-30
- Datei mbox, 7-14
- Dateiauswahldialog, 4-30
- Dateibaum, Beschreibung, 2-2
- Dateideskriptor, Erklärung, 2-29
- Dateien
  - ändern, von einer verbundenen Datei aus, 3-12
  - anzeigen, 3-5
  - ASCII-, 3-4
  - ausführbare, 3-4
  - ausgewählte Felder ausschneiden, mit dem Befehl cut, 3-26-3-30
  - Befehle zum Arbeiten mit, 3-5
  - benennen, 3-4
  - Berechtigungen, 3-4
  - binäre, 3-4
  - Definition, 3-2, 3-3
  - DOS-Dateien löschen, 3-39
  - DOS-Dateien umsetzen, 3-7
  - editieren, 3-5
  - Eignerstatus, 3-13
  - Eignerstatus ändern, 3-36
  - einfügen, 3-27
  - entfernen, 3-16-3-25
  - Erklärung der Deskriptoren, 2-29
  - erstellen, 1-8, 3-5
  - erweitern, 3-30
  - formatieren für die Anzeige, 3-19-3-25
  - in die Ausgabe schreiben, von einer angegebenen Stelle aus, 3-24-3-25
  - Inhalt anzeigen, 3-19
  - kombinieren, 3-5
  - komprimieren, 3-29
  - komprimierte Dateien entpacken, 3-30

- Komprimierung, 3-28
- kopieren, 3-6, 3-17-3-25
  - aus DOS, 3-38-3-39
- kopieren in DOS, 3-38-3-39
- löschen, 3-6, 3-16-3-25
- nach übereinstimmenden Ausdrücken
  - suchen, 3-21-3-25
- packen, 3-29
- Pfadnamen, 3-9
- reguläre, 3-4
- sichern, 3-7, 3-44
- Spalten löschen, 3-28
- Teil des Inhalts anzeigen, 3-24
- Text bearbeiten, 3-26-3-30
- Typ anzeigen, 3-24-3-25
- Typen, 3-4
- Überblick, 3-3
- umbenennen, 3-6, 3-19
- verbinden, 3-6, 3-11
- vergleichen, 3-37
- verknüpfen, 3-26-3-30
- Verknüpfen der Zeilen mehrerer Dateien, mit dem Befehl paste, 3-27-3-30
- verschieben, 3-6, 3-17-3-25
- Verwaltungsvorgänge, 3-16-3-25
- zählen
  - Wörter, 3-23
  - Zeichen, 3-23
  - Zeilen, 3-23
- Zeilen numerieren, 3-28-3-30
- Zugriffsmodi. *Siehe*
  - zurückschreiben, 3-7
- Dateiindex, 3-4, 3-8
- Dateiindexnummern, 3-8
- Dateimenü, Arbeitsoberfläche
  - Alle Verzeichnisse schließen (Option), 5-27
  - Desktop laden... (Option), 5-26
  - Desktop neu starten (Option), 5-27
  - Desktop sichern als... (Option), 5-26
  - Desktop wechseln (Option), 5-27
  - Dieses Desktop schließen (Option), 5-27
  - Neues Desktop... (Option), 5-27
  - Option Öffnen, 5-26
  - Optionen, 5-26
  - Verzeichnis öffnen... (Option), 5-26
- Dateien, verbinden, 3-26
- Dateinamenssubstitution
  - Bourne-Shell, B-19
  - Korn-Shell, A-33
- Dateisymbole öffnen, Arbeitsoberfläche, 5-20
- Dateisymbole, Arbeitsoberfläche
  - Beschreibung, 5-14
  - editieren, 5-20
  - öffnen, 5-20
- Dateisystem
  - Aufzeichnungstechniken, Beschreibung, 2-2
  - erstellen, 2-2
  - Überblick, 3-2
  - verfügbaren Speicherbereich anzeigen, 3-25
  - Verzeichnisstruktur, erstellen, 2-2
- Dateisysteme
  - Diagramm, 3-2
  - Einführung in das Dateisystem /usr, 2-8
  - Einführung in das Dateisystem /var, 2-11
  - Einführung in das Stammdateisystem, 2-6
  - Einführung in das Verzeichnis /export, 2-12
  - Einführung in das Verzeichnis /usr/share, 2-10
  - Einführung in den reorganisierten Dateibaum, 2-5
  - Überblick über Dateisysteme, 2-2
- Dateneneinrichtung. *Siehe* DTE
- Datenfernverarbeitung
  - Funktionen, 6-6
  - Unterstützung für Netz, 6-3
- Datenstationen, Anzeige löschen, 3-62
- Datenträgerzugriffssteuerung. *Siehe* MAC
- Datenübertragungseinrichtung. *Siehe* DCE
- DCE (Datenübertragungseinrichtung), 6-3
- dd, Befehl, Funktion, 2-4, 3-40
- Definieren von Benutzern, 5-57
- dehnbarer Rahmen, Symbole, Arbeitsoberfläche, 5-18
- del, Befehl, 3-16
- Desktop einrichten, Symbol, neue Benutzer definieren, 5-57
- Desktop laden... (Option),
  - Arbeitsoberflächenmenü, 5-26
- Desktop neu starten (Option),
  - Arbeitsoberflächenmenü, 5-27
- Desktop sichern als... (Option),
  - Arbeitsoberflächenmenü Datei, 5-26
- Desktop-Überblick (Option),
  - Arbeitsoberflächenmenü Hilfe, 5-29
- Details (Option),
  - Arbeitsoberflächenverzeichnismenü, 5-32
- df, Befehl, 3-25
  - Funktion, 2-4
- Dialogfenster
  - Arbeitsdialog, 4-31
  - Auswahldialog, 4-30
  - Befehlsdialog, 4-30
  - Dateiauswahldialog, 4-30
  - Eingabedialog, 4-30
  - Fehlerdialog, 4-31
  - Fragedialog, 4-31
  - Informationsdialog, 4-31
  - Modaldialogfenster, 4-29
  - Nachrichtendialog, 4-30
  - nichtmodales, 4-29
  - Warnungsdialog, 4-31
- Dialogfenster, Arbeitsoberfläche, 5-8
- Dialogfenstermenüs, 4-27
- Dieses Desktop schließen (Option),
  - Arbeitsoberflächenmenü, 5-27
- dircmp, Befehl, 3-37
- Direktaufrufe, 4-16
- Direktaufrufe, Arbeitsoberfläche, 5-34
  - Menü auswählen, 5-34
- Direktaufruftasten, 4-26
- Disketten, formatieren
  - Befehl fdformat, 3-56-3-62
  - Befehl format, 3-56-3-62
- Dokumentsymbole, Arbeitsoberfläche
  - Beschreibung, 5-14
  - editieren, 5-20
  - öffnen, 5-20
- Domänen

Netz, allgemein, 6-4  
 Post adressieren, 7-6  
 DOS-Dateien  
   arbeiten mit, Überblick, 3-7  
   Kopieren in DOS-Dateien, 3-38  
   löschen, 3-39  
   umwandeln, 3-38  
   Verzeichnis auflisten, 3-39  
 drag and drop, starting a drag, 4-39  
 dreistellige Fehlernummer, 1-3  
 Drucken von ASCII-Dateien, auf einem  
   PostScript-Drucker, 3-53  
 Drucker  
   Druckjob abbrechen, 3-47  
   Priorität an einen Druckjob vergeben, 3-47  
   Starten eines Druckjobs, 3-46  
   Status eines Druckjobs anzeigen, 3-47  
 Druckerwarteschlange, Verarbeitung, 3-46  
 Druckjobs  
   abbrechen, 3-50  
   Priorität vergeben, 3-52  
   starten, 3-48  
   Status anzeigen, 3-51  
 Druckknöpfe, 4-22  
 DTE (Datenendeinrichtung), 6-3  
 Duplizieren (Option),  
   Arbeitsoberflächenverzeichnismenü, 5-31  
 Duplizieren (Option),  
   Arbeitsoberflächenverzeichnismenü Hilfe, 5-32

## E

E/A, umleiten, 2-28  
 E/A-Umadressierung  
   Bourne-Shell, B-21  
     Verwendung von Dateideskriptoren, B-23  
   C-Shell, C-27  
   Korn-Shell, A-35  
 Echtzeitnachrichten, an andere Benutzer senden,  
   mit dem Befehl write, 3-60–3-62  
 Edit\_Object, Arbeitsoberfläche, 5-3  
 Editieren, Befehlszeile, Korn-Shell, A-38  
 Editieren (Menü), Arbeitsoberfläche  
   Alle abwählen (Option), 5-27  
   Alle auswählen (Option), 5-27  
   Optionen, 5-27  
   Zurückstellen (Option), 5-27  
 Editieren (Menü), Arbeitsoberflächenverzeichnis  
   Alle abwählen (Option), 5-31  
   Alle auswählen (Option), 5-31  
   Nach Muster auswählen... (Option), 5-31  
   Optionen, 5-31  
 editieren, Dateisymbole, Arbeitsoberfläche, 5-20  
 editieren, Dokumentsymbole, Arbeitsoberfläche,  
   5-20  
 Editor, Arbeitsoberflächenmenü, 5-29  
 Editor ed, Funktionen, 2  
 Editor für elektronische Post  
   Beschreibung, 7-10  
   Kopffelder  
     ändern, 7-11–7-12, 7-28  
     hinzufügen, 7-11–7-12, 7-28  
   Nachricht  
     ändern, 7-11–7-12

    anzeigen, 7-11–7-12  
   Nachricht editieren, 7-40  
   Nachricht neu formatieren, 7-32  
   Texteditor auswählen, 7-54  
   Unterbefehle, 7-28  
 Editor INed, Funktionen, 2–3  
 Editor vi, Funktionen, 3  
 Editoren  
   ed, 2  
   edit, 3  
   ex, 3  
   INed, 2  
   vi, 3  
 Ein- und Ausgabe umleiten, von Koprozessen,  
   A-4  
 Ein-/Ausgabe. *Siehe* E/A  
 Einfahrt verboten, Symbol, Arbeitsoberfläche,  
   5-15  
 Einfügebereich, 4-36  
 Eingabebereich, 4-13  
 Eingabebereichsverfahren  
   explizit, 4-14  
   implizit, 4-14  
   Zeiger, 4-14  
 Eingabedialog, 4-30  
 Eingeschränkte Shell, Aufruf, B-3  
 Einheit /dev/rfd0, Diskettenlaufwerk,  
   Verwendung, 3-41  
 Einheit /dev/rmt0, Datenstrombandeinheit,  
   Verwendung, 3-41  
 Einheiten, Dateien, definieren für, 3-4  
 einleitendes Programm laden (IPL), 1-2  
 Einpassen, Symbole, Arbeitsoberfläche, 5-19  
 einstellbare Komponente, 4-32  
 Elektronische Handbuchseiten,  
   Online-Informationen anzeigen, 8-38–8-39  
 Elektronischer Briefkasten  
   Beschreibung, 7-14  
   Postordner, 7-14  
   Unterbefehle  
     Beschreibung, 7-15  
     Nachrichtenliste, 7-15  
     Ordnername, 7-16  
 Elektronischer Systembriefkasten, 7-14  
 Entfernen von Symbolen, Arbeitsoberfläche, 5-23  
 Entsprechungsliste, Einträge auswählen, 8-33  
 env, Befehl, 3-59  
 Erste Schritte (Option), Arbeitsoberflächenmenü  
   Hilfe, 5-29  
 expliziter Eingabebereich, 4-14  
 expressions, finding files with matching,  
   3-21–3-25

## F

Fehlerdialog, 4-31  
 Fenster  
   auf maximale Größe vergrößern, 4-17  
   auf Symbolgröße verkleinerte Fenster  
     wiederherstellen, 4-19  
   auswählen, 4-14  
   Basisfenster, 4-6  
   Beschreibung, 4-5  
   Geometrie, 4-6

- Größe ändern, 4-17
- Halten und Freigeben,
  - InfoExplorer-Grafikschnittstelle, 8-30
  - in den Vordergrund stellen, 4-14
  - in Symbol umwandeln, 4-18
- Lesefenster, 8-27, 8-28
- Menü, 4-6
- Navigationsfenster, 8-26
- Primärfenster, 4-6
  - schließen, 4-19
- Sekundärfenster, 4-6
  - Typen, 8-26
  - verschieben, 4-16
  - wiederherstellen, 4-18
- Fenster in den Vordergrund stellen, 4-14
- Fenster verschieben, 4-16
- Fenstermenü
  - auswählen aus, 4-15
  - Auswahlmöglichkeiten, 4-15
- Fenstermenüknopf, Position, 4-12
- Fensterobjekt, 4-8
- Fensterrahmen, Teile
  - Abbildung, 4-12
  - Knopf für maximale Größe, 4-17
- Fernanmeldung, Zweck, 2-33
- Ferne Knoten, 6-5
- Ferne Systeme
  - angemeldete Benutzer, Dialog führen, 6-21
- Dateien kopieren
  - Befehl ftp, 6-16
  - Befehl tftp, 6-23
- drucken an, 6-20
- drucken von, 6-19
- Host-Verbindungen, 6-22
- Informationen über Benutzer anzeigen,
  - Befehl finger, 6-15
- file, Befehl, 3-24
- File System, Journaled, 2-2
- Dateinamenssubstitution, C-Shell, C-24
- find, Befehl, 3-21
- Fraagedialog, 4-31
- fsck, Befehl, Funktion, 2-4
- Funktionstasten, für das Betriebssystem, 2-38

## G

- Ganzzahlberechnungen, A-23
- Gateways, Netz, 6-5
- Geisterprozeß, Beschreibung, 2-14
- Geistsymbole, Arbeitsoberfläche, Beschreibung, 5-14
- General Desktop, 5-39
  - sichtbare Dateien, 5-42
  - spezielle Dateien, 5-42
  - Symbole, 5-40
  - Zubehörsymbole, 5-43
- Geometrie von Fenstern, 4-6
- Getting Help (Option), Arbeitsoberflächenmenü, Hilfemenü, 5-29
- Grafikschnittstelle
  - Beschreibung, 1-4
  - erste Schritte, Überblick, 8-25–8-36
  - neue Funktionen, 8-9
- Grafische Benutzerschnittstelle, Definition, 4-3
- grep, Befehl, 3-22–3-25

- Größe, Fenstermenüauswahl, 4-15
- Größe von Fenster ändern, 4-17
- Größensteuerfeld, 4-32
- Größensteuerungselemente, Position, 4-12

## H

- Hardware, Informationen zur Systemhardware, 10-5
- Hauptfenster, 4-21
- head, Befehl, 3-24
- Hilfe
  - Anzeigen von Hilfetexten, 8-37
  - Aufrufen, Grafikschnittstelle, 8-34
  - Hilfelinie, 8-35
- Hilfe (Menü), Arbeitsoberfläche
  - Desktop-Überblick (Option), 5-29
  - Erste Schritte (Option), 5-29
  - Optionen, 5-29
  - README Desktop (Option), 5-29
  - Über (Option), 5-29
- Hilfe (Menü), Arbeitsoberflächenverzeichnis
  - Duplizieren (Option), 5-32
  - Kopieren (Option), 5-32
  - Löschen (Option), 5-33
  - Optionen, 5-32
  - Umbenennen (Option), 5-33
  - Verknüpfen (Option), 5-32
  - Verschieben (Option), 5-32
  - Verzeichnisüberblick (Option), 5-32
- Hilfe bei InfoExplorer, Liste von Hilfetexten aufrufen, 8-34
- Hilfelinie, 8-35
- Hilfetextkennungen, verfügbare, 8-13
- Hilfsprogrammoptionen für ... (Option), Arbeitsoberflächenmenü, 5-28
- Hintergrund, Fenstermenüauswahl, 4-15
- Hintergrundprozesse, Beschreibung, 2-14
- Host-Verbindungen
  - Befehle telnet, tn oder tn3270, 6-22
  - lokal /fern, 6-22
- Hypertextdatenbank
  - Blättern, InfoExplorer-Grafikschnittstelle, 8-29
  - Informationsarten, 8-3
  - Inhalt, 8-2
  - Überblick, 8-2
  - Verbindungen auswählen, 8-28
- Hypertextprogramm, Fenster, halten, 8-30

## I

- IDs
  - Benutzer-, 3-13
  - Gruppen-, 3-13
- idsrtstp, 2-33
- impliziter Eingabebereich, 4-14
- InfoExplorer
  - Anzeigen
    - Arten, 8-15
    - ASCII-Schnittstelle, 8-15
  - Aufgabenindex, 8-9
  - Beenden, ASCII-Schnittstelle, 8-24
  - Fenstertypen, 8-26
  - Hilfe, ASCII-Schnittstelle, 8-20



- in Anzeigentext blättern, ASCII-Schnittstelle, 8-18
- Informationen abrufen, 8-4
- Informationsarten, 8-3
- Liste der Hilfetexte, ASCII-Schnittstelle, 8-23
- Menüoptionen auswählen, ASCII-Schnittstelle, 8-18
- neue Funktionen, 8-9
- Starten, ASCII-Schnittstelle, 8-14
- Suchen, ASCII-Schnittstelle, 8-20
- Suchfunktion, 8-10
- Typen von Schnittstellen, Einführung, 8-8
- Überblick, 8-2
- Verbindungen auswählen, ASCII-Schnittstelle, 8-17
- Verlassen, ASCII-Schnittstelle, 8-24
- InfoExplorer-Grafikschnittstelle, Überblick
- Blättern, 8-29
- erste Schritte, 8-25–8-36
- Fenster halten und freigeben, 8-30
- Fenstertypen, 8-26
- Hilfe, 8-34
- Knöpfe auswählen, 8-28
- Menüoptionen auswählen, 8-29
- starten, 8-25
- Stoppen, 8-35
- Suchen, 8-32
- Verbindungen auswählen, 8-28
- zu einer vorherigen Position zurückkehren, 8-31
- Informationsdialog, 4-31
- Installation des codierten Zeichensatzes ISO8859, 5-58
- Interaktion mit MWM, 4-11
- interner Selbsttest (POST), 1-2
- ISO8859, codierter Zeichensatz, Installation, 5-58

## J

- Jobsteuerung
  - C-Shell, C-29
  - Korn-Shell, A-37

## K

- Kalender, anzeigen, mit dem Befehl cal, 3-61
- Kennungen, Hilfe, verfügbare, 8-13
- Kennwort, ändern, 1-6–1-10
  - mit dem Befehl passwd, 3-59–3-62
- kill, Befehl, 2-20
- Klient
  - Beschreibung, 4-5
  - Definition, 4-3
- Klienten, Beschreibung, 6-5
- Knopf für maximale Größe
  - Benutzung, 4-17
  - Position, 4-12
- Knopf für Symbolgröße, Position, 4-12
- Knöpfe
  - Abreißknopf, 4-28
  - Auswählen, 8-28
  - Druckknöpfe, 4-22
  - Knopf für Symbolgröße, 4-18
  - Markierungsknöpfe, 4-24
  - Maximieren, 4-17
  - Radioknöpfe, 4-23

- Umschaltknöpfe, 4-23
- Knoten
  - ferne, 6-5
  - lokale, 6-5
  - Netz, 6-5
- Kopieren (Option), Arbeitsoberflächenverzeichnismenü, 5-32
- Kopieren in... (Option), Arbeitsoberflächenverzeichnismenü, 5-31
- Koprozeßeinrichtung, Korn-Shell, A-3
- Korn-Shell
  - arithmetische Berechnungen, A-23–A-25
  - Aufrufen, A-1
  - Ausführungsstatus, A-3
  - Bedingungsausdrücke, A-9
  - Befehle
    - Verwendung, A-5
    - zusammengesetzt, A-7
  - Befehls-Alias-Namenprüfung, A-26
    - Tildensubstitution, A-27
  - Befehlsprotokoll, A-22
  - Befehlssubstitution, A-24
  - Befehlszeileneditierung, A-38
    - Editiermodus emacs, A-38
    - Editiermodus vi, A-41
  - Dateinamenssubstitution, A-33
  - Ein- und Ausgabe umleiten, A-35
  - Jobsteuerung, A-37
  - Koprozesse, Ein- und Ausgabe umleiten, A-4
  - Koprozeßeinrichtung, A-3
  - Parametersubstitution, A-28
  - Platzhalterzeichen, A-33
  - Quotierungszeichen, A-6
  - reservierte Wörter, A-8
  - Signalverarbeitung, A-3
  - Sonderbefehle, A-11
  - Umgebung, A-1

## L

- LAN (lokales Netz), Beschreibung, 6-3
- Leitwegwahl, Netz, 6-5
- Lesefenster, 8-27
- li, Befehl, 3-33
- Link-Verbindungen, 3-6
  - harte, 3-12
  - symbolische, 3-12
  - Überblick, 3-11
- Listenkomponente, 4-35
- LLC (Steuerung logischer Verbindungen), 6-5
- Lokaler Knoten, 6-5
- Lokales Netz. *Siehe* LAN
- Löschen (Option), Arbeitsoberflächenverzeichnismenü, 5-31
- Löschen (Option), Arbeitsoberflächenverzeichnismenü Hilfe, 5-33
- Löschen von Symbolen, Arbeitsoberfläche, 5-23
- ls, Befehl, 3-34
- lsfs, Befehl, Funktion, 2-3

## M

- MAC (Datenträgerzugriffssteuerung), 6-5
- Markierungsknöpfe, 4-24
- Maus, Arbeitsoberfläche
  - Basisaktionen der Maustasten, 5-10

- Doppelklicken, 5-10
- Ziehen und übergeben, 5-10
- drei Tasten, 5-9
- einfaches Klicken, 5-10
- Drücken-und-halten, 5-10
- einpassen, 5-10
- verwenden, 5-9
- Zeigen, 5-10
- Zeiger, 5-8
- Ziehen, 5-10
- zwei Tasten, 5-9–5-11
- Mausaktionen, 4-12
- Maustasten, Desktop
  - Basisaktionen, 5-10
  - Doppelklicken, 5-10
  - Ziehen und übergeben, 5-10
  - einfaches Klicken, 5-10
  - Drücken-und-halten, 5-10
  - einpassen, 5-10
  - Zeigen, 5-10
  - Ziehen, 5-10
- Maximieren, Fenstermenüauswahl, 4-15
- mehrere Arbeitsoberflächen, 5-2
- Menüknopf, Position, 4-12
- Menüleiste, 4-21, 4-24
- Menüleiste, Arbeitsoberfläche, 5-8
- Menüoptionen, Auswählen, 8-29
- Menüoptionen und Tastenäquivalente, Arbeitsoberfläche, 5-34
- Menüoptionen, Arbeitsoberfläche, auswählen, 5-25
- Menüs
  - Abreißmenüs, 4-28
  - Aktionsfenster, 4-24
  - Arten von, 4-24
  - Auswahl, 4-28
  - Dialogfenster, 4-27
  - Fenster, 4-15
  - Interaktion mit Festhalten der Maus, 4-25
  - Interaktion ohne Festhalten der Maus, 4-25
  - Navigation über die Tastatur, 4-38
  - Stammenü, 4-19
  - Tastatordirektaufrufe, 4-26
  - weiterführende, 4-25
- Menüs, Arbeitsoberfläche
  - Arbeitsoberflächenmenüs und -Optionen, 5-26
  - Menüoptionen auswählen, 5-25
  - Optionen und Tastenäquivalente, 5-34
  - Verwendung, 5-24
  - Verzeichnismenüs und Optionen, 5-30
- Merkmale... (Option), Arbeitsoberflächenverzeichnismenü, 5-31
- Merkmale, Arbeitsoberfläche, 5-2
- Minimieren, Fenstermenüauswahl, 4-15
- mkdir, Befehl, 3-31
- mkfs, Befehl, Funktion, 2-4
- Mnemonicische Zeichen, 4-16, 4-26
- Mnemonicische Zeichen, Arbeitsoberfläche, Menü anzeigen, 5-34
- mnemonicische Zeichen, Arbeitsoberfläche, 5-34
- mnemonicische Zeichen, Window Manager, Menüoption auswählen, 5-34
- Modaldialogfenster, 4-29

- Modi. *Siehe* Zugriffsmodi
- Moduswählschalter, einstellen, 1-2
- more, Befehl, 3-19
- mount, Befehl, Funktion, 2-3, 2-4
- mv, Befehl, 3-17, 3-19, 3-31
- MWM
  - auf Symbolgröße verkleinerte Fenster wiederherstellen, 4-19
  - Auswählen eines Fenster, 4-13
  - Benutzung der Tastatur, 4-12
  - Benutzung einer Maus, 4-12
  - Einführung, 4-7
  - Fenster auf maximale Größe vergrößern, 4-17
  - Fenster in den Vordergrund stellen, 4-14
  - Fenster in Symbol umwandeln, 4-18
  - Fenster schließen, 4-19
  - Fenster verschieben, 4-16
  - Fensterelemente, 4-12
  - Fenstermenü, 4-15
  - Größe von Fenstern ändern, 4-17
  - Interaktion mit, 4-11
  - neu starten, 4-20
  - starten, 4-9
  - Symbole, 4-18
  - Symbofeld, 4-18
  - Verschieben eines Symbols, 4-19
  - Zeigerformen, 4-11
- MWM neu starten, 4-20

## N

- Nach Muster auswählen... (Option), Arbeitsoberflächenverzeichnismenü, 5-31
- Nachrichten
  - an andere Benutzer senden, mit dem Befehl write, 3-60–3-62
  - auf der Anzeige löschen, 3-62
- Nachrichtendialog, 4-30
- Namen (Option), Arbeitsoberflächenverzeichnismenü, 5-32
- Namensanzeige, Arbeitsoberflächenmenü, 5-29
- Navigation, Text, 4-36
- Navigation über die Tastatur, 4-38
- Navigationsfenster, 8-26
  - Überblick über Fenster, 8-26
- Navigationsfunktionen, neue Funktionen, 8-9
- Netze
  - lokale Netze. *Siehe* LAN
  - öffentliche Netze. *Siehe* WAN
  - physische, 6-3–6-7
  - Post, 7-4
    - Leitwegwahl, 7-4
    - Überblick, 6-2–6-7
- Neu anordnen (Option), Arbeitsoberflächenmenü, 5-28
- Neue Datei (Option), Arbeitsoberflächenverzeichnismenü, 5-30
- Neue Funktionen, InfoExplorer, 8-9
- Neues Desktop... (Option), Arbeitsoberflächenmenü, 5-27
- Neues Verzeichnis (Option), Arbeitsoberflächenverzeichnismenü, 5-30
- Nichtmodales Dialogfenster, 4-29

nl, Befehl, 3-28  
Notizen, Einführung, 8-8

## O

Öffentliches Netz. *Siehe* WAN  
Öffnen (Option), Arbeitsoberfläche,  
Verzeichnismenü, 5-30  
Öffnen (Option), Arbeitsoberflächenmenü, 5-27  
Öffnen von Dokumentsymbolen,  
Arbeitsoberfläche, 5-20  
Öffnen von Verzeichnissen, Arbeitsoberfläche,  
5-20  
Öffnen, Option, Arbeitsoberflächenmenü, 5-26  
Online, Anzeigen von Hilfetexten, 8-37  
Optionen (Menü), Arbeitsoberfläche  
  Befehloptionen für ...(Option), 5-29  
    Befehl cc, 5-29  
    Befehl ps, 5-29  
    Namensanzeige, 5-29  
  Hilfsprogrammoptionen für ...(Option), 5-28  
    Blättern, 5-29  
    Editor, 5-29  
  Optionen, 5-28  
  Präferenzen (Option), 5-28  
  Standardeinheiten für ...(Option), 5-28  
    1/4 Zoll Band, 5-28  
    8-mm-Band, 5-28  
    Alternativdrucker, 5-28  
    Primärdrucker, 5-28  
Optionen des Arbeitsoberflächenmenüs, 5-26.

## P

pack, Befehl, 3-29–3-30  
  Funktion, 3-40  
Papierkorbsymbol, Arbeitsoberfläche  
  Beschreibung, 5-14  
  Inhalt anzeigen, 5-23  
  Inhalt löschen, 5-23  
  Inhalt wiederherstellen, 5-23  
Parametersubstitution, Korn-Shell, A-28  
passwd, Befehl, 1-6–1-10, 3-59–3-62  
paste, Befehl, 3-27–3-30  
Pfade, 3-9  
Pfadfunktion, InfoExplorer-Grafikschnittstelle,  
8-31  
Pfadname, Anzeigen des Arbeitsverzeichnisses,  
3-32–3-37  
Pfadnamen, 3-9–3-10  
  absolute, 3-2, 3-10  
  Definition, 3-2  
  relative, 3-9  
pg, Befehl, 3-19  
Platzhalterzeichen  
  Bourne-Shell, B-19  
  Korn-Shell, A-33  
Positionscursor, 4-36  
Positionsparameter, Bourne-Shell, B-18  
Post, Nachrichten erstellen, Rechtschreibprüfung,  
7-31  
Post adressieren  
  Beschreibung, 7-3  
  BNU-(UUCP-)Verbindung, 7-7  
  Domänenname, 7-6

Post beantworten, 7-35  
Post empfangen  
  Befehl mail, 7-33  
  Beschreibung, 7-13  
  elektronischer Briefkasten  
    Beschreibung, 7-14  
    bestimmte Nachrichten anzeigen, 7-38  
  Datei dead.letter, 7-14  
  elektronischer Systembriefkasten, 7-14  
  Inhalt anzeigen, 7-37  
  Nachricht editieren, 7-40  
  persönlicher (mbox), 7-14  
  Post speichern, 7-41  
  Unterbefehle  
    Beschreibung, 7-13, 7-15–7-17  
    Nachrichtenliste angeben, 7-15  
    Ordnername, 7-16  
    Verwendung von Ordnern, 7-43  
  geheime Post, 7-47  
  lesen, 7-33  
  Unterbefehle, 7-22  
  weiterleiten, 7-13, 7-45  
Post lesen, 7-13  
Post senden  
  beantworten, 7-35  
  Befehl mail, 7-23  
  geheime Post, 7-47  
  lokales System, 7-4  
  Nachricht neu formatieren, 7-32  
  Netze, 7-4  
  Unterbefehle, 7-25  
  Urlaubsnachricht, 7-46  
Postfunktionen  
  Antworten, 7-35  
  Datei dead.letter, 7-14  
  Datei mbox, 7-14  
  Empfangen  
    Befehl mail, 7-33  
    Unterbefehle, 7-22  
  Lesen, 7-13, 7-33  
  Ordner, 7-15  
  Post speichern, 7-41  
  Programme, Post, 7-3  
  Senden  
    Befehl mail, 7-23  
    Netze, 7-4  
    Unterbefehle, 7-25  
  Überblick, Benutzer, 7-2  
Postprogramm anpassen  
  Alias-Namen  
    Liste anzeigen, 7-51  
    Liste erstellen, 7-50  
  Anzeige, 7-19–7-22  
  Beschreibung, 7-18–7-19  
  Betrefffeld, 7-52  
  erforderliche Schritte, 7-48  
  Kopienfeld (Cc:), 7-53  
  Optionen  
    aktivieren/inaktivieren, 7-19  
    Beschreibung, 7-18–7-19  
  Texteditor, 7-54  
  Verteiler, 7-50, 7-51  
Postprogramm konfigurieren

- Anzeige, 7-19–7-22
- Optionen
  - aktivieren/inaktivieren, 7-19
  - Beschreibung, 7-18–7-19
- Power Desktop, 5-45
  - sichtbare Dateien, 5-49
  - Sonderdateien, 5-48
  - Symbole, 5-46
- Power, Verzeichnis, Symbole, 5-49
- Präferenzen (Option), Arbeitsoberflächenmenü, 5-28
- Primärdrucker, Arbeitsoberflächenmenü, 5-28
- Primärfenster, 4-6
- Profile, Verwendung, 2-31
- Programm, Ausgabe in eine Datei kopieren, 3-20–3-25
- Programmierung, Arbeitsoberfläche, 5-3
- Programmsymbole, Arbeitsoberfläche, 5-20
  - ausführen, 5-20
  - Beschreibung, 5-13
- Protokoll, Einführung, 8-8
- Protokolle, Netz, allgemein, 6-4
- Protokollfunktion, InfoExplorer-Grafikschnittstelle, 8-31
- Protokollsubstitution, C-Shell, C-14
- Prozeß qdaemon, Beschreibung, 2-14
- Prozeß-ID, Beschreibung, 2-14
- Prozesse
  - abbrechen, 2-16
  - alle aktiven anzeigen, 2-18
  - Beschreibung, 2-14
  - löschen, 2-20
  - starten, 2-15
  - Status anzeigen, 2-19
- Prozeßidentifikationsnummer. *Siehe* Prozeß-ID
- ps, Befehl, 2-18, 2-19

## Q

- Quotierungszeichen
  - Bourne-Shell, B-5
  - Korn-Shell, A-6

## R

- Radioknöpfe, 4-23
- rdump, Befehl, Funktion, 3-40
- README Desktop (Option), Arbeitsoberflächenmenü Hilfe, 5-29
- Reservierte Wörter, Korn-Shell, A-8
- restore, Befehl, Funktion, 2-4, 3-41
- rm, Befehl, 3-17
- rmdir, Befehl, 3-36
- rmfs, Befehl, Funktion, 2-3
- rrstore, Befehl, Funktion, 3-41

## S

- Schiebefeld, 4-33
- Schiebeleiste, 4-21, 4-33
  - Verwendung, 8-30
- Schiebeleisten, Überblick, 8-29
- Schließen, Fenstermenüauswahl, 4-15
- Schließen von Fenstern, 4-19
- Schreib-/Lesezugriff, Arbeitsoberfläche, 5-15
- Sekundärfenster, 4-6

- Server, 4-5
  - Beschreibung, 6-5
- Shell-Prozeduren
  - eine Datei angeben, 2-25
  - eine Shell angeben, 2-25
  - Verwendung, 2-25
- Shell-Umgebungen, anpassen, mit Hilfe von Profilen, 2-31
- Shells
  - Begriffe, Definitionen, 2-22
  - Bourne-Shell
    - Aufruf, B-1
    - bedingte Substitution, B-17
    - Befehlssubstitution, B-12
    - integrierte Befehle, B-7
    - Dateinamenssubstitution, B-19
    - Ein- und Ausgabe umleiten, B-21
    - Positionsparameter, B-18
    - Sonderbefehle, B-7
    - Umgebung, B-1
    - benutzerdefinierte Variablen, B-13
    - Variablen, B-14
    - Variablensubstitution, B-13
    - vordefinierte Sondervariablen, B-16
  - C-Shell
    - Alias-Namenssubstitution, C-18
    - Aufruf, C-1
    - Befehlssubstitution, C-13
    - integrierte Befehle, C-3
    - Dateinamenssubstitution, C-24
    - Ein- und Ausgabe umleiten, C-27
    - Jobsteuerung, C-29
    - Protokollsubstitution, C-14
    - Signalverarbeitung, C-1
    - Variablensubstitution, C-19
    - vordefinierte und Umgebungsvariablen, C-21
  - eingeschränkt, Aufruf, B-3
  - Korn-Shell
    - arithmetische Berechnungen, A-23
    - Aufrufen, A-1
    - Ausführungsstatus, A-3
    - Bedingungsausdrücke, A-9
    - Befehls-Alias-Namenprüfung, A-26
    - Befehlsprotokoll, A-22
    - Befehlssubstitution, A-24
    - Befehlszeileneditierung, A-38
    - Befehlszeileneditierung mit emacs, A-38
    - Befehlszeileneditierung mit vi, A-41
    - Dateinamenssubstitution, A-33
    - Ein- und Ausgabe umleiten, A-35
    - Jobsteuerung, A-37
    - Koprozeßeinrichtung, A-3
    - Parametersubstitution, A-28
    - Quotierungszeichen, A-6
    - reservierte Wörter, A-8
    - Signalverarbeitung, A-3
    - Sonderbefehle, A-11
    - Umgebung, A-1
    - Verwendung von Befehlen, A-5
    - zusammengesetzte Befehle, A-7
- Shell-Prozeduren
  - eine Datei angeben, 2-25

- eine Shell angeben, 2-25
- Verwendung, 2-25
- Typen, 2-22
- Überblick, 2-21
- Sichern, Zweck, 3-40
- Sicherung, Datenträger, Abbildung, 3-41
- Sicherungskopien, 3-7
- Signalverarbeitung
  - Bourne-Shell, B-5
  - C-Shell, C-1
  - Korn-Shell, A-3
- Skala, 4-34
- SMIT, Drucken, steuern, 3-46
- Sohnprozeß, Beschreibung, 2-14
- Sonderbefehle
  - Bourne-Shell, B-7
  - Korn-Shell, A-11
- Sonderdateien
  - Block-, 3-4
  - FIFO- (First In/First Out), 3-4
  - Pipes, 3-4
  - Zeichen-, 3-4
- Sortieren (Option),
  - Arbeitsoberflächenverzeichnismenü, 5-32
  - menu, 5-32
- Sortieren (Option),
  - Arbeitsoberflächenverzeichnismenü Anzeigen
  - Nach Größe, 5-32
  - Nach Name, 5-32
  - Nach Verzeichnis, 5-32
- spezielle Symbole, Arbeitsoberfläche,
  - Beschreibung, 5-14
- Stammenü, 4-19
- Standard-Shell, Bedingungsausdrücke, A-9
- Standardeingabe, in eine Datei kopieren, 3-20
- Standardeinheiten für ... (Option),
  - Arbeitsoberflächenmenü, 5-28
  - , 5-28
- Standardeinheiten für ... (Option),
  - Arbeitsoberflächenmenü Optionen,
  - Primärdrucker, 5-28
- Standardeinstellungen, wiederherstellen, 4-9
- Starten der Arbeitsoberfläche, 5-5
- Starten der Arbeitsoberfläche in der
  - Anmelde-Shell, 5-5
- Starten der Arbeitsoberfläche mit Befehl, 5-6
- Starten der Arbeitsoberflächen, automatisch, 5-6
- Starten von MWM, 4-9
- Starten von X, 4-9
- Startprozedur für Desktop, ändern, 5-6
- Steuerung logischer Verbindungen. *Siehe* LLC
- Stoppen, InfoExplorer, 8-35
- su, Befehl, 3-59-3-62
- Suche, einfache, Grafikschnittstelle, 8-32
- Suchen, nach Aufgaben, verfügbare Kennungen,
  - 8-13
- Suchfenster, InfoExplorer, 8-32
- Symbol
  - Definition, 4-18
  - Feld, Beschreibung, 4-18
- Symbole, 5-8
- Symbole (Option),
  - Arbeitsoberflächenverzeichnismenü, 5-32

- Symbole für das Verzeichnis mit Anwendungen,
  - Arbeitsoberfläche, Beschreibung, 5-13
- Symbole für Verzeichnis mit Zubehör,
  - Arbeitsoberfläche, Beschreibung, 5-13
- Symbole, Arbeitsoberfläche, 5-2
  - aktivieren, 5-20
  - Anwendung, 5-13
  - Anwendungen ausführen, 5-20
  - Arbeitsoberfläche, 5-14
  - Ausführungsberechtigung, 5-16
  - Auswahl zurücknehmen, 5-18, 5-19
  - Auswahl zurücknehmen durch Einpassen,
    - 5-19
  - auswählen, 5-18
  - Benutzerverzeichnis, 5-13
  - Brille, 5-15
  - Datei, 5-14
  - Dateien editieren, 5-20
  - Dateien öffnen, 5-20
  - Dokument, 5-14
  - Dokumente editieren, 5-20
  - Dokumente öffnen, 5-20
  - doppelt klicken, 5-20
  - ziehen und übergeben, 5-21
  - Verwendung von Ziehen und Übergeben,
    - 5-20
  - Einführung, 5-12
  - entfernen, 5-23
  - Geist, 5-14
  - Getting Started, 5-14
  - in ein anderes Verzeichnis kopieren, 5-21
  - in ein anderes Verzeichnis verschieben, 5-21
  - kopieren, 5-21
  - löschen, 5-23
  - Menüs, 5-16
  - Einfahrt verboten, 5-15
  - Papierkorb, 5-14
  - Papierkorbsymbol
    - Inhalt anzeigen, 5-23
    - Inhalt löschen, 5-23
    - Inhalt wiederherstellen, 5-23
  - Programm, 5-13
  - Programme ausführen, 5-20
  - Schreib-/Lesezugriff, 5-15
  - auswählen durch Erstellen eines dehnbaren
    - Rahmens, 5-18
  - spezielle, 5-14
  - Typen, 5-13
  - umbenennen, 5-22
  - verschieben, 5-21-5-23
  - Verzeichnis, 5-13
  - Verzeichnis mit Anwendungen, 5-13
  - Verzeichnisse öffnen, 5-20
  - Vorhängeschloß, 5-15
  - Warnungen, beim Kopieren oder
    - Verschieben, 5-22
  - Zubehörverzeichnis, 5-13
  - Zugriff auf Hilfemenüs, 5-16
  - Zugriff auf Menüs, 5-16
  - Zugriffsmerkmale, 5-15
- Symbolgröße, Beschreibung, 4-18
- Symbolmenüs, Arbeitsoberfläche, 5-2, 5-16
  - Beispiel, 5-17

- Zugriff, 5-16
- Zugriff auf Hilfe, 5-16
- Syntaxdiagramm
  - numerierte Beispiele, Abbildung, 2-42
  - Nummern, 2-42-2-44
- System
  - Netzschalter, 1-2
  - starten, 1-2
- Systemeinheit, Hardwareinformationen, 10-5
- Systemverwaltung
  - Dateisysteme, Aufgaben, 2-3
  - mehrere Dateisysteme, Verwendung, 2-2

## T

- tar, Befehl, Funktion, 3-40, 3-41
- Tastatur, Benutzung, 4-38
- Tastaturaktionen, 4-12
- Tastaturdirektaufrufe, 4-26
- Tastatureingabebereich, 4-14
- Tastenfolgen
  - für das Betriebssystem, 2-38
  - Tabelle, 1-7
- Tastenfunktionen, 1-7
- TCP/IP \*
  - Befehl rlogin, Überblick, 6-12
  - Befehle
    - Befehl rexec, 6-10, 6-13
    - Befehl rlogin, 6-12-6-13
    - Befehl rsh, 6-12, 6-13
    - Befehl telnet, 6-12
    - Dateiübertragung, 6-9
    - Endbenutzer, 6-9
    - Fernanmeldung, 6-10
    - ferne Ausführung, 6-13-6-14
    - Fernverbindung, 6-11
    - Netzstatus, 6-10
  - Befehle für Dateiübertragung, Befehl ftp, 6-16
  - Befehle für Fernanmeldung
    - Befehl rlogin, 6-10
    - Befehle rsh oder remsh, 6-10
    - Befehle tn, telnet oder tn3270, 6-10
  - Befehle für Fernverbindung
    - Befehl mail, 6-11
    - Befehl talk, 6-11
  - Befehle tn, telnet oder tn3270
    - 3270-Datenstationsemulation, 6-12
    - Datenstationsemulation für DEC VT100, 6-12
    - Überblick, 6-12
  - Befehle zur Dateiübertragung
    - Befehl ftp, 6-9
    - Befehl rcp, 6-9
    - Befehl tftp, 6-23
    - Befehle tftp oder utftp, 6-9
  - Dateien kopieren
    - Befehl ftp, 6-16
    - Befehl tftp, 6-23
  - Defense Advanced Research Projects Agency (DARPA), 6-8
  - Echtzeitdialog, Befehl talk, 6-21
  - File Transfer Protocol (FTP), 6-9

- Informationen über Benutzer anzeigen, Befehl finger, 6-15

## Netze

- lokales Netz (LAN), 6-8
- öffentliches Netz (WAN), 6-8
- Überblick, 6-8
- Sicherheit
  - gesicherte Befehle, 6-12
  - nicht gesicherte Befehle, 6-12-6-13
- Statusbefehle
  - Befehl finger oder f, 6-10
  - Befehl host, 6-10
  - Befehl ping, 6-10
  - Befehl rwho, 6-10
  - Befehl whois, 6-10
  - Trivial File Transfer Protocol (TFTP), 6-9
- TCP/IP (Transmission Control Protocol/Internet Protocol)
  - Siehe auch* Internet
  - Internet, 6-8
  - Überblick, Benutzer, 6-8
- TCP/IP-Druckoperationen, ferne Systeme
  - drucken an, 6-20
  - drucken von, 6-19
- TCP/IP-Protokolle, Überblick, 6-8
- tee, Befehl, 3-20
- Teilfenster, 4-32
- Text, eingeben und editieren, 4-36
- Text auswählen, 4-37
- Text eingeben und editieren, 4-36
- Text übertragen
  - in die Zwischenablage übertragen, 4-37
  - Primärübertragung, 4-37
  - Schnellübertragung, 4-37
  - Übertragung durch Ziehen, 4-37
- Texteditoren, Überblick, 2-3
- Textformatierung, Überblick, 9-4
- Textkomponente, 4-36
- Tildensubstitution, Befehls-Alias-Namenprüfung, Korn-Shell, A-27
- Titelbereich, Position, 4-12
- Titelleiste, Position, 4-12
- Transmission Control Protocol/Internet Protocol. *Siehe* TCP/IP
- Typen, Arbeitsoberfläche, General Desktop, 5-39
- Typen, Arbeitsoberflächen, 5-38

## U

- Über (Option), Arbeitsoberflächenmenü Hilfe, 5-29
- Übergabepunkt, 4-39
- Umbenennen (Option),
  - Arbeitsoberflächenverzeichnismenü Hilfe, 5-33
- Umbenennen von Symbolen, Arbeitsoberfläche, 5-22
- Umbenennen (Option),
  - Arbeitsoberflächenverzeichnismenü, 5-31
- Umgebung, aktuelle Umgebung anzeigen, 3-59
- umount, Befehl, Funktion, 2-4
- Umschaltknöpfe, 4-23
- Umwandeln eines Fensters in ein Symbol, 4-18

uname, Befehl, Systemnamen anzeigen, 3-58

unpack, Befehl, 3-30

Funktion, 3-40

Unterbrechen, Tastenfolge, 1-7

## V

Variablen

Bourne-Shell, B-14

benutzerdefiniert, B-13

vordefinierte Sondervariablen, B-16

C-Shell, vordefinierte und  
Umgebungsvariablen, C-21

Variablensubstitution

Bourne-Shell, B-13

C-Shell, C-19

Verknüpfen (Option),

Arbeitsoberflächenverzeichnismenü Hilfe, 5-32

Verknüpfen mit... (Option),

Arbeitsoberflächenverzeichnismenü, 5-31

Verschieben, Fenstermenüauswahl, 4-15

Verschieben (Option),

Arbeitsoberflächenverzeichnismenü Hilfe, 5-32

Version 3

CD-ROM-Dateisystem, 2-3

Journaled File System, 2-3

Network File System, 2-3

Verteiler, Post, 7-50

Verwandte Editoren, allgemeine Informationen, 3

Verzeichnis, Pfadnamen anzeigen, 3-32-3-37

Verzeichnis öffnen... (Option),

Arbeitsoberflächenmenü, 5-26

Verzeichnis schließen (Option),

Arbeitsoberflächenverzeichnismenü, 5-31

Verzeichnisse

ändern, Überblick, 3-11

anzeigen, Überblick, 3-11

Befehl zum Arbeiten mit, 3-11

benennen, 3-4

Benutzer-, Definition, 3-9

Dateien auflisten, 3-33

Definition, 3-2, 3-4

DOS-Dateien auflisten, 3-39

Eignerstatus ändern, 3-36

Einführung, 3-8

Einträge, 3-8

entfernen, 3-36-3-37

Überblick, 3-11

erstellen, 3-31-3-37

Überblick, 3-11

Inhalt anzeigen, 3-33

kopieren, 3-17-3-25, 3-33-3-37

Überblick, 3-11

löschen, 3-36-3-37

mit Verzeichnissen arbeiten, 3-31-3-37

Pfadnamen, 3-9

Stamm-, Definition, 3-2

Struktur, 3-8

Typen, 3-10-3-12

umbenennen, Überblick, 3-11

Unterverzeichnisse, 3-8

Vater-, 3-8

verbinden, 3-11

verschieben, 3-31

Überblick, 3-11

wechseln, 3-32-3-37

Zugriffsmodi, 3-13

zurückkehren in das, Benutzerverzeichnis,  
3-9

zwei Verzeichnisse vergleichen, 3-37

Verzeichnisse . (Punkt), 3-10

Verzeichnisse .. (Punkt Punkt), 3-10

Verzeichnissymbole, Arbeitsoberfläche  
Beschreibung, 5-13

öffnen, 5-20

Verzeichnisüberblick (Option),

Arbeitsoberflächenverzeichnismenü, 5-32

virtuelle Datenstationen

Prozedur zum Starten, 2-40

Prozedur zum Stoppen, 2-40

Vordergrundprozesse, Beschreibung, 2-14

Vorgehensweise beim Sichern, allgemeine  
Richtlinien, 3-42

Vorhängeschloß auf Symbol, Arbeitsoberfläche,  
5-15

Vorherige Position, Zurückkehren, 8-31

## W

WAN (öffentliches Netz), Beschreibung, 6-3

Warnungen, Arbeitsoberfläche, beim Kopieren  
oder Verschieben, 5-22

Warnungsdialo, 4-31

wc, Befehl, 3-23-3-25

Weiterführendes Menüs, 4-25

Wiederherstellen, Fenstermenüauswahl, 4-15

Wiederherstellen der Standardeinstellungen, 4-9

window, help, 8-34

Window Manager

Beschreibung, 4-7

Interaktion mit der Arbeitsoberfläche, 4-4

Window Manager und AIXwindows Desktop,  
Interaktion, 4-4

Wörter, zählen, mit dem Befehl wc, 3-23-3-25

Wörter suchen, Einführung, 8-6

write, Befehl, 3-60-3-62

## X

X Server, 4-5

X Window System

Beschreibung, 4-5

starten, 4-9

xdt3newuser, Befehl, Definieren neuer Benutzer,  
5-57

X-Window System, Definition, 4-3

## Z

Zeichen, zählen, mit dem Befehl wc, 3-23-3-25

Zeiger, 4-11

Zeigereingabebereich, 4-14

Zeigerformen, 4-11

Zeilen, zählen, mit dem Befehl wc, 3-23-3-25

Ziehen und übergeben

Hilfe während eines Ziehvorgangs, 4-42

Operationssymbol, 4-40

Quellensymbol, 4-40

Statussymbol, 4-41

Übergabepunkt, 4-39

Zieheffekte für Ziehsymbol, 4-41

- Ziehoperationen, 4-39
- Ziehsymbol, 4-40
- Ziehvorgang abbrechen, 4-43
- Zielquelle, 4-39
- Ziehen und Übergeben von Symbolen,  
Arbeitsoberfläche, 5-21
- Ziehenn und übergeben, gezogenes Objekt  
übergeben, 4-42
- Zielquelle, 4-39
- Zugriffsmerkmale für Symbole, Arbeitsoberfläche,  
5-15
- Zugriffsmerkmale, Arbeitsoberfläche  
ausführbar, 5-16
- Schreib-/Lesezugriff, 5-15
- Zugriffsmodi

- ändern, 3-15
- Darstellung
  - numerisch, 3-14
  - symbolisch, 3-13
- Dateien, 3-13
- Standardeinstellung
  - numerische Darstellung, 3-15
  - symbolische Darstellung, 3-14
- steuern, 3-13
- Verzeichnisse, 3-13
- Zurückstellen (Option), Arbeitsoberflächenmenü,  
5-27







**IBM Deutschland Informationssysteme GmbH**  
**MH NLS Center**  
**Postfach 80 08 80**

**70548 Stuttgart 80**

---

**Absender:**

\_\_\_\_\_

Name

\_\_\_\_\_

Firma

\_\_\_\_\_

Straße

\_\_\_\_\_

PLZ

\_\_\_\_\_

Ort

\_\_\_\_\_

Telefon

\_\_\_\_\_

Telefax

---

**AIX Version 3.2**  
**Dokumentationsüberblick**  
**IBM Form SC23-2456-01 (SC12-2158-01)**





**IBM Deutschland Informationssysteme GmbH**  
**MH NLS Center**  
**Postfach 80 08 80**  
**70548 Stuttgart 80**

---

**Absender:**

\_\_\_\_\_

Name

\_\_\_\_\_

Firma

\_\_\_\_\_

Straße

\_\_\_\_\_

PLZ

\_\_\_\_\_

Ort

\_\_\_\_\_

Telefon

\_\_\_\_\_

Telefax

---

**AIX Version 3.2**  
**Dokumentationsüberblick**  
**IBM Form SC23-2456-01 (SC12-2158-01)**





Printed in Italy



Printed on recycled paper

\* G C 1 2 - 2 0 9 2 - 0 3 \*