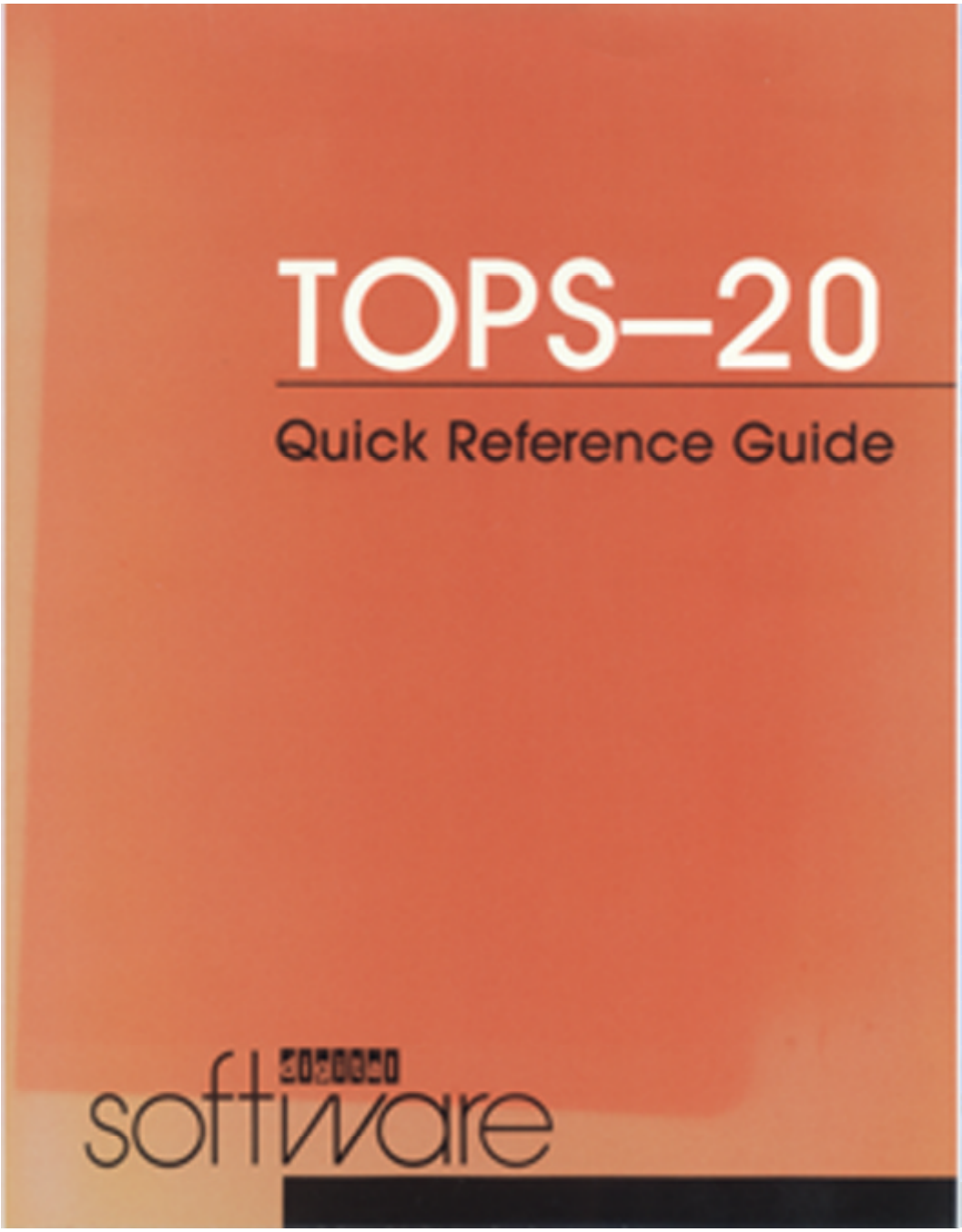
TOPS–20

Quick Reference Guide



TOPS–20
Monitor Calls
Quick Reference Guide

AV–P173A–TM

December, 1982

This guide provides a brief description of all of the TOPS–20 monitor calls and many of the blocks in the monitor's data base. It is intended for use by experienced MACRO–20 programmers who require a reminder of calling sequences and function codes. MACRO–20 programmers who require a more detailed description of the monitor calls should use the *TOPS–20 Monitor Calls Reference Manual*: those who desire a more introductory discussion on the use of monitor calls should refer to the *TOPS–20 Monitor Calls User's Guide*.

OPERATING SYSTEM: TOPS–20, V5.1

Software and manuals should be ordered by title and order number. In the United States, send orders to the nearest distribution center. Outside the United States, orders should be directed to the nearest DIGITAL Field Sales Office or representative.

Northeast/Mid-Atlantic Region

Digital Equipment Corporation
PO Box CS2008
Nashua, New Hampshire 03061
Telephone:(603)884–6660

Central Region

Digital Equipment Corporation
Accessories and Supplies Center
1050 East Remington Road
Schaumburg, Illinois 60195
Telephone:(312)640–5612

Western Region

Digital Equipment Corporation
Accessories and Supplies Center
632 Caribbean Drive
Sunnyvale, California 94086
Telephone:(408)734–4915

digital equipment corporation • marlboro massachusetts

© Digital Equipment Corporation, 1982.
All Rights Reserved.

Printed in U.S.A.

TABLE OF CONTENTS

CONVENTIONS

Bn or Bn-m	Bit n or bits n through m; bit positions are always decimal
(enabled priv)	Designates a capability that must be enabled for the specified function to be legal
filespec	Designates a complete TOPS-20 file specification
IPCF	IPCF capability required
underline	Designates a variable argument, as in 1Bn
MNT	MAINTENANCE capability required
mss.	Milliseconds
number	Designates an octal number
number.	Designates a decimal number
number.number	Designates a floating point number
OPR	OPERATOR capability required
OWGBP	A One Word Global Byte Pointer; see Pointer Formats for format
(priv)	Designates a capability that must exist for the specified function to be legal, but need not be enabled
R-J	The data is or should be right-justified in the specified field
<value>,,<value>	The left and right half-word (18-bit) values of a full-word (36-bit) value
WHL	WHEEL capability required

MONITOR CALLS FUNCTIONAL ORGANIZATION

Accounting Functions

GACCT	Reads a job's account
GACTF	Reads a file's account
LOGIN	Logs a job into the system
SACTF	Sets a file's account
USAGE	Writes entries into the system's accounting file
VACCT	Validates an account

File Functions

ACCES	Allows access to a directory
BKJFN	Backspaces file's pointer
CHFDB	Changes a File Descriptor Block
CHKAC	Checks access to a file
CLOSF	Closes a file
CLZFF	Closes a process' files
CRLNM	Creates a logical name
DELDF	Expunges deleted files
DELF	Deletes a file
DELNF	Retains specified number of generations of file
DIRST	Translates directory or user number to a string
FFFFP	Finds first free file page
FFUFD	Finds first used file page
GACTF	Reads a file's account
GFUST	Reads the author or last writer name string
GNJFN	Assigns a JFN to the next file
GPJFN	Returns primary JFN's
GTFDB	Reads a File Descriptor Block
GTJFN	Assigns a JFN to a file
GTSTS	Reads file's status
INLNM	Writes logical names
JFNS	Translates a JFN to a string
LNMST	Translates logical name to string
OPENF	Opens a file
RCDIR	Translates directory name to number
RCUSR	Translates user name to number
RFBSZ	Reads file's byte size
RFPTR	Reads file's pointer
RFTAD	Reads file's time and dates
RLJFN	Releases a JFN
RNAMF	Renames a file
SACTF	Sets a file's account
SFBSZ	Sets file's byte size
SFPTR	Sets file's pointer
SFTAD	Sets file's time and dates
SFUST	Changes the author or last writer name string
SIZEF	Obtains file's length
SPJFN	Sets primary JFN's
STSTS	Sets file's status
SWJFN	Transposes two JFN's
UFPGS	Updates file's pages

WILD%	Compares a wild filespec against a non-wild filespec

I/O Functions

BIN	Reads the next byte
BOUT	Writes the next byte
DUMPI	Reads data in unbuffered data mode
DUMPO	Writes data in unbuffered data mode
FLIN	Reads a floating-point number
FLOUT	Writes a floating-point number
NIN	Reads a number
NOUT	Writes a number
PSOUT	Writes string to primary output designator
PBIN	Reads byte from primary input designator
PBOUT	Output byte to primary output designator
PMAP	Maps pages
RDTTY	Reads data from primary input designator
RIN	Reads a byte nonsequentially
ROUT	Writes a byte nonsequentially
RSCAN	Reads and outputs rescan buffer
SIN	Reads a string
SOUT	Writes a string
SINR	Reads a record
SOUTR	Writes a record
SMAP%	Maps sections
TEXTI	Reads data from terminal or file

Information Functions

ERSTR	Translates an error code to a string
ESOUT	Returns an error string
GETAB	Returns a word from a system table
GETER	Returns the last error condition
GETJI	Returns job information for specified job
GETNM	Returns the program name being used by the job
GJINF	Returns job information for current job
GTAD	Returns the system's date
GTDAL	Returns the disk allocation of a directory
GTRPI	Returns the paging trap information
GTRPW	Returns the trap words
HPTIM	Returns the high-precision clock values
RUNTM	Returns the runtime of a job or process
SYSGT	Returns values for a system table
TIME	Returns the time since the system was restarted

Device Control Functions

ASND	Assigns a device
ATACH	Attaches controlling terminal to a job
CFIBF	Clears terminal's input buffer
CFOBF	Clears terminal's output buffer
DEVST	Translates a device designator to a string
DIBE	Dismisses until terminal input buffer is empty
DOBE	Dismisses until terminal output buffer is empty
DTACH	Detaches controlling terminal from a job
DVCHR	Returns device characteristics
GDSKC	Returns disk usage
GDSTS	Returns the device status
GTTYP	Returns terminal type number
LPINI	Loads VFU or translation RAM
MSTR	Performs structure-dependent functions
MTOPR	Performs device-dependent functions
MTU%	Performs functions for logical tape devices
RELD	Releases a device
RFCOC	Returns control character output control words
RFMOD	Returns the JFN mode word
RFPOS	Returns current position of the terminal
SDSTS	Sets the device status
SFCOC	Sets control character output control words
SFMOD	Sets program-related fields in the JFN mode word
SFPOS	Sets position of the terminal's cursor
SIBE	Skips if input buffer is empty
SOBE	Skips if output buffer is empty
SOBF	Skips if output buffer is full
SPOOL	Defines and initializes input spooling
STDEV	Translates a string to a device designator
STPAR	Sets device-related fields in the JFN mode word
STTYP	Sets terminal type number
TLINK	Controls terminal linking

Software Interrupt System Functions

AIC	Activates interrupt channels
ATI	Assigns terminal code to channel
CIS	Clears the interrupt system
DEBRK	Dismisses current interrupt
DIC	Deactivates interrupt channels
DIR	Disables the interrupt system
DTI	Deassigns terminal code
EIR	Enables the interrupt system
GTRPW	Returns trap words
IIC	Initiates interrupts on specific channels in a process
RCM	Reads activated channel word mask
RIR	Reads the interrupt table addresses for a single-section program
RIRCM	Reads inferior reserved channel mask
RTIW	Reads terminal interrupt word
RWM	Reads waiting channel word mask

SCTTY	Changes source of terminal interrupts
SIR	Sets the interrupt table addresses for a single-section process
SIRCM	Sets inferior reserved channel mask
SKPIR	Skips if the interrupt system is enabled
STIW	Sets terminal interrupt word
XGTPW%	Returns page-fail words
XRIR%	Reads the interrupt table addresses for a multiple-section program
XSIR%	Sets the interrupt table addresses for a multiple-section process

Process/Capability Handling Functions

ADBRK	Controls address breaks
CFORK	Creates inferior process
DISMS	Dismisses process for specified amount of time
EPCAP	Enables process capabilities word
FFORK	Freezes one or more processes
GFRKH	Gets process handle
GFRKS	Gets current process structure
HALTF	Halts a process
HFORK	Halts an inferior process
KFORK	Kills one or more processes
PRARG	Sets or returns process argument block
RESET	Resets and initializes current process
RFACS	Returns process' accumulators
RFORK	Resumes one or more processes
RFRKH	Releases process handles
RFSTS	Returns process' status
RMAP	Obtains a handle on a page in a process
RPACS	Returns accessibility of page
RPCAP	Returns process capabilities word
RSMAP%	Returns information about the mapping of one section of a process
RTFRK	Returns the handle of a process suspended because of a monitor call intercept
RWSET	Releases working set
SFACS	Sets process' accumulators
SFORK	Starts a process in section zero
SPACS	Sets accessibility of page
SPLFK	Splices a process structure
TFORK	Sets and removes monitor call intercepts
UTFRK	Resumes a process suspended because of a monitor call intercept
WAIT	Dismisses process until interrupt occurs
WFORK	Waits for process to terminate
XSFRK%	Starts a process in a non-zero section

Save File Handling Functions

GCVEC	Gets compatibility package entry vector
GDVEC	Gets RMS entry vector
GET	Obtains a saved file
GEVEC	Gets process entry vector of a single-section program
SAVE	Saves a process as nonsharable
SCVEC	Sets compatibility package entry vector
SDVEC	Sets RMS entry vector
SEVEC	Sets the entry vector for a single-section program
SFRKV	Starts process using its entry vector
SSAVE	Saves a process as sharable
XGVEC%	Gets process entry vector for a multiple-section program
XSFRK%	Starts a process using a user-supplied, global PC
XSVEC%	Sets the entry vector for a multiple-section program

Date/Time Conversion Functions

GTAD	Gets current date and time in internal format
IDCNV	Converts from day, month, year to internal date and time
IDTIM	Inputs date and time, converting to internal format
IDTNC	Inputs date and time without converting to internal format
ODCNV	Converts from internal date and time to day, month, year
ODTIM	Outputs date and time, converting from internal format to text
ODTNC	Outputs date and time in internal format

Archive/Virtual Disk Functions

ARCF	Performs archive/virtual-disk operations
CRDIR	Creates or modifies a directory
DELDF	Expunges deleted files
DELNF	Retains specified number of generations of file
GTJFN	Assigns a JFN to a file
GNJFN	Assigns a JFN to the next file
JFNS	Translates a JFN to a string
OPENF	Opens a file
RFTAD	Reads file's time and dates
SETJB	Sets job parameters
SFTAD	Sets file's time and dates
SMON	Sets monitor flags
TMON	Reads monitor flags

Privileged Functions

NOTE: Calls marked with an asterisk (*) require privileges for specific functions only.

ACCES*	Accesses a directory
ALLOC	Allocates a device to a particular job
ARCF*	Performs archive/virtual-disk operations
ASNSQ	Assigns ARPANET special message queue
ATACH*	Attaches job to new controlling terminal
BOOT	Performs functions required for loading front-end software
CRDIR*	Creates or modifies a directory
CRJOB*	Creates a new job
DELDF*	Expunges deleted files
DELF*	Deletes files
DIAG	Reserves and releases hardware channels
DSKAS	Assigns specific disk addresses
DSKOP	Allows hardware address specification in disk transfers
ENQ*	Places a request in ENQ/DEQ resource queue
ENQC*	Returns status of a resource
FLHST	Flushes an ARPANET host
GACCT*	Returns job account information
GIVOK%	Allows/denies access to a protected system resource
GTDIR*	Returns directory information
HALTF*	Halts a process
HSYS	Halts the monitor
LGOUT*	Logs a job out
LPINI	Loads line-printer VFU
MDDT%	Enters MDDT program
MRECV*	Retrieves IPCF message
MSEND*	Sends IPCF message
MSFRK	Starts a process in monitor mode
MSTR*	Performs structure-related functions
MTALN	Associates magtape drive with logical unit number
MTOPR*	Performs device-related functions
MTU%	Performs MT-device functions
MUTIL*	Performs IPCF functions
NODE*	Performs DECnet functions
NTMAN%*	Performs DECnet network management functions
PEEK	Reads monitor data
PLOCK	Locks physical pages
PMCTL	Controls physical memory
RCVOK%	Services GETOK% requests
SETJB*	Sets job parameters
SFTAD*	Sets file date/time
SFUST*	Sets file author
SJPRI	Sets job priority
SKED%*	Manipulates scheduler data base
SMON	Sets monitor flags
SNOOP	Performs system performance analysis
SPOOL	Performs spooling-related functions
SPRIW	Sets process priority
STAD*	Sets system date/time

STI*	Simulates terminal input
SYERR	Places information in the System Error file
TTMSG*	Sends a message to a terminal
USAGE	Makes entries in accounting file
USRIO	Places program in user I/O mode
UTEST	Monitors executed instructions

TOPS-20 MONITOR CALLS

ACCES JSYS 552

FUNCTION
Gives a particular type of access to a given directory.

RESTRICTIONS
Requires WHEEL or OPERATOR capability for some functions.

CALLING SEQUENCE
```
AC1:  BO(AC%CON)   Connect job to directory
      B1(AC%OWN)   Give job owner access to directory
      B2(AC%REM)   Relinquish owner access to directory
      B18-35       Length of argblk
AC2:  Address of argblk
```

RETURNS +1: Always

ARGUMENT BLOCK

Word	Symbol	Meaning
0	.ACDIR	36-bit directory number or byte pointer to ASCIZ string containing full directory name
1	.ACPSW	Byte pointer to ASCIZ string containing password of specified directory
2	.ACJOB	Job # or -1 for current job (WHL/OPR if not -1)

ADBRK JSYS 570

FUNCTION
Controls address breaks.

RESTRICTIONS
Not available on KS-10 hardware.

CALLING SEQUENCE
```
AC1:  <function code>,,<process handle>
AC2:  Address of location at which to break  (.ABSET only)
AC3:  Flags  (.ABSET only)
      BO(AB%RED)     Break on read reference
      B1(AB%WRT)     Break on write reference
      B2(AB%XCT)     Break on execute reference
```

```
RETURNS  +1:  Always, with
              AC2:  Address of break location
              AC3:  Flags   (.ABRED only)
                    BO(AB%RED)   Break set for read
                    B1(AB%WRT)   Break set for write
                    B2(AB%XCT)   Break set for execute
```

FUNCTION CODES

Code	Symbol	Meaning
0	.ABSET	Set address break
1	.ABRED	Read address break
2	.ABCLR	Clear address break
3	.ABGAD	Return address of break instruction

AIC JSYS 131

FUNCTION
Activates specific software interrupt channels.

CALLING SEQUENCE
AC1: Process handle
AC2: 36-bit word (1B*n* activates channel *n*)

RETURNS +1: Always

ALLOC JSYS 520

FUNCTION
Allocates a device to a job or to the device pool.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability.

CALLING SEQUENCE
AC1: Function code (.ALCAL)
AC2: Device designator
AC3: Job # to allocate designated device,
-1 to deallocate designated device, or
-2 to assign device to monitor's resource allocator

RETURNS +1: Failure, error code in AC1
+2: Success

ARCF JSYS 247

FUNCTION
Performs archive and virtual disk operations.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability for some functions.

CALLING SEQUENCE
AC1: JFN
AC2: Function code
AC3: Function-specific argument

RETURNS +1: Always

FUNCTION CODES

Code	Symbol	Function
0	.ARRAR	Set/clear user request for archival AC3: 0(.ARCLR) to clear; 1(.ARSET) to set
1	.ARRIV	Set/clear system request for file migration AC3: 0(.ARCLR) to clear; 1(.ARSET) to set
2	.AREXM	Set/clear exemption from involuntary migration (enabled WHL/OPR) AC3: 0(.ARCLR) to clear; 1(.ARSET) to set
3	.ARRFR	Request that contents of file be restored to disk AC3: 1B0(AR%NMS) Don't send msg when restored 1B1(AR%WAT) Wait for file
4	.ARDIS	Discard tape information for file AC3: 1B0(AR%CR1) Clear run 1 information 1B1(AR%CR2) Clear run 2 information
5	.ARSST	Set tape information for file; (enabled WHL/OPR) AC3: Pointer to argblk
6	.ARRST	Restore file to disk; (enabled WHL/OPR) AC3: JFN for a DUMPER temporary file
7	.ARGST	Get tape information for file AC3: Pointer to argblk
10	.ARRFL	Retrieve for file failed; (WHL/OPR)
11	.ARNAR	Set/clear resist involuntary migration AC3: 0(.ARCLR) to clear; 1(.ARSET) to set

Argument Block for Functions .ARSST and .ARGST

Word	Symbol	Contents
0	.AROFL	Flags B0(AR%01) Set information for run 1 B1(AR%02) Set information for run 2 B2(AR%OFL) Delete content of disk file when done B3(AR%ARC) Archive the file B4(AR%CRQ) Clear archive and/or migration requests
1	.ARTP1	Tape 1 identification
2	.ARSF1	<tape 1 saveset number>,,<tape 1 file number>
3	.ARTP2	Tape 2 identification
4	.ARSF2	<tape 2 saveset number>,,<tape 2 file number>
5	.ARODT	Time/date of tape write in internal format
6	.ARPSZ	Number of pages in file

ASND JSYS 70

FUNCTION
Assigns a device to the caller

CALLING SEQUENCE
AC1: Device designator

RETURNS +1: Failure, error code in AC1
+2: Success

ASNSQ JSYS 752

FUNCTION
Assigns a special message queue to a job.

RESTRICTIONS
For ARPANET systems only; requires enabled NET WIZARD capability.

CALLING SEQUENCE
AC1: Mask
AC2: Header value

RETURNS +1: Failure, error code in AC1
+2: Success, special message queue assigned with queue handle in AC1

ATACH JSYS 116

FUNCTION
Detaches the specified job from its controlling terminal (if any) and optionally attaches it to a new controlling terminal.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability for some functions.

CALLING SEQUENCE
AC1: B0(AT%CCJ) Generate a CTRL/C interrupt to attached job
B1(AT%NAT) Do not attach job
B2(AT%TRM) Attach job to terminal specified in AC4
B18-35(AT%JOB) Job # of desired job
AC2: Logged-in user number of job to be attached
AC3: Byte pointer to ASCIZ password string
AC4: Number of terminal to be attached to specified job

RETURNS +1: Failure, error code in AC1
+2: Success

ATI JSYS 137

FUNCTION
Assigns a terminal code to a software interrupt channel.

CALLING SEQUENCE
AC1: <terminal interrupt code>,,<channel number>

RETURNS +1: Always

ATNVT JSYS 274

FUNCTION
Creates the Network Virtual Terminal (NVT) connection.

RESTRICTIONS
For ARPANET systems only

CALLING SEQUENCE
AC1: Flags,,<JFN of opened receive connection>
B2(AT%NTP) Indicates new (1) or old (0) TELNET protocol
AC2: JFN of opened send connection

RETURNS +1: Failure, with error code in AC1
+2: Success, with NVT-specific terminal designator in AC1

BIN JSYS 50

FUNCTION
Inputs the next byte from the specified source.

CALLING SEQUENCE
AC1: Source designator

RETURNS +1: Always, with the byte right-justified in AC2 or 0 indicating EOF

BKJFN JSYS 42

FUNCTION
Backs up the source designator's pointer by one byte.

RESTRICTIONS
Cannot be used with DECNET devices SRV: or DCN:.

CALLING SEQUENCE
AC1: Source designator

```
RETURNS    +1:  Failure, error code in AC1
           +2:  Success, updated byte pointer in AC1, if
                pertinent
```

BOOT JSYS 562

```
FUNCTION
Performs basic maintenance and utility functions required
for loading and dumping communications software.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability.

CALLING SEQUENCE
AC1:  Function code
AC2:  Address of argblk

RETURNS    +1:  Always

FUNCTION CODES
KS-10 Processor Functions
Code  Symbol    Meaning/Argblk
  0   .BTROM    Put line in MOP mode; activate front end
                bootstrap ROM
                0   .BTPRT   Line number
  1   .BTLDS    Load secondary bootstrap program into front
                end
                0   .BTPRT   Line number
                1            Not used; must be zero
                2   .BTSEC   Address of bootstrap program
  2   .BTLOD    Load front-end memory using previously loaded
                secondary or tertiary bootstrap program
                0   .BTDTE   Line number
                1            Not used; must be zero
                2            Not used; must be zero
                3            Not used; must be zero
                4   .BTCNT   Number of bytes to transfer
                5   .BTLPT   Pointer to data to be loaded
  4   .BTIPR    Generate and link DDCMP Station Table; start
                up lines/terminals not previously known to
                system
                0   .BTPRT   Drop,,<line number>
                1   .BTPRV   Protocol version number to use
  5   .BTTPR    Stop protocol currently running on front end
                or line
                0   .BTPRT   Line number
  6   .BTSTS    Return status type of protocol running on
                front end to specified DTE or line, and name
                of adjacent DECNET node for this front end
                0   .BTPRT   Line number
                1   .BTCOD   Returned protocol version type;
                             or -1 if no protocol is running
 10   .BTRMP    Read MOP message from front end using
                previously loaded secondary or tertiary
```

```
                bootstrap program
                0    .BTPRT    Line number
                1              Not used; must be zero
                2              Not used; must be zero
                3              Not used; must be zero
                4    .BTCNT    Number of bytes to transfer
                5    .BTMPT    Pointer to MOP message
                               destination
11    .BTKML    Load KMC11, or CRAM, DRAM, and 4 UNIBUS
                registers
                0    .BTKMC    KMC11 address
                1    .BTKER    <error flags>,,<bad data word>
                               (16 bit)
                               B0(BT%CVE)  CRAM verify error
                               B1(BT%DVE)  DRAM verify error
                               B2(BT%RVE)  Register verify
                                           error
                2    .BTKCC    Count of CRAM data
                3    .BTKCP    Pointer to CRAM data (16-bit)
                4    .BTKDC    Count of DRAM data
                5    .BTKDP    Pointer to DRAM data (8-bit)
                6    .BTKRC    Count of register data
                7    .BTKRP    Pointer to register data
                               (16-bit)
                8    .BTKSA    If 1B0, B18-35 contain start
                               address
                               B0(BT%KSA)  Right half >0; start
                                           KMC11
12    .BTKMD    Dump KMC11, or CRAM, DRAM, and registers if
                space provided
                0    .BTKMC    KMC11 address
                1              Not used; must be zero
                2    .BTKCC    Count of CRAM data
                3    .BTKCP    Pointer to CRAM data (16-bit)
                4    .BTKDC    Count of DRAM data
                5    .BTKDP    Pointer to DRAM data (8-bit)
                6    .BTKRC    Count of register data
                7    .BTKRP    Pointer to register data
                               (16-bit)
13    .BTRLC    Return line counters
                0    .BTPRT    Port number
                1    .BTZTM    Time since counters were last
                               zeroed
                2    .BTSCC    # of status counts to return
                3    .BTSCP    Pointer to area for status
                               counters
                4    .BTRCC    # of receive counts to return
                5    .BTRCP    Pointer to area for receive
                               counters
                6    .BTTCC    # of transmit counts to return
                7    .BTTCP    Pointer to area for transmit
                               counters
14    .BTCLI    Convert line ID to port number
                0    .BTPRT    Port number
15    .BTCPN    Convert NSP port number to line ID
                0    .BTPRT    Port number
```

```
16    .BTSTA   Set station's polling state to
               activate/deactivate terminal polling
               (Requires VT62)
               0    .BTPRT   Drop,,<line number>
               1    .BTCOD   Flags
                             0  .BTACT  Set line active
                             1  .BTIDL  Set line idle
17    .BTSSP   Set start-up priority value (Requires VT62)
               0    .BTPRT   Line number
               1    .BTSPR   Start priority count
20    .BTSTP   Set polling priority (Requires VT62)
               0    .BTPRT   Drop,,<line number>
               1    .BTPRI   Priority value: 1 (high) to 5
21    .BTSDD   Send a DDCMP message
               0    .BTPRT   Drop,,<line number>
               1    .BTMSG   Address of or byte pointer to
                             message
               2    .BTLEN   Byte count of message
22    .BTRDD   Receive a DDCMP message; .BTLEN is set to
               zero if queue is empty
               0    .BTPRT   Line number
               1    .BTMSG   Address of or byte pointer to
                             buffer
               2    .BTLEN   Size of user buffer
                             Returned in .BTLEN:
                             1B0(BT%CTL) +
                               .BTSUP (1) - station came up
                               .BTSDW (2) - station went down
                               .BTCMP (3) - transmit complete
                               .BTSSF (4) - start-up failed
23    .BTCHN   Set interrupt channel
               0    .BTPRT   Drop,,<line number>
               1    .BTCOD   Software interrupt channel
24    .BTSLS   Set type of line service for synchronous
               communications lines
               0    .BTPRT   Drop,,<line number>
               1    .BTCOD   Define protocol

KL-10 Processor Functions
Code  Symbol   Meaning/Argblk
 0    .BTROM   Put line in MOP mode; activate front end
               bootstrap ROM
               0    .BTDTE   DTE-20 number
               1    .BTERR   Error flags on failure (RET)
 1    .BTLDS   Load secondary bootstrap program into front
               end
               0    .BTDTE   DTE-20 number
               1    .BTERR   Error flags on failure (RET)
               2    .BTSEC   Address of bootstrap to load
 2    .BTLOD   Load front-end memory using previously loaded
               secondary or tertiary bootstrap program
               0    .BTDTE   DTE-20 number
               1    .BTERR   Error flags on failure (RET)
               2             Not used; must be zero
               3    .BTFLG   User-supplied flag word
                             B0(BT%BEL)  Send to -11 doorbell
```

```
                                        after setup
            4    .BTCNT    Number of bytes to transfer
            5    .BTLPT    Pointer to data to be loaded
 3   .BTDMP  Dump front-end memory using ROM bootstrap
            program
            0    .BTDTE    DTE-20 number
            1    .BTERR    Error flags on failure (RET)
            2              Not used; must be zero
            3              Not used; must be zero
            4    .BTCNT    Number of bytes to transfer
            5    .BTDPT    Pointer to dump data destination
 4   .BTIPR  Initialize front end protocol
            0    .BTDTE    DTE-20 number
            1    .BTPRV    Protocol version number to use
 5   .BTTPR  Stop protocol currently running on front end
            or line
            0    .BTDTE    DTE-20 number
 6   .BTSTS  Return status type of protocol running on
            front end to specified DTE or line, and name
            of adjacent DECNET node for front end
            0    .BTDTE    DTE-20 number
            1    .BTCOD    Returned protocol version type;
                           or -1 if no protocol is running
 7   .BTBEL  Block until signal to TOPS-20 is initiated by
            front end
            0    .BTDTE    DTE-20 number
10   .BTRMP  Read data from front end using previously
            loaded secondary or tertiary bootstrap
            program
            0    .BTDTE    DTE-20 number
            1    .BTERR    Error flags on failure (RET)
            2              Not used; must be zero
            3    .BTFLG    User-supplied flag word
                           BO(BT%BEL)  Send doorbell after
                                       transfer
            4    .BTCNT    Maximum # of bytes to transfer
            5    .BTMPT    Pointer to data destination
14   .BTCLI  Convert line ID to port number
            1    .BTLID    Pointer to ASCIZ line ID
15   .BTCPN  Convert NSP port number to line ID
            1    .BTLID    Pointer to ASCIZ line ID
16   .BTD6O  Send message to or receive message from DN6O
            front end using .VND6O protocol (Requires
            DN6O on KL-10 Model B)
             0   .BT6DTE   DTE number
             1   .BT6ERR   Error flags (RET)
                           30  D6%BDP  Byte pointer is bad
                           31  D6%ARD  -11 attempted to
                                       send data
                           32  D6%TRS  DTESRV timed out
                                       waiting for response
                                       header from -11
                           33  D6%TDT  DTESRV timed out
                                       waiting for data
                                       from -11
                           34  D6%TPO  DTESRV timed out
```

			waiting for DTE to be free
		35 D6%NT6	-11 is not running DN60 protocol
2	.BT6HBC	B0-17	DN60 header byte count
	.BT6HDR	B18-35	DN60 header address
3	.BT6DBC	Number of bytes of data	
4	.BT6PTR	Pointer to first byte of data	
5	.BT6TMR	Time request was made (RET)	
6	.BT6TAS	Time DTE was assigned (RET)	
7	.BT6THQ	Time TOPS-20 queued header to DTE (RET)	
10	.BT6TRD	Time	
11	.BT6TDD	Time	
12	.BT6TFR	Time TOPS-20 satisfied request	

BOUT JSYS 51

FUNCTION
Outputs a byte sequentially to the specified destination.

CALLING SEQUENCE
AC1: Destination designator
AC2: Byte to be output, right-justified

RETURNS +1: Always

CACCT JSYS 4

FUNCTION
Changes the account for the current job.

RESTRICTIONS
In non-zero sections, OWGBPs must specify 7-bit bytes.

CALLING SEQUENCE
AC1: Byte pointer to the new account string;
in section 0, may contain <5B2+<account number>B35>

RETURNS +1: Failure, error code in AC1
+2: Success, updated byte pointer in AC1

CFIBF JSYS 100

FUNCTION
Clears the designated file input buffer.

CALLING SEQUENCE
AC1: Source designator

CFIBF

RETURNS +1: Always

CFOBF JSYS 101

FUNCTION
Clears the designated file output buffer

CALLING SEQUENCE
AC1: Destination designator

RETURNS +1: Always

CFORK JSYS 152

FUNCTION
Creates a process inferior to the calling process.

CALLING SEQUENCE

AC1:	B0(CR%MAP)	Make inferior process' map same as current process' map
	B1(CR%CAP)	Make inferior process' capabilities same as current process'
	B3(CR%ACS)	Set inferior process' ACs from block whose address is in AC2
	B4(CR%ST)	Set PC of inferior process to value in B18-35 of AC1 and start process
	B18-35(CR%PCV)	PC value for inferior process if CR%ST is on
AC2:	Address of optional 20 word block containing AC values for inferior process	

RETURNS +1: Failure, error code in AC1
+2: Success, relative process handle in AC1

CHFDB JSYS 64

FUNCTION
Changes words in the File Descriptor Block for the specified file.

CALLING SEQUENCE

AC1:	B0(CF%NUD)	Don't wait for disk copy of directory to be updated
	B9-17(CF%DSP)	Index into FDB of word to be changed
	B18-35(CF%JFN)	JFN for a disk file
AC2:	Mask indicating bits to be changed; -1 if changing a count value in AC3	
AC3:	New values for changed bits corresponding to mask given in AC2	

RETURNS +1: Always

CHKAC JSYS 521

FUNCTION
Checks if a user is allowed access to files in a given directory.

RESTRICTIONS
In non-zero sections, OWGBPs must specify 7-bit bytes.

CALLING SEQUENCE
AC1: Flags,,<length of argblk>
BO(CK%JFN) JFN in word .CKAUD of the argblk
AC2: Address of argblk

RETURNS +1: Failure, error code in AC1
+2: Success, access check is completed, with AC1 containing -1 if access is allowed or 0 if access is not allowed

ARGUMENT BLOCK

Word	Symbol	Meaning
0	.CKAAC	Code of desired access to files
1	.CKALD	Byte pointer to username string, or 36-bit user number
2	.CKACD	Byte pointer to directory name string, or 36-bit directory number of user's connected directory
3	.CKAEC	Enabled capabilities of user
4	.CKAUD	Byte pointer to directory name string, or 36-bit directory number of directory being accessed; if BO(CK%JFN) is on, contains JFN for file being accessed
5	.CKAPR	Protection of files being accessed; (not required if a JFN is supplied in word .CKAUD)

ACCESS CODES

Code	Symbol	Meaning
0	.CKARD	Read existing files
1	.CKAWR	Write existing files
2	.CKAEX	Execute existing files
3	.CKAAP	Append to existing files
4	.CKADL	Obtain directory listing of existing files
6	.CKADR	Read the directory
10	.CKACN	Connect to the directory
11	.CKACF	Create files in the directory

CIS JSYS 141

FUNCTION
Clears the software interrupt system for the current process.

RETURNS +1: Always

CLOSF JSYS 22

FUNCTION
Closes a specific file or all files.

CALLING SEQUENCE

AC1:	B0(CO%NRJ)	Do not release the JFN
	B6(CZ%ABT)	Abort any output operations currently being done
	B7(CZ%NUD)	Do not update copy of directory on disk
	B18-35(CO%JFN)	JFN of file being closed

RETURNS +1: Failure, error code in AC1
+2: Success

CLZFF JSYS 34

FUNCTION
Closes all files and/or releases all JFNs at or below a specified process.

CALLING SEQUENCE

AC1:	B0(CZ%NIF)	Do not close files of inferior processes
	B1(CZ%NSF)	Do not close files of this process
	B2(CZ%NRJ)	Do not release JFNs
	B3(CZ%NCL)	Do not close any files; only release nonopen JFNs
	B4(CZ%UNR)	Unrestrict files opened with restricted access for specified process
	B5(CZ%ARJ)	Wait until file can be closed, close it, and release JFNs
	B6(CZ%ABT)	Abort any output operations currently being done
	B7(CZ%NUD)	Do not update copy of directory on disk
	B18-35(CZ%PRH)	Process handle

RETURNS +1: Always

COMND JSYS 544

FUNCTION
Parses one or more fields of a command that is either typed by a user or contained in a file.

CALLING SEQUENCE
AC1: Address of the command state block
AC2: Address of first alternate function descriptor block

RETURNS +1: Always (unless a reparse is needed and the right half of .CMFLG is nonzero), with
AC1: Flags,,<address of command state block>
AC2: Data obtained for field; or error code if field could not be parsed (CM%NOP is on)
AC3: B0-17 Address of function descriptor block given
B18-35 Address of function descriptor block used

COMMAND STATE BLOCK

Word	Symbol	Meaning
0	.CMFLG	<flag bits>,,<reparse dispatch address>
1	.CMIOJ	<input JFN>,,<output JFN>
2	.CMRTY	Byte pointer to beginning of the prompting text
3	.CMBFP	Byte pointer to beginning of the user's input
4	.CMPTR	Byte pointer to beginning of next field to be parsed
5	.CMCNT	Count of space remaining in buffer after .CMPTR pointer
6	.CMINC	Count of number of unparsed characters in buffer after .CMPTR pointer
7	.CMABP	Byte pointer to atom buffer containing last field parsed by COMND
10	.CMABC	Size of atom buffer in bytes
11	.CMGJB	Address of 16 word, writable GTJFN argblk

Settable Bits in Word .CMFLG of the Command State Block

Bit	Symbol	Meaning
6	CM%RAI	Convert lowercase input to uppercase
7	CM%XIF	"@" is punctuation, not indirect file designator
8	CM%WKF	Begin parsing after each field is terminated without waiting for action character (CRLF, ESC, CTRL/F, ?)

FUNCTION DESCRIPTOR BLOCK

Word	Symbol	Meaning
0	.CMFNP	Function code and pointer to next function descriptor block
		B0-8(CM%FNC) Function code
		B9-17(CM%FFL) Function-specific flags
		B18-35(CM%LST) Address of next function descriptor block; or 0

		if last
1	.CMDAT	Data for the specific function, if any
2	.CMHLP	Byte pointer to help text for this field
3	.CMDEF	Byte pointer to default string for this field
4	.CMBRK	Address of 4-word break mask that specifies which characters terminate a field

FUNCTIONS FOR WORD .CMFNP OF THE FUNCTION DESCRIPTOR BLOCK

Code	Symbol	Meaning
0	.CMKEY	Parse a keyword
1	.CMNUM	Parse a number
2	.CMNOI	Parse a guide word string
3	.CMSWI	Parse a switch
4	.CMIFI	Parse an input filespec
5	.CMOFI	Parse an output filespec
6	.CMFIL	Parse a general (arbitrary) filespec
7	.CMFLD	Parse an arbitrary field
10	.CMCFM	Wait for user to confirm command with CRLF
11	.CMDIR	Parse a directory name
12	.CMUSR	Parse a user name
13	.CMCMA	Parse a comma
14	.CMINI	Initialize the command line
15	.CMFLT	Parse a floating-point number
16	.CMDEV	Parse a device name
17	.CMTXT	Parse input text up to next carriage return, place text in atom buffer, and return
20	.CMTAD	Parse a date and/or time field according to setting of bits CM%IDA and CM%ITM
21	.CMQST	Parse a quoted string up to terminating quote
22	.CMUQS	Parse an unquoted string up to one of the specified break characters
23	.CMTOK	Parse input and compare it with a given string
24	.CMNUX	Parse a number and terminate on 1st nonnumeric character
25	.CMACT	Parse an account string
26	.CMNOD	Parse a network node name

Function-specific Flags in B9-B17 (CM%FFL) of Word .CMFNP

Bit	Symbol	Meaning
12	CM%NSF	Suffix is optional; functions .CMDEV and .CMNOD only
13	CM%BRK	Word .CMBRK of function descriptor block contains a pointer to a 4-word break mask
14	CM%PO	Field is parse only (no existence verification); functions .CMDEV, .CMDIR, .CMNOD, and .CMUSR only
15	CM%HPP	Byte pointer to program-supplied help message for field is in word 2 (.CMHLP) of function descriptor block
16	CM%DPP	Byte pointer to program-supplied default string for field is in word 3 (.CMDEF) of function descriptor block
17	CM%SDH	Suppress output of default help message if user types a question mark

ADDITIONAL DATA IN WORD .CMDAT OF THE FUNCTION DESCRIPTOR BLOCK

Function	Contents of Word .CMDAT
.CMKEY	Address of keyword symbol table whose entries point to argblks; B18-35 of Word 0 of argblk may contain flags: B33(CM%ABR) Keyword is abbreviation B34(CM%NOR) Do not recognize keyword B35(CM%INV) Make keyword invisible
.CMNUM	Radix of the number (from 2 to 10)
.CMNOI	Byte pointer to an ASCIZ string that contains the guide word
.CMSWI	Address of switch keyword table, whose entries point to argblks; B18-35 of Word 0 of argblk may contain flags: B33(CM%ABR) Keyword is abbreviation B34(CM%NOR) Do not recognize keyword B35(CM%INV) Make keyword invisible
.CMDIR	Data bits B0(CM%DWC) Allow wildcard characters in directory name
.CMTAD	<flag bits>,,<address of 3-word block> B0(CM%IDA) Parse a date B1(CM%ITM) Parse a time B2(CM%NCI) Do not convert date/time to internal format
.CMUQS	Address of 4-word block of 128. break character mask bits
.CMTOK	Byte pointer to the given string
.CMNUX	The radix (from 2 to 10) of the number

DEFAULT HELP MESSAGES

Function	Message
.CMKEY (keyword)	ONE OF THE FOLLOWING if no keyword matches the currently typed field KEYWORD (NO DEFINED KEYWORDS MATCH THIS INPUT)
.CMNUM (number)	OCTAL NUMBER (radix 8) DECIMAL NUMBER (radix 10) A NUMBER IN BASE nn (radix nn)
.CMNOI (guide word)	None
.CMSWI (switch)	ONE OF THE FOLLOWING
.CMIFI (input file) }	Depending on flag settings for
.CMOFI (output file) }	GTJFN call, OUTPUT FILESPEC or
.CMFIL (any file) }	INPUT FILESPEC
.CMFLD (any field)	None
.CMCFM (confirm)	CONFIRM WITH CARRIAGE RETURN
.CMDIR (directory)	DIRECTORY NAME
.CMUSR (user)	USER NAME
.CMCMA (comma)	COMMA
.CMINI (initialize)	None
.CMFLT (floating point)	NUMBER
.CMDEV (device)	DEVICE NAME
.CMTXT (text)	TEXT STRING
.CMTAD (date)	Depending on bits set in .CMDAT,

	DATE, TIME, or DATE AND TIME
.CMQST (quoted)	QUOTED STRING
.CMUQS (unquoted)	UNQUOTED STRING if "?" is a break character
.CMTOK (token)	None
.CMNUX (number)	Same as .CMNUM
.CMACT (account)	None
.CMNOD (node)	NODE NAME

Functions That Use Masks (Word .CMBRK)

Mask Symbol	Function	Changeable by User
KEYBO. - KEYB3.	.CMKEY	Yes
DEVBO. - DEVB3.	.CMDEV	Yes (if parse-only)
FLDBO. - FLDB3.	.CMFLD	Yes
EOLBO. - EOLB3.	.CMTXT	Yes
KEYBO. - KEYB3.	.CMSWI	Yes
User-specified	.CMDAT	Yes
USRBO. - USRB3.	.CMUSR	No
FILBO. - FILB3.	.CMFIL	No
FILBO. - FILB3.	.CMIFI	No
FILBO. - FILB3.	.CMOFI	No
internal	.CMNUM	No
FILBO. - FILB3.	.CMDIR	No
internal	.CMFLT	No
ACTBO. - ACTB3.	.CMACT	No

RETURNED BITS IN WORD .CMFLG OF THE FUNCTION DESCRIPTOR BLOCK

Bit	Symbol	Meaning
0	CM%ESC	ESC was typed by user as terminator for this field
1	CM%NOP	Field could not be parsed because it did not conform to specified function(s)
2	CM%EOC	Field was terminated with a carriage return
3	CM%RPT	Characters already parsed need to be reparsed because user edited them
4	CM%SWT	Switch field was terminated with a colon
5	CM%PFE	Previous field was terminated with an ESC

CRDIR JSYS 240

FUNCTION
Creates, changes, or deletes a directory entry.

RESTRICTIONS
Enabled WHEEL or OPERATOR capability required for some functions.

CALLING SEQUENCE

AC1:	Byte pointer to ASCIZ string containing str:<directory>	
AC2:	BO(CD%LEN)	Set flags and length of argblk from values in word .CDLEN
	B1(CD%PSW)	Set password from argblk

```
        B2(CD%LIQ)      Set working disk storage limit from
                        argblk
        B3(CD%PRV)      Set capability bits from argblk
        B4(CD%MOD)      Set mode bits from argblk
        B5(CD%LOQ)      Set permanent disk storage limit from
                        argblk
        B6(CD%NUM)      Set directory number from argblk
        B7(CD%FPT)      Set default file protection from argblk
        B8(CD%DPT)      Set directory protection from argblk
        B9(CD%RET)      Set default retention count from argblk
        B10(CD%LLD)     Set last LOGIN date from argblk
        B11(CD%UGP)     Set user groups from argblk
        B12(CD%DGP)     Set directory groups from argblk
        B13(CD%SDQ)     Set subdirectory quota from argblk
        B14(CD%CUG)     Set user groups assignable by directory
                        from argblk
        B15(CD%DAC)     Set default account from argblk
        B17(CD%DEL)     Delete this directory entry
        B18-35(CD%APB) Address of the argblk
AC3:    Byte pointer to ASCIZ string containing password of
        directory

RETURNS    +1:  Always, with directory number in AC1

ARGUMENT BLOCK
Word   Symbol    Meaning
  0    .CDLEN    <flag bits>,,<length of argblk>
                 BO(CD%NSQ)    On restore, do not update
                               superior directory's quotas
                               (enabled WHL/OPR required)
                 B1 (CD%NCE)   On restore or reconstruction, do
                               not change directory parameters
                               if directory currently exists
                               (enabled WHL/OPR)
                 B2(CD%NED)    Set default on-line expiration
                               date from word .CDDNE
                 B3(CD%FED)    Set default on-line expiration
                               date from word .CDDFE
  1    .CDPSW    Byte pointer to password string
  2    .CDLIQ    Working disk storage quota
  3    .CDPRV    Capabilities for this user
  4    .CDMOD    Mode word
                 BO(CD%DIR)    Directory is files-only
                 B1(CD%ANA)    Obsolete
                 B2(CD%RLM)    Repeat messages from file
                               <SYSTEM>MAIL.TXT each time user
                               logs in
                 B7(CD%DAR)    File should be archived rather
                               than migrated when on-line
                               expiration date reached
  5    .CDLOQ    Permanent disk storage quota
  6    .CDNUM    Directory number (valid only when creating a
                 directory)
  7    .CDFPT    Default file protection (18 bits, R-J)
 10    .CDDPT    Directory protection (18 bits, R-J)
 11    .CDRET    Default generation retention count
```

12	.CDLLD	Date of last login
13	.CDUGP	Address of user group list for this directory
14	.CDDGP	Address of directory group list
15	.CDSDQ	Maximum number of sub-directories allowed
16	.CDCUG	Address of user group list
17	.CDDAC	Byte pointer to default account string
20	.CDDNE	Default on-line expiration date and time
21	.CDDFE	Default off-line expiration date and time

DEFAULT ARGUMENTS

Bit	Symbol	Default Argument
2	CD%LIQ	250 working pages
3	CD%PRV	No special capabilities
4	CD%MOD	Directory name for login
5	CD%LOQ	250 permanent pages
6	CD%NUM	First unused directory number
7	CD%FPT	Default file protection to 777700
8	CD%DPT	Directory protection to 777700
9	CD%RET	Default file retention count to 1
10	CD%LLD	Never logged in
11	CD%UGP	No user groups
12	CD%DGP	No directory groups
13	CD%SDQ	No ability to create inferior directories
14	CD%CUG	No assignable user groups for inferior directories
15	CD%DAC	No default account

CRJOB JSYS 2

FUNCTION
Creates a new job and optionally logs it in.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability for some functions; in non-zero sections, OWGBPs must specify 7-bit bytes.

CALLING SEQUENCE
AC1: <flag bits>,,0
AC2: Address of argblk
AC3: Job # of previously-created job if B17(CJ%DSN) is on in AC1

RETURNS +1: Failure, with error code in AC1
+2: Success, with number of new job in AC1

Flag Bits in AC1

Bit	Symbol	Meaning
0	CJ%LOG	Log in the new job
1	CJ%NAM	Set user name and password from argblk
2-3	CJ%ACT	Account code for new job
	0 .CJUCA	Use current account of caller
	1 .CJUAA	Use account from the argblk
	2 .CJUDA	Use default account of caller

4	CJ%ETF	Place TOPS-20 command processor in top process of new job
5	CJ%FIL	Move the file pointed to by word .CJFIL of the argblk into a process in new job
6	CJ%ACS	Load ACs from the address in argument block; loaded only if the program being run is not the command processor
7	CJ%OWN	Maintain ownership of the new job
8	CJ%WTA	Do not start new job until it is attached to a terminal
9	CJ%NPW	Do not check password given when new job is logged in
10	CJ%NUD	Do not update LOGIN date for user logging in to new job
11	CJ%SPJ	Set primary I/O designators from argblk before starting job
12	CJ%CAP	Set allowed capabilities of new job to be same as caller's currently enabled capabilities, until new job is logged in
13	CJ%CAM	Set allowed capabilities of new job to combination and function of capability mask in argblk and new job's user capabilities
14	CJ%SLO	Send IPCF message to PID supplied in argblk when new job is logged out
17	CJ%DSN	Release ownership of previously created job whose number is in AC3; if on, overrides all other bits set in AC1

ARGUMENT BLOCK

Word	Symbol	Meaning
0	.CJNAM	Byte pointer to ASCIZ user name string
1	.CJPSW	Byte pointer to ASCIZ password string
2	.CJACT	5B2 + account number or byte pointer to account string
3	.CJFIL	Byte pointer to name of file to be moved into a process of new job
4	.CJSFV	Offset in entry vector to use as start address of the file to which word .CJFIL points
5	.CJTTY	TTY designator of new job's controlling terminal
6	.CJTIM	Connect-time for new job before LGOUT is forced on it; 0 indicates no limit
7	.CJACS	Address of 16-word block to be loaded in new job's ACs if program other than Command Processor is being run
10	.CJEXF	Flag bits to be passed to Command Processor in top-level process of new job B0 Suppress herald printed by Command Processor B1 Move file to which word .CJFIL points into process whose handle is in PRARG block B2 Start process at offset in entry vector given in word .CJSFV after Command

		Processor is initialized
	B3	Output text printed when LOGIN command is given
11	.CJPRI	Primary input and output device designators for the inferior processes of the new job
12	.CFCPU	Run-time limit for new job
13	.CJCAM	Capability mask for new job; used only if CJ%CAM is set
14	.CJSLO	PID to which IPCF message is to be sent when new job is logged out

Format of IPCF Logout Message

Word	Contents
0	0,,.IPCLO
1	<count of remaining words>,,<# of job logged out>
2	Flags,,reserved
	0 SP%BAT Job is controlled by batch
	1 SP%DFS Spooling is deferred
	2 SP%ELO Job executed LGOUT
	3 SP%FLO Job was forced to logout
	4 SP%OLO Job was logged out by another job
3	Job connect time
4	Job CPU time
5	TTY number of job at logout (-1 if detached)
6	Job # of job that did logout
7	Reserved
10	Most recent monitor call error code

CRLNM JSYS 502

FUNCTION
Defines or deletes a logical name assignment.

CALLING SEQUENCE
AC1: Function code
AC2: Byte pointer to the logical name
AC3: Byte pointer to the logical name definition string

RETURNS +1: Failure, error code in AC1
+2: Success, updated byte pointer in AC3

FUNCTION CODES

Code	Symbol	Meaning
0	.CLNJ1	Delete one logical name from the job
1	.CLNS1	Delete one logical name from the system
2	.CLNJA	Delete all logical names from the job
3	.CLNSA	Delete all logical names from the system
4	.CLNJB	Create a logical name for the job
5	.CLNSY	Create a logical name for the system

CVHST JSYS 276

FUNCTION
Converts a host number to a primary name.

RESTRICTIONS
For use with ARPANET systems only.

CALLING SEQUENCE
AC1: Destination for ASCIZ host name string
AC2: Host number

RETURNS +1: Failure
+2: Success

CVSKT JSYS 275

FUNCTION
Converts a local socket number to absolute form.

RESTRICTIONS
For use with ARPANET systems only.

CALLING SEQUENCE
AC1: JFN

RETURNS +1: Failure, error code in AC1
+2: Success, absolute socket number in AC2

DEBRK JSYS 136

FUNCTION
Dismisses the software interrupt routine in progress and resumes the process at the location specified by the PC stored in the priority level table.

RETURNS +1: Only if no software interrupt is currently in progress and if an ERJMP or ERCAL instruction follows the DEBRK

DELDF JSYS 67

FUNCTION
Reclaims disk space by expunging deleted disk files.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability for some functions.

CALLING SEQUENCE
AC1: B0(DD%DTF) Delete temporary files (;T) also
B1(DD%DNF) Delete nonexistent files that are not now open
B2(DD%RST) Rebuild the symbol table
B3(DD%CHK) Check internal consistency of directory
AC2: Directory number

RETURNS +1: Always

DELF JSYS 26

FUNCTION
Deletes specified disk file and, if the file is closed, releases the JFN.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability for some functions.

CALLING SEQUENCE
AC1: B0(DF%NRJ) Do not release the JFN
B1(DF%EXP) Expunge file and delete FDB entry in directory
B2(DF%FGT) Expunge file but do not deassign its addresses; (enabled WHL/OPR)
B3(DF%DIR) Delete and expunge a directory file; (enabled WHL/OPR)
B4(DF%ARC) Allow a file with archive status to be deleted
B5(DF%CNO) Delete and expunge file but preserve filename and FDB (except for page count and page table address)
B18-35(DF%JFN) JFN of the file being deleted

RETURNS +1: Failure, error code in AC1
+2: Success, JFN is released unless B0(DF%NRJ) is on or file is open

DELNF JSYS 317

FUNCTION
Marks for deletion all but the specified number of generations of a disk file.

CALLING SEQUENCE
AC1: B0(DF%NRJ) Do not release the JFN
B4(DF%ARC) Allow a file with archive status to be deleted
B5(DF%CNO) Delete and expunge file but preserve filename and FDB (except for page count and page table address)

```
      B18-35       JFN of the file being deleted
AC2:  Number of generations to retain

RETURNS    +1:  Failure, error code in AC1
           +2:  Success, with number of files deleted in AC2
```

DEQ JSYS 514

FUNCTION
Removes a request for a specific resource from the queue associated with that resource.

RESTRICTIONS
In non-zero sections, OWGBPs must specify 7-bit bytes.

CALLING SEQUENCE
AC1: Function code
AC2: Address of argblk

RETURNS +1: Failure, error code in AC1
+2: Success

FUNCTION CODES

Code	Symbol	Meaning
0	.DEQDR	Remove specified requests from queue; requires argblk
1	.DEQDA	Remove all requests for this process from queue
2	.DEQID	Remove all requests corresponding to specified request ID

ARGUMENT BLOCK FOR FUNCTION .DEQDR

Word	Symbol	Meaning
0	.ENQLN	B0-5 Header length B6-17 # of locks B18-35 Length of argblk
1	.ENQID	Not used; must be zero
2	.ENQLV	<flags & level number>,,[JFN][-1][-2][-3] B0(EN%SHR) Access to this resource is to be shared B1(EN%BLN) Ignore level number of resource B2(EN%NST) Allow ownership of this lock to be nested B3(EN%LTL) Allow a long-term lock on this resource B9-17(EN%LVL) Level number associated with this resource B18-35 JFN Associated file has standard protection; or -1 Resource accessible only by processes in job; or -2 Resource accessible by any job on system; or

		-3 Resource accessible only by enabled WHL/OPR processes
3	.ENQUC	Pointer to string; or 5B2+33-bit user code
4	.ENQRS	<# of resources in pool>,,<# requested>; or 0,,<group number> if only one resource of specific type
5	.ENQMS	Address of a resource mask block
n-4		<flags & level number>,,[JFN][-1][-2][-3]
n-3		Pointer to string; or 5B2+33-bit user code
n-2		<# of resources in pool>,,<# requested> or 0,,<group number>
n-1		Address of a resource mask block

DEVST JSYS 121

FUNCTION
Translates the given device designator to its corresponding ASCIZ device name string (excluding colon).

CALLING SEQUENCE
AC1: Destination designator
AC2: Device designator

RETURNS +1: Failure, error code in AC1
+2: Success, updated byte pointer in AC1, if pertinent

DFIN JSYS 234

FUNCTION
Inputs a double-precision, floating-point number, rounding if necessary.

CALLING SEQUENCE
AC1: Source designator

RETURNS +1: Failure, error code in AC4 and updated string pointer in AC1, if pertinent
+2: Success, double-precision, floating-point number in AC2 and AC3 and updated byte pointer in AC1, if pertinent

DFOUT JSYS 235

FUNCTION
Outputs a double-precision, floating-point number.

CALLING SEQUENCE
AC1: Destination designator
AC2: 1st word of a normalized, double-precision,

```
      floating-point number
AC3:  2nd word of a normalized, double-precision,
      floating-point number
AC4:  Format control word

RETURNS   +1:  Failure, error code in AC4 and updated string
               pointer in AC1, if pertinent.
          +2:  Success, updated byte pointer in AC1, if
               pertinent
```

DIAG JSYS 530

FUNCTION
Reserves/releases a channel and either a single device or all devices attached to that channel.

RESTRICTIONS
Requires enabled WHEEL, OPERATOR, or MAINTENANCE capability.

CALLING SEQUENCE
AC1: -<length of argblk>,,<address of argblk>

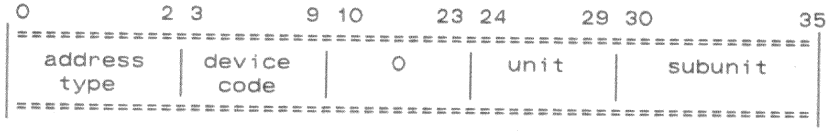
RETURNS +1: Failure, error code in AC1
+2: Success

FUNCTION CODES

Code	Symbol	Meaning/Argblk
1	.DGACU	Assign channel and a single device; release device after time limit specified
		0 Function code
		1 Device address
		2 Time limit in mss
2	.DGACH	Assign the channel and all devices
		0 Function code
		1 Device address
3	.DGRCH	Release channel and all assigned devices
		0 Function code
		1 Device address
4	.DGSCP	Set up channel program
		0 Function code
		1 Device address
		2 Channel control word 0
		3 Channel control word 1
		n+2 Channel control word n
5	.DGRCP	Release channel program
		0 Function code
		1 Device address
6	.DGGCS	Return status of channel
		0 Function code
		1 Device address
		2-5 4-word channel logout area
100	.DGGEM	Get memory (for TGHA)
		0 Function code
		1 1st page in user address space

```
                2   1st physical memory page
                3   Number of pages
                4   User address of AR/ARX parity trap
                    routines
101    .DGREM   Release memory (for TGHA)
                0   Function code
102    .DGPDL   Inform the monitor that a device previously
                unknown is now online
                0   Function code
                1   Primary channel number
                2   Primary unit number
                3   Primary controller number; -1 if no
                    controller
                4   Alternate channel number
                5   Alternate unit number
                6   Alternate controller number; -1 if no
                    controller
```

DEVICE ADDRESS WORD



DIBE JSYS 212

FUNCTION
Dismisses the process until the designated file input buffer is empty.

CALLING SEQUENCE
AC1: File designator

RETURNS +1: Always

DIC JSYS 133

FUNCTION
Deactivates the specified software interrupt channels.

AC1: Process handle
AC2: 36-bit word (1B*n* means deactivate channel *n*)

RETURNS +1: Always

DIR JSYS 130

FUNCTION
Disables the software interrupt system for a process.

CALLING SEQUENCE
AC1: Process handle

RETURNS +1: Always

DIRST JSYS 41

FUNCTION
Translates the specified 36-bit user or directory number to its corresponding string and writes it to the given destination.

CALLING SEQUENCE
AC1: Destination designator
AC2: User or directory number

RETURNS +1: Failure, with error code in AC1.
+2: Success, string written to destination, updated string pointer, if pertinent, in AC1

DISMS JSYS 167

FUNCTION
Dismisses this process for the specified amount of time.

CALLING SEQUENCE
AC1: Number of mss. for which the process is to be dismissed

RETURNS +1: When the elapsed time is up

DOBE JSYS 104

FUNCTION
Dismisses the process until the designated file output buffer is empty.

CALLING SEQUENCE
AC1: Destination designator

RETURNS +1: Always

DSKAS JSYS 244

FUNCTION
Assigns or deassigns specific disk addresses.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability.

CALLING SEQUENCE

AC1:	B0(DA%DEA)	Deassign specified address
	B1(DA%ASF)	Assign a free page near specified address
	B2(DA%CNV)	Convert specified address according to setting of B3(DA%HWA)
	B3(DA%HWA)	Specified address is a hardware address
	B4(DA%INI)	Initialize a private copy of bit table
	B5(DA%WRT)	Write private copy of bit table to new bit table file
	B18-35(DA%ADR)	Disk address
AC2:	Device designator of structure; not required if DA%CNV is on in AC1	

RETURNS +1: Failure, address already assigned or cannot be assigned
+2: Success, address assigned in AC1

DSKOP JSYS 242

FUNCTION
Allows the process to reference physical disk addresses when performing disk transfers.

RESTRICTIONS
Requires WHEEL, OPERATOR, or MAINTENANCE capabilities enabled.

CALLING SEQUENCE

AC1:	B0-1(DOP%AT)	Field indicating the address type 1(.DOPPU) for physical channel/unit addresses, with B2-6(DOP%CN) channel number B7-12(DOP%UN) unit number B13-35(DOP%UA) unit address 2(.DOPSR) for structure or relative addresses, with B2-10(DOP%SN) structure designator 0 PS: -1 structure designator in AC4 B11-35(DOP%RA) relative address
AC2:	<control flags>,,<number of words to transfer>	
	B9(DOP%NF)	Use channel/controller/unit #s in AC4
	B10(DOP%EO)	Error if unit off-line
	B11(DOP%IL)	Inhibit error logging

```
        B12(DOP%IR)     Inhibit error recovery
        B14(DOP%WR)     Write data to disk; if off, read data
                        from disk
        B18-35(DOP%CT)  Word count
AC3:    Address in caller's address space from which data is
        read or into which data is written
AC4:    Device designator of structure if -1 in DOP%SN;
        physical channel, controller, and unit numbers if
        1B9(DOP%NF) with
        B0-11(DOP%C2)   Channel number
        B12-23(DOP%K2)  Controller number
        B13-35(DOP%U2)  Unit number

RETURNS    +1:  Always, AC1 is nonzero if an error occurred,
                or zero if no error occurred.
```

DTACH JSYS 115

```
FUNCTION
Detaches the controlling terminal from the current job.

RETURNS    +1:  Always
```

DTI JSYS 140

```
FUNCTION
Deassigns a terminal interrupt code.

CALLING SEQUENCE
AC1:  Terminal interrupt code

RETURNS    +1:  Always
```

DUMPI JSYS 65

```
FUNCTION
Reads data words into memory in unbuffered data mode.

RESTRICTIONS
File must be open for data mode 17

CALLING SEQUENCE
AC1:  JFN
AC2:  B0(DM%NWT)      Do not wait for completion of requested
                      operation
      B18-35(DM%PTR)  Address of command list in memory

RETURNS    +1:  Failure, error code in AC1, pointer to bad
                command in AC2
           +2:  Success, pointer in AC2 updated to last
```

command

COMMAND LIST FORMAT

Entry	Meaning
IOWD n,loc	Causes n words to be transferred from file to locations loc through loc+n-1 of process address space
XWD 0,y	Causes next command to be taken from location y
0	Terminates the command list

DUMPO JSYS 66

FUNCTION
Writes data words from memory in unbuffered data mode.

RESTRICTIONS
File must be open for data mode 17

CALLING SEQUENCE
AC1: JFN
AC2: B0(DM%NWT) Do not wait for completion of requested operation
B18-35(DM%PTR) Address of command list in memory

RETURNS +1: Failure, error code in AC1, pointer to bad command in AC2
+2: Success, pointer in AC2 updated to last command

COMMAND LIST FORMAT

Entry	Meaning
IOWD n,loc	Causes n words to be transferred from file to locations loc through loc+n-1 of process address space
XWD 0,y	Causes next command to be taken from location y
0	Terminates command list

DVCHR JSYS 117

FUNCTION
Returns the characteristics of the specified device.

CALLING SEQUENCE
AC1: JFN or device designator

RETURNS +1: Always, with
AC1: Device designator (even if JFN given)
AC2: Device characteristics word
AC3: <job # to which assigned>,,<unit #>
<job #>,,-1 if no units
-1,,<[unit #][-1]> if not assigned
-2,,<[unit #][-1]> if assigned to

device allocator

DEVICE CHARACTERISTICS WORD

Bit	Symbol	Meaning
0	DV%OUT	Device can do output
1	DV%IN	Device can do input
2	DV%DIR	Device has a directory
3	DV%AS	Device is assignable with ASND
4	DV%MDD	Device has multiple directories
5	DV%AV	Device is available or assigned to this job
6	DV%ASN	Device is assigned by ASND
8	DV%MNT	Device is mounted
9-17	DV%TYP	Device type
		0 .DVDSK Disk
		2 .DVMTA Magtape
		7 .DVLPT Line printer
		10 .DVCDR Card reader
		11 .DVFE Front-end pseudo-device
		12 .DVTTY Terminal
		13 .DVPTY Pseudo-terminal
		15 .DVNUL Null device
		16 .DVNET ARPA network
		22 .DVDCN DECnet active component
		23 .DVSRV DECnet passive component
20-35	DV%MOD	Data mode in which device can be opened
		B20 DV%M17 Dump mode
		B27 DV%M10 Image mode
		B34 DV%M1 Small buffer mode
		B35 DV%MO Normal mode

EIR JSYS 126

FUNCTION
Enables the software interrupt system for a process.

CALLING SEQUENCE
AC1: Process handle

RETURNS +1: Always

ENQ JSYS 513

FUNCTION
Requests access to a specific resource by placing a request in the queue for that resource.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability for some functions. In non-zero sections, OWGBPs must specify 7-bit bytes.

ENQ

CALLING SEQUENCE
AC1: Function code
AC2: Address of argblk

RETURNS +1: Failure, error code in AC1
+2: Success

FUNCTION CODES

Code	Symbol	Meaning
0	.ENQBL	Queue requests and block process until all requested locks are acquired
1	.ENQAA	Queue requests and acquire locks only if all requested resources are immediately available
2	.ENQSI	Queue requests
3	.ENQMA	Modify access of a previously queued request

ARGUMENT BLOCK

Word	Symbol	Meaning
0	.ENQLN	B0-5 Header length B6-17 # of locks B18-35 Length of argblk
1	.ENQID	<PSI channel number>,,<request ID>
2	.ENQLV	<flags & level number>,,<[JFN][-1][-2][-3]> B0(EN%SHR) Access to this resource is to be shared B1(EN%BLN) Ignore level number of resource B2(EN%NST) Allow ownership of this lock to be nested B3(EN%LTL) Allow a long-term lock on this resource B9-17(EN%LVL) Level number associated with this resource B18-35 JFN Associated file has standard protection -1 Resource accessible only by processes in job -2 Resource accessible by any job on system -3 Resource accessible only by enabled WHL/OPR processes
3	.ENQUC	Pointer to string or a 5B2+33-bit user code
4	.ENQRS	<# of resources in pool>,,<# requested> or 0,,<group number> if only one resource of type exists
5	.ENQMS	Address of a resource mask block
n-4		<flags & level number>,,[JFN][-1][-2][-3]
n-3		Pointer to string or 5B2+33-bit user code
n-2		<# of resources in pool>,,<# requested> or 0,,<group number>
n-1		Address of a resource mask block

ENQC JSYS 515

FUNCTION
Returns the current status of the given resource and obtains information about the state of the queues.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability for some functions. In non-zero sections, OWGBPs must specify 7-bit bytes.

CALLING SEQUENCE
AC1: Function code
AC2: Address of argblk
AC3: Address of block for status information (.ENQCS only)

RETURNS +1: Failure, error code in AC1
+2: Success

FUNCTION CODES

Code	Symbol	Meaning
0	.ENQCS	Returns status of specified resources
1	.ENQCG	Return ENQ/DEQ quota for specified job
2	.ENQCC	Change ENQ/DEQ quota for specified job (enabled WHL)
3	.ENQCD	Dump ENQ/DEQ locks and queue entries into argblk (enabled WHL)

STATUS BLOCK

Word	Meaning	
0	Resource status flags	
	B0(EN%QCE)	Error has occurred in corresponding resource request; B18-35 contain error code
	B1(EN%QCO)	Process owns the lock
	B2(EN%QCQ)	Process is in queue waiting for this resource
	B3(EN%QCX)	Lock has been allocated for exclusive access
	B4(EN%QCB)	Process is in queue waiting for exclusive access to resource
	B9-17(EN%LVL)	Level number of the resource
	B18-35(EN%JOB)	Job # of lock owner
1	36-bit time stamp	
2	<# of processes with lock>,,<request ID>	

ARGUMENT BLOCK

Function	Word	Contents
.ENQCS		See ENQ JSYS for argblk
.ENQCG	0	<ignored>,,<job #>
.ENQCC	0	<new quota>,,<job #>
.ENQCD	0	Length of block
	n	Returned data

Data Returned by Function .ENQCD
Lock Data

Word	Symbol	Meaning
0	.ENQDF	B0-8 Flags B9-17 Level number B18-35 OFN, 40000+job #, -2, or -3
1	.ENQDR	<total resources in pool>,,<# remaining> or 0,,<group number>
2	.ENQDT	Time stamp of last request locked
3	.ENQDC	User code of lock or beginning of string

Queue Data

0	.ENQDF	B0-8 Flags B9-17 PSI channel B18-35 job # queue entry creator
1	.ENQDI	<group # or number requested>,,<request ID>

Flags Returned in Word 0 for Function .ENQCD

Bit	Symbol	Meaning
B0	EN%QCL	Block contains lock data (if off, queue data)
B1	EN%QCO	Process owns the lock
B2	EN%QCT	Lock contains a text string
B3	EN%QCX	Lock is for exclusive access
B4	EN%QCB	Process is blocked until exclusive access is available

EPCAP JSYS 151

FUNCTION
Enables the capabilities for the specified process.

CALLING SEQUENCE
AC1: Process handle
AC2: Capabilities the process can enable
AC3: Capabilities to enable

RETURNS +1: Always

ERSTR JSYS 11

FUNCTION
Translates a TOPS-20 error number to its corresponding text string and writes the string to the specified destination.

CALLING SEQUENCE
AC1: Destination designator
AC2: <process handle>,,<error number>; -1 for most recent
AC3: -<maximum number of bytes to transfer>,,0;
or 0 for no limit

RETURNS +1: Failure, undefined error number
+2: Failure, string size out of bounds or invalid

destination designator
+3: Success

ESOUT JSYS 313

FUNCTION
Outputs an error string.

CALLING SEQUENCE
AC1: Byte pointer to ASCIZ error string

RETURNS +1: Always, with updated byte pointer in AC1

FFFFP JSYS 31

FUNCTION
Finds the 1st free page in the specified file.

CALLING SEQUENCE
AC1: <starting page number>,,JFN

RETURNS +1: Always, with
AC1: JFN,,<page number>
or -1 if there is no free page

FFORK JSYS 154

FUNCTION
Freezes one or more processes.

CALLING SEQUENCE
AC1: Process handle

RETURNS +1: Always

FFUFP JSYS 211

FUNCTION
Finds the first used page of the file at or beyond the specified page number.

CALLING SEQUENCE
AC1: JFN,,<starting page number>

RETURNS +1: Failure, error code in AC1
+2: Success, page number in the right half of AC1

FLHST JSYS 277

FUNCTION
"Flushes" an ARPANET host, causing the NCP tables containing that host's status information to be purged of all information regarding previous partially terminated connections.

RESTRICTIONS
For ARPANET systems only. Requires enabled WHEEL, OPERATOR, or NET WIZARD capability.

CALLING SEQUENCE
AC1: Number of host to be flushed

RETURNS +1: Always

FLIN JSYS 232

FUNCTION
Inputs a floating-point number from the specified source.

CALLING SEQUENCE
AC1: Source designator

RETURNS +1: Failure, with
AC1: Updated byte pointer, if pertinent
AC3: Error code
+2: Success, with
AC1: Updated byte pointer, if pertinent
AC2: Single-precision, floating-point number

FLOUT JSYS 233

FUNCTION
Outputs a floating-point number to the specified destination.

CALLING SEQUENCE
AC1: Destination designator
AC2: Normalized, single-precision, floating-point number
AC3: Format control word

RETURNS +1: Failure, with
AC1: Updated byte pointer, if pertinent
AC3: Error code
+2: Success, with
AC1: Updated byte pointer, if pertinent

GACCT JSYS 546

FUNCTION
Returns the current account for the specified job.

RESTRICTIONS
Requires enabled WHEEL, OPERATOR, or CONFIDENTIAL INFORMATION ACCESS capability.

CALLING SEQUENCE
AC1: Job #, or -1 for current job
AC2: Byte pointer to string for alphanumeric account designator (if any)

RETURNS +1: Always, with updated pointer to account string in AC2

GACTF JSYS 37

FUNCTION
Returns the account designator to which the specified file is being charged.

CALLING SEQUENCE
AC1: JFN
AC2: Byte pointer to string for account (if any)

RETURNS +1: Failure, error code in AC1
+2: Success, updated byte pointer in AC2
+3: Success, 5B2+account number returned in AC2

GCVEC JSYS 300

FUNCTION
Returns the entry vector and the UUO locations for the compatibility package.

CALLING SEQUENCE
AC1: Process handle

RETURNS +1: Always, with
AC1: B0-17 Entry vector length
B18-35 Entry vector address
AC2: <UUO location>,,<PC location>

GDSKC JSYS 214

FUNCTION
Returns information on the given structure's disk usage and availability.

CALLING SEQUENCE
AC1: Device designator (structure) or DSK: for connected structure

RETURNS +1: Always, with
AC1: Number of pages in use
AC2: Number of pages available

GDSTS JSYS 145

FUNCTION
Returns the status of a device for user I/O.

CALLING SEQUENCE
AC1: JFN

RETURNS +1: Always, with
AC2: Device-dependent status bits
AC3: Device-dependant
For magtape:
<# of hardware bytes transferred>,,0
For lineprinter:
last value of page counter register or
-1 if no page counter register
For ARPANET network-connection files:
AC2: Connection state (01-16) in B0-3
AC3: Foreign host number (octal)
AC4: Foreign socket number (octal)

GDVEC JSYS 542

FUNCTION
Returns the entry vector for the Record Management System (RMS).

RESTRICTIONS
Requires RMS software (currently available only with BASIC and COBOL).

CALLING SEQUENCE
AC1: Process handle

RETURNS +1: Always, with
AC2: B0-17 Entry vector length
B18-35 Entry vector address

GET JSYS 200

FUNCTION
Gets a save file, copying or mapping it into the process as appropriate, and updates the monitor's data base for the process by copying the entry vector and the list of program data vector addresses (PDVA's) from the save file.

RESTRICTIONS
Some functions require WHEEL or OPERATOR capability.

CALLING SEQUENCE

AC1:	B0-17	Process handle
	B19(GT%ADR)	Use memory address limits given in AC2
	B20(GT%PRL)	Preload pages being mapped
	B21(GT%NOV)	Do not overlay existing pages; return error
	B22(GT%ARG)	If on, AC2 contains address of argblk
	B24-35(GT%JFN)	JFN of the save file
AC2:	<lowest process page #>,,<highest process page #> or address of argblk	

RETURNS +1: Always

ARGUMENT BLOCK

Word	Symbol	Meaning
0	.GFLAG	Flags for remainder of argblk
	0 GT%LOW	.GLOW contains lowest page number within process to use
	1 GT%HGH	.GHIGH contains highest page number within process to use
	2 GT%BAS	.GBASE contains the section number to use
	3 GT%CCH	Clear system's program cache (WHL/OPR)
	4 GT%CSH	Place in cache the program name being loaded into memory (WHL/OPR)
1	.GLOW	Lowest process page number into which file page gets loaded
2	.GHIGH	Highest process page number into which file page gets loaded
3	.GBASE	Section number into which file pages are loaded (single-section save files only)

GETAB JSYS 10

FUNCTION
Returns a word from the specified system table.

RESTRICTIONS
Requires GETAB capability (bit SC%GTB in process capability word).

CALLING SEQUENCE
AC1: <index into table>,,<table number>

RETURNS +1: Failure, error code in AC1
+2: Success, 36-bit word from table in AC1

GETER JSYS 12

FUNCTION
Returns the most recent error condition encountered in a process.

CALLING SEQUENCE
AC1: Process handle

RETURNS +1: Always, with
AC2: <process handle>,,<most recent error>

GETJI JSYS 507

FUNCTION
Obtains information about the specified job.

CALLING SEQUENCE
AC1: Job #; -1 for current job; or 400000+TTY number
AC2: -<length of destination block>,,<address of block>
AC3: Offset of 1st entry desired from job information table

RETURNS +1: Failure, error code in AC1
+2: Success, with updated pointer in AC2 and requested entries stored in specified block

JOB INFORMATION TABLE

Word	Symbol	Meaning
0	.JIJNO	Job #
1	.JITNO	Job's terminal number; -1 if detached
2	.JIUNO	Job's user number
3	.JIDNO	Job's connected directory number
4	.JISNM	Subsystem name (SIXBIT)
5	.JIPNM	Program name (SIXBIT)
6	.JIRT	Runtime (in mss.)
7	.JICPJ	Controlling PTY job #; -1 if no PTY
10	.JIRTL	Runtime limit; -1 if no time limit
11	.JIBAT	If -1, job is controlled by batch
12	.JIDEN	Default magtape density
13	.JIPAR	Default magtape parity
14	.JIDM	Default magtape data mode
15	.JIRS	Default magtape record size in bytes
16	.JIDFS	If 1, deferred spooling in effect
17	.JILNO	Job's logged-in directory number
20	.JISRM	Byte pointer to destination for job's session remark

```
21      .JILLN   Date and time of user's last login before the
                 current job
22      .JISRT   Job CPU time at start of last session
23      .JISCT   Console time at start of last session
24      .JIT2O   O if job is at EXEC level; -1 if at program
                 level
25      .JISTM   Time when job was created; -1 if system
                 time/date not set when job created
26      .JIBCH   Batch stream number and batch flags
                 BO-1    OB%WTO  Write-to-operator capabilities
                                 O  .OBALL  WTO and WTOR
                                 1  .OBNWR  No WTOR allowed
                                 2  .OBNOM  No message allowed
                 B1O     OB%BSS  OB%BSN contains batch-stream #
                 B11-17  OB%BSN  Batch-stream #
27      .JILLO   Logical location (node name)
```

GETNM JSYS 177

FUNCTION
Returns the name of the program currently being used by the job.

RETURNS +1: Always, with SIXBIT program name in AC1

GETOK% JSYS 574

FUNCTION
Requests access to the specified system resource from the installation's access-control program.

CALLING SEQUENCE
AC1: Function code
AC2: Address of argblk
AC3: Length of the argblk
AC4: Job # or user number request is for

RETURNS +1: Always, with
O in Word O of status block if access granted
1B18+error number in Word O of status block
if request denied

FUNCTION CODES

Code	Symbol	Meaning/Argblk
1	.GOASD	Assign a device
		O .GEERB Error block address
		1 .GEADD Device designator
2	.GOCAP	Enable capabilities (right half privileges only)
		O .GEERB Error block address
		1 .GENCP New capability word
3	.GOCJB	Allow CRJOB JSYS to be executed

```
                0   GEERB  Error block address
 4      .GOLOG  Allow LOGIN
                0  .GEERB  Error block address
                1  .GELUN  User number
 5      .GOCFK  Allow CFORK JSYS to be executed
                0  .GEERB  Error block address
                1  .GEFCT  # of forks already in use by job
 6      .GOTBR  Allow setting of terminal baud rate
                0  .GEERB  Error block address
                1  .GELIN  Line number
                2  .GESPD  <input speed>,,<output speed>
 7      .GOLGO  Inform access-control program of a logout
                0  .GEERB  Error block address
                1  .GEUSD  Number of pages used
                2  .GEQUO  Directory quota
                3  .GERLG  Job # logging out; -1 if caller
10      .GOENQ  Allow setting of ENQ quota
                0  .GEERB  Error block address
                1  .GEEQU  Desired quota
                2  .GEEUN  Job #
11      .GOCRD  Allow directory creation
                0  .GEERB  Error block address
12      .GOSMT  Allow MOUNT of structure
                0  .GEERB  Error block address
                1  .GESDE  Device designator
13      .GOMDD  Allow entry to MDDT
                0  .GEERB  Error block address
14      .GOCLS  Set scheduler class for a job
                0  .GEERB  Error block address
                1  .GEJOB  Job #
                2  .GECLS  Class desired
15      .GOCLO  Set scheduler class at login
                0  .GEERB  Error block address
16      .GOMTA  MT: access request
                0  .GEERB  Error block address
                1  .GEACC  Access code from HDR1 label
                2  .GEUSN  User number
                3  .GEUNT  MT: unit number
                4  .GEACD  Desired access bits (FP%xxx)
                5  .GELTP  Label type (.LTxxx)
17      .GOACC  Allow ACCESS or CONNECT
                0  .GEERB  Error block address
                1  .GOACO  Flags from ACCES JSYS
                2  .GOAC1  Directory number
20      .GOOAD  Allow device assignment due to OPENF
                0  .GEERB  Error block address
                1  .GEADD  Device designator
21      .GODNA  Allow DECNET access
                0  .GEERB  Error block address
22      .GOANA  Allow ARPANET access
                0  .GEERB  Error block address
23      .GOATJ  Allow ATTACH
                0  .GOTJB  Target job #
                1  .GEADD  Source TTY number
400000+n        Customer-reserved functions
```

ERROR BLOCK FORMAT (RET)

Word	Symbol	Contents
0	.GESIZ	Count of words in block (including this word)
1	.GEERN	Error number
2	.GEPTR	Byte pointer to error string location
3	.GEBSZ	Maximum bytes user can accept in error string

GEVEC JSYS 205

FUNCTION
Returns the section-relative entry vector of the specified process.

RESTRICTIONS
Process must run in a single section of memory.

CALLING SEQUENCE
AC1: Process handle

RETURNS +1: Always, with entry vector word in AC2

GFRKH JSYS 164

FUNCTION
Gets a handle on a process that currently is not known to the calling process but is known to another process.

CALLING SEQUENCE
AC1: Handle of process that has handle on desired process
AC2: Process handle relative to process in AC1 that refers to desired process

RETURNS +1: Failure, with error code in AC1
+2: Success, with
AC1: Relative handle of the desired process

GFRKS JSYS 166

FUNCTION
Returns the process structure of the current job from a given process downward.

RESTRICTIONS
Requires WHEEL or OPERATOR capability for some functions.

CALLING SEQUENCE
AC1: Process handle of the starting point
AC2: B0(GF%GFH) Return relative process handles for each process
B1(GF%GFS) Return status for each process

AC3: -<word count in PSB>,,<address of PSB>

RETURNS +1: Failure, error code in AC1
+2: Success, all process handles are returned

GFUST JSYS 550

FUNCTION
Returns the name of either the author of the file or the user who last wrote to the file.

CALLING SEQUENCE
AC1: <function code>,,JFN
AC2: Pointer to author/user string

RETURNS +1: Always, with an updated byte pointer in AC2

FUNCTION CODES

Code	Symbol	Meaning
0	.GFAUT	Return name of author of file
1	.GFLWR	Return name of user who last wrote to file

GIVOK% JSYS 576

FUNCTION
Allows a privileged access-control program to permit or refuse a user program's access to a specified system resource.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability.

CALLING SEQUENCE
AC1: Request number (from RCVOK% message)
AC2: 0 to permit request
1B18 + error number to refuse request
AC3: Pointer to ASCIZ message string (80. characters maximum); or 0

RETURNS +1: Always

GJINF JSYS 13

FUNCTION
Returns information pertaining to the current job.

RETURNS +1: Always, with
AC1: User number under which job is running
AC2: Directory number to which job is connected

AC3: Job #
AC4: TTY # attached to job; or -1 if none

GNJFN JSYS 17

FUNCTION
Assigns the JFN to the next file in a group of files that have been specified with wildcard characters.

CALLING SEQUENCE
AC1: Indexable file handle returned by GTJFN (flags,,JFN)

RETURNS +1: Failure; occurs on 1st call to GNJFN with no flags indicating wildcard fields on in B18-35 of AC1 (JFN released if no more files in group)
+2: Success, same JFN is assigned to next file in group, with
AC1: B13 GN%STR Structure changed
B14 GN%DIR Directory changed
B15 GN%NAM Name changed
B16 GN%EXT File type changed

GPJFN JSYS 206

FUNCTION
Returns the primary JFNs of the specified process.

CALLING SEQUENCE
AC1: Process handle

RETURNS +1: Always, with
AC2: B0-17 Primary input JFN
B18-35 Primary output JFN

GTAD JSYS 227

FUNCTION
Returns the current date in the internal system format.

RETURNS +1: Always, with
AC1: Day,,<fraction of day> or
-1 if system date not set

GTDAL JSYS 305

FUNCTION
Returns the disk allocation for the specified directory.

CALLING SEQUENCE
AC1: Directory number; -1 for connected directory

RETURNS +1: Always, with
AC1: Working storage limit for directory
AC2: Number of pages being used
AC3: Permanent storage limit for directory

GTDIR JSYS 241

FUNCTION
Returns information about the given directory.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability.

CALLING SEQUENCE
AC1: Directory number; or 0 for system default settings
AC2: Address of block to store directory information
AC3: Byte pointer to password string

RETURNS +1: Always, with updated byte pointer in AC3

ARGUMENT BLOCK (RET)

Word	Symbol	Meaning
0	.CDLEN	Length of argblk; defaults to 15
1	.CDPSW	Byte pointer to password string
2	.CDLIQ	Working disk storage quota
3	.CDPRV	Capabilities for this user
4	.CDMOD	Mode word
		B0(CD%DIR) Directory is files-only
		B1(CD%ANA) Obsolete
		B2(CD%RLM) Repeat messages from file <SYSTEM>MAIL.TXT each time user logs in
		B7(CD%DAR) File should be archived rather than migrated when on-line expiration date reached
5	.CDLDQ	Permanent disk storage quota
6	.CDNUM	Directory number
7	.CDFPT	Default file protection (18 bits, R-J)
10	.CDDPT	Directory protection (18 bits, R-J)
11	.CDRET	Default generation retention count
12	.CDLLD	Date of last login
13	.CDUGP	Address of user group list for this directory
14	.CDDGP	Address of directory group list
15	.CDSDQ	Maximum number of sub-directories allowed
16	.CDCUG	Address of user group list
17	.CDDAC	0

```
20    .CDDNE    Default on-line expiration date and time
21    .CDDFE    Default off-line expiration date and time
```

GTFDB JSYS 63

FUNCTION
Returns some or all of the file descriptor block (FDB) for the specified file.

CALLING SEQUENCE
AC1: JFN
AC2: <# of FDB words to read>,,<offset of 1st word desired>
AC3: Address of block for returned data

RETURNS +1: Always

GTHST JSYS 273

FUNCTION
Obtains information about ARPANET hosts.

RESTRICTIONS
For ARPANET systems only

CALLING SEQUENCE
AC1: Function code
AC2: Function-specific argument
AC3: Function-specific argument
AC4: Function-specific argument

RETURNS +1: Failure, error code in AC1
         +2: Success, data returned in ACs

FUNCTION CODES

Code	Symbol	Function
0	.GTHSZ	Returns host table sizes Returns AC2: -<number host names>,,0 AC3: -<length of HSTSTS table>,,0 AC4: Local host number (in 32-bit Internet format)
1	.GTHIX	Returns name string associated with host Arguments AC2: Byte pointer to destination for name string AC3: Index into name table (returned by GETAB) Returns AC2: Updated byte pointer AC3: Host number AC4: Host status; if name is a nickname, HS%NCK is on

2	.GTHNS	Returns primary name for given host number Arguments AC2: Byte pointer to destination for primary name AC3: Host number Returns AC2: Updated byte pointer AC3: Host number AC4: Host status
3	.GTHSN	Translates specified host name string to its host number Arguments AC2: Byte pointer to host name string Returns AC2: Updated byte pointer AC3: Host number AC4: Host status
4	.GTHHN	Returns current status of given host Arguments AC3: Host number Returns AC4: Host status
5	.GTHHI	Returns host number and host status Arguments AC3: Index into HSTSTS (returned by GETAB) Returns AC3: Host number AC4: Host status

FLAGS IN HOST STATUS WORD

Bits	Symbol	Meaning
1B0	HS%UP	Host is up
1B1	HS%VAL	Valid status
B2-4	HS%DAY	Day when up if currently down
B5-9	HS%HR	Hour
B10-13	HS%MIN	5 minute interval
B14-17	HS%RSN	Reason
1B18	HS%SRV	Host is server
1B19	HS%USR	Host is user
1B20	HS%NCK	Nickname
B21-26	HS%STY	System type mask
1B27	HS%NEW	RAS, RAR, RAP, etc

System Type Flags (HS%STY)

Bits	Symbol	Meaning
1B26	.HS10X	TENEX
2B26	.HSITS	ITS
3B26	.HSDEC	TOPS-10
4B26	.HSTIP	TIP
5B26	.HSMTP	MTIP
6B26	.HSELF	ELF
7B26	.HSANT	ANTS
10B26	.HSMLT	MULTICS
11B26	.HST20	TOPS-20
12B26	.HSUNX	UNIX

GTJFN JSYS 20

FUNCTION
Returns a JFN for the specified file. The short form accepts the filespec from a string in memory or from a file, but not from both; the long form accepts the filespec from either memory or a file (if both are provided, the string in memory is used first).

CALLING SEQUENCE
AC1: Flags,,generation (short form)
O,,<address of argblk> (long form)
AC2: Source designator from which to obtain filespec (short form)
Byte pointer to ASCIZ filespec string; or O if none (long form)

RETURNS +1: Failure, error code in AC1
+2: Success, with
AC1: Flags,,JFN
AC2: Updated byte pointer, if pertinent

GTJFN FLAG BITS

Bit	Symbol	Meaning
0	GJ%FOU	File is to be assigned next higher generation
1	GJ%NEW	File must not exist (no effect on a parse-only JFN)
2	GJ%OLD	File must exist (no effect on a parse-only JFN)
3	GJ%MSG	Print message after filespec if user types ESC; possible messages: !NEW FILE! !NEW GENERATION! !OLD GENERATION! !OK! !CONFIRM!
4	GJ%CFM	Require confirmation from user (if GJ%FNS is on) to verify filespec
5	GJ%TMP	File specified is a temporary file
6	GJ%NS	Search only the 1st specification in a multiple logical name assignment for file
7	GJ%ACC	JFN cannot be accessed by inferior processes
8	GJ%DEL	Consider deleted files when searching for file
9-10	GJ%JFN	Associate JFN in word 10 (.GJJFN) of argblk with filespec according to value (long form only) 0(.GJDNU) Ignore JFN supplied 2(.GJERR) Assign JFN supplied; return error if not available 3(.GJALT) Assign JFN supplied; assign alternate if not available
11	GJ%IFG	Allow use of wildcards in fields of filespec
12	GJ%OFG	Associate JFN with filespec string only, not file
13	GJ%FLG	Return flags in the left half of AC1 if

		successful
14	GJ%PHY	Ignore job-wide logical names
15	GJ%XTN	Argblk contains more than 10 words (long form only)
16	GJ%FNS	If on, AC2 contains <input JFN>,,<output JFN> if off, AC2 contains byte pointer to ASCIZ filespec string (short form only)
17	GJ%SHT	Must be on for short form GTJFN; must be off for long form GTJFN
18-35		Generation of file or one of:
		0(.GJDEF) Use next higher generation if 1B0(GJ%FOU); use highest existing generation if 0B0(GJ%FOU)
		-1(.GJNHG) Use next higher generation if none supplied
		-2(.GJLEG) Use lowest existing generation
		-3(.GJALL) Use all generations and assign JFN to 1st file in group (GJ%IFG must be set)

ARGUMENT BLOCK (Long Form Only)

Word	Symbol	Meaning
0	.GJGEN	Flags,,<generation>
1	.GJSRC	<input JFN>,,<output JFN>
2	.GJDEV	Byte pointer to ASCIZ default device string; or 0 for user's connected structure
3	.GJDIR	Byte pointer to ASCIZ default directory string; or 0 for user's connected directory
4	.GJNAM	Byte pointer to ASCIZ default filename string; if 0, string in AC2 or input JFN must supply filename
5	.GJEXT	Byte pointer to ASCIZ default file type string; or 0 for null file type
6	.GJPRO	Byte pointer to ASCIZ default protection string; or 0 for default directory protection or protection of next lower generation
7	.GJACT	Byte pointer to ASCIZ default account string; or 0 for user's LOGIN account (unless changed)
10	.GJJFN	JFN to associate with filespec if GJ%JFN is set in word 0 (.GJGEN)
11	.GJF2	Flags,,<count of remaining words in block> if GJ%XTN is set in word 0 (.GJGEN) (OPT)
		B0(G1%RND) Return if filename buffer empty and user attempts to delete character
		B2(G1%NLN) Filenames limited to 6 characters, file types to 3 characters; generation, temporary status, protection, and account fields not allowed in string or input data
		B3(G1%RCM) Return confirmation message in destination buffer
		B4(G1%RIE) Return if input buffer empty, and user attempts to delete character

```
                    B5(G1%IIN)  Consider invisible files when
                                searching for file
                    B6(G1%SLN)  Prohibit expansion of logical
                                names
12      .GJCPP      Byte pointer to destination string for copy
                    of user's typescript
13      .GJCPC      Number of bytes available in destination
                    string; if 0, 130 bytes assumed
14      .GJRTY      Byte pointer to CTRL/R buffer
15      .GJBFP      Obsolete
16      .GJATR      Pointer to filespec attribute block

ATTRIBUTE BLOCK (Long Form Only)
Word  Contents
 0    Count of words in block (including this word)
 1    Byte pointer to argument string
1+n   Byte pointer to argument string

ATTRIBUTE VALUES (Long Form Only)
Keyword             Attribute Value
A:                  Installation-defined account string
BDATA:              DECnet binary optional data
BLOCK-LENGTH:       Magnetic-tape block length (in bytes)
BPASSWORD:          DECnet binary password
CHARGE:             DECnet account string
DATA:               DECnet optional data
EXPIRATION-DATE:    Magnetic-tape expiration date
FORMAT:             Magnetic-tape record format
                    F  Fixed-length records
                    D  Variable-length records
                    S  Spanned records
                    U  Binary files with 36-bit words
OFF-LINE            NONE - display-only keyword
P:                  File protection value (octal)
PASSWORD:           DECnet password string
POSITION:           File sequence number for positioning
                    magnetic-tape
RECORD-LENGTH:      Magnetic-tape record length (in bytes)
T                   NONE - display-only keyword
USERID:             DECnet user ID string

Flags Returned in AC1
Bit   Symbol   Meaning
 0    GJ%DEV   Device field of filespec contained wildcards
 1    GJ%UNT   Unit field of filespec contained wildcards
 2    GJ%DIR   Directory field of filespec contained
               wildcards
 3    GJ%NAM   Filename field of filespec contained
               wildcards
 4    GJ%EXT   File type field of filespec contained
               wildcards
 5    GJ%VER   Generation field of filespec contained
               wildcards
 6    GJ%UHV   File used has highest generation
 7    GJ%NHV   File used has next higher generation
 8    GJ%ULV   File used has lowest generation
```

9	GJ%PRO	Protection field of filespec was given
10	GJ%ACT	Account field of filespec was given
11	GJ%TFS	Filespec is for temporary file
12	GJ%GND	Deleted files were not considered when assigning JFNs
17	GJ%INV	Invisible files were not considered when assigning JFNs

GTRPI JSYS 172

FUNCTION
Returns paging trap information for the specified process.

CALLING SEQUENCE
AC1: Process handle

RETURNS +1: Always, with
AC1: # of pager traps since process started
AC2: # of page faults since process started
AC3: Time spent (in mss) in page routines since process started

GTNCP% JSYS 272

FUNCTION
Obtains information about the NCP.

RESTRICTIONS
For ARPANET systems only

CALLING SEQUENCE
AC1: Function code
AC2: Function-specific argument
AC3: Function-specific argument
AC4: Function-specific argument

RETURNS +1: Failure, error code in AC1
+2: Success, data returned in AC's

FUNCTION CODES

Code	Symbol	Function
0	.GTNSZ	Returns negative number of NCP connections Returns AC2: -<# of NCP connections>,,0 AC3: -<# of NVTs>,,<line # of 1st NVT>
1	.GTNIX	Returns status of connection number Arguments AC2: Connection number AC3: 30-bit address of data block AC4: -<block length>,,<index of 1st item> Returns Data in data block

```
2     .GTNNI    Return status of NVT line number (input
                connection)
                Arguments
                AC2:  NVT line number (input)
                AC3:  30-bit address of data block
                AC4:  -<block length>,,<index of 1st item>
                Returns
                Data in data block
3     .GTNNO    Return status of NVT connection (output
                connection)
                Arguments
                AC2:  NVT line number (output)
                AC3:  30-bit address of data block
                AC4:  -<block length>,,<index of 1st item>
                Returns
                Data in data block
4     .GTNJF    Return status of network-connection JFN
                Arguments
                AC2:  JFN
                AC3:  30-bit address of data block
                AC4:  -<block length>,,<index of 1st item>
                Returns
                Data in data block

FORMAT OF RETURNED DATA BLOCK
Word  Symbol    Contents
 0    .NCIDX    NCP connection index
 1    .NCFHS    Foreign host
 2    .NCLSK    Local socket
 3    .NCFSK    Foreign socket
 4    .NCFSM    State of connection
 5    .NCLNK    Link
 6    .NCNVT    NVT, or -1 if none
 7    .NCSIZ    Byte size of connection
10    .NCMSG    Message allocation
11    .NCBAL    Bit allocation
12    .NCDAL    Desired allocation
13    .NCBTC    Bits transferred
14    .NCBPB    Bytes per buffer
15    .NCCLK    Time-out countdown
16    .NCSTS    Connection status
```

GTRPW JSYS 171

```
FUNCTION
Returns the trap words.

CALLING SEQUENCE
AC1:  Process handle

RETURNS   +1:  Always, with
               AC1:  Status word from last memory trap or 0
                     if no traps
               AC2:  Last monitor call that had an error
```

STATUS WORD

Bit	Meaning
B0(PF%USR)	Page failure - user mode reference
B5(PF%WRT)	Page failure - write reference
B14(TSW%RD)	Trap status - read (always on)
B15(TSW%WT)	Trap status - write (same setting as B5)
B16(TSW%EX)	Trap status - execute (always on)
B17(TSW%MN)	Trap status - monitor mode reference
B18-B35	Address of reference that caused trap

GTSTS JSYS 24

FUNCTION
Returns the status of a file associated with a JFN.

CALLING SEQUENCE
AC1: 0,,JFN

RETURNS +1: Always, with
AC2: status: OB10, if JFN illegal

JFN STATUS WORD

Bit	Meaning
B0(GS%OPN)	File is open
B1(GS%RDF)	File is open for read access
B2(GS%WRF)	File is open for write access
B3(GS%XCF)	File is open for execute access
B4(GS%RND)	File is open for non-append access
B7(GS%LNG)	File is longer than 512 pages
B8(GS%EOF)	Last read was past end of file
B9(GS%ERR)	File may be in error
B10(GS%NAM)	Filespec is associated with this JFN
B11(GS%AST)	JFN is parse-only
B12(GS%ASG)	JFN is currently being assigned
B13(GS%HLT)	I/O errors are considered terminating conditions
B17(GS%FRK)	JFN is restricted
B18(GS%PLN)	If on, file line numbers are passed during input; if 0, line numbers are stripped before input
B32-B35(GS%MOD)	Data mode of the file

GTTYP JSYS 303

FUNCTION
Returns the terminal type number for the specified terminal line.

CALLING SEQUENCE
AC1: Terminal designator

RETURNS +1: Always, with

```
AC2:  Terminal type number
AC3:  B0-17    # of input buffers to allocate
      B18-35   # of output buffers to allocate
```

HALTF JSYS 170

FUNCTION
Halts the current process and any inferior processes of the current process. Sets B1-17(RF%STS) in the Process Status Word to 2(.RFHLT).

HFORK JSYS 162

FUNCTION
Halts one or more inferior processes.

CALLING SEQUENCE
AC1: Process handle (inferior processes only)

RETURNS +1: Always

HPTIM JSYS 501

FUNCTION
Returns the value of one of the high precision system clocks.

CALLING SEQUENCE
AC1: Number of the clock to read

```
RETURNS   +1:  Failure, error code in AC1
          +2:  Success, with
               AC1:  Value of specified clock
```

CLOCKS

Code	Symbol	Meaning
0	.HPELP	Elapsed time since system startup
1	.HPRNT	CPU runtime for this process

HSYS JSYS 307

FUNCTION
Initiates an orderly shutdown of timesharing.

RESTRICTIONS
Requires enabled WHEEL, OPERATOR, or MAINTENANCE capability.

CALLING SEQUENCE
AC1: Shutdown time with date and time in internal format
AC2: Date/time (internal format) when system will resume; 0 if unknown

RETURNS +1: Failure, error code in AC1
+2: Success, shutdown procedure initiated

IDCNV JSYS 223

FUNCTION
Converts separate numbers for the local year, month, day, and time into internal date and time format.

CALLING SEQUENCE
AC2: Year,,month
AC3: <day of month>,,0
AC4: B0(IC%DSA) Apply daylight savings according to setting of B1(IC%ADS)
B1(IC%ADS) Apply daylight savings if B0(IC%DSA) is on
B2(IC%UTZ) Use time zone in B12-17; if off, use local zone
B3(IC%JUD) Number in B18-35 of AC2 is in Julian day format
B12-17(IC%TMZ) Time zone if B2(IC%UTZ) is on
B18-35(IC%TIM) Local time in seconds since midnight

RETURNS +1: Failure, error code in AC1
+2: Success, with
AC2: Internal date and time
AC3: B0 and B2 On for compatibility with ODCNV
B1(IC%ADS) Daylight savings was applied
B12-17(IC%TMZ) Time zone used

IDTIM JSYS 221

FUNCTION
Inputs the date and time and converts them to internal date and time format.

CALLING SEQUENCE
AC1: Source designator
AC2: Format option flags

RETURNS +1: Failure, with
AC1; Updated byte pointer
AC2: Error code
+2: Success, with
AC1: Updated byte pointer

AC2: Date and time in internal format

IDTIM Option Flags

B1(IT%NNM)	Month may not be numeric; ignore B2-3
B2(IT%SNM)	2nd number in date is month
B3(IT%ERR)	Return error if order of day and month does not agree with setting of B2(IT%SNM)
B7(IT%NIS)	Seconds cannot be included in time specification
B8(IT%AIS)	Seconds (preceded by colon) must be included in time specification
B9(IT%NAC)	Colon cannot be used to separate hours and minutes
B10(IT%AAC)	Colon must be used to separate hours and minutes
B11(IT%AMS)	If B7-10 off, interpret time specification containing one colon as hhmm:ss
B12(IT%AHM)	If B7-10 off, interpret time specification containing one colon as hh:mm; return error if first field too large
B14(IT%N24)	Do not allow time specification in 24-hour format; require AM/PM specification
B15(IT%NTM)	Do not allow time specification to include AM, PM, NOON, or MIDNIGHT
B16(IT%NTZ)	Do not allow time zone specification

IDTNC JSYS 231

FUNCTION
Inputs the date and/or time and converts it into separate numbers for the local year, month, day, or time.

CALLING SEQUENCE
AC1: Source designator
AC2: Format option flags

RETURNS +1: Failure, with
AC1: Updated byte pointer
AC2: error code
+2: Success, with
AC1: updated byte pointer
If date was input
AC2: Year,,month
AC3: <day of month>,,<day of week>
If time was input

AC4:	BO(IC%DAS)	On if IT%NTI was set, or if IT%NDA was set and a time zone was input
	B1(IC%ADS)	On if daylight savings time zone was input, or if IT%NTI was set
	BO(IC%UTZ)	On if IT%NTI was set, or if IT%NDA was set

	and a time zone was input
B3(IC%JUD)	On if a number in Julian day format was input
B12-17(IC%TMZ)	Time zone supplied, or local time zone
B18-35(IC%TIM)	Time as seconds since midnight

IDTNC Option Flags

B0(IT%NDA)	Do not input date and ignore B1-3; if off, date required
B1(IT%NNM)	Month may not be numeric; ignore B2-3
B2(IT%SNM)	2nd number in date is month
B3(IT%ERR)	Return error if order of day/month does not match setting of B2(IT%SNM)
B6(IT%NTI)	Do not input time and ignore B7-16; if off, time required
B7(IT%NIS)	Seconds cannot be included in time specification
B8(IT%AIS)	Seconds (preceded by colon) must be included
B9(IT%NAC)	Colon cannot be used to separate hours and minutes
B10(IT%AAC)	Colon must be used to separate hours and minutes
B11(IT%AMS)	If B7-10 off, interpret time specification containing one colon as hhmm:ss
B12(IT%AHM)	If B7-10 off, interpret time specification containing one colon as hh:mm; return error if 1st field too large
B14(IT%N24)	Do not allow time specification in 24-hour format; require AM/PM specification
B15(IT%NTM)	Do not allow time specification to include AM, PM, NOON, or MIDNIGHT
B16(IT%NTZ)	Do not allow time zone specification

IIC JSYS 132

FUNCTION
Initiates software interrupts on the specified channels in a process.

CALLING SEQUENCE
AC1: Process handle
AC2: 36-bit word (1B*n* initiates interrupt on channel *n*)

RETURNS +1: Always

INLNM JSYS 503

FUNCTION
Returns a logical name that is defined either for this job or for the system.

CALLING SEQUENCE
AC1: B0-17 Function code
      B18-35 Index into table of defined logical names
AC2: Byte pointer to string for logical name

RETURNS +1: Failure, error code in AC1
         +2: Success, updated byte pointer in AC2

FUNCTION CODES

Code	Symbol	Meaning
0	.INLJB	List logical names defined for this job
1	.INLSY	List logical names defined for system

JFNS JSYS 30

FUNCTION
Returns the filespec currently associated with the JFN.

CALLING SEQUENCE
AC1: Destination designator for ASCIZ filename string
AC2: Indexable file handle; or pointer to filename string
AC3: Format control bits for string; or 0

RETURNS +1: Always, with updated byte pointer in AC1

FORMAT CONTROL VALUES

Value	Symbol	Meaning
0	.JSNOF	Do not output this field
1	.JSAOF	Always output this field
2	.JSSSD	Suppress this field if system default

FORMAT CONTROL FIELDS

Field	Meaning
B0-2(JS%DEV)	Output for device field
B3-5(JS%DIR)	Output for directory field
B6-8(JS%NAM)	Output for filename field (2 is illegal)
B9-11(JS%TYP)	Output for file type field (2 is illegal)
B12-14(JS%GEN)	Output for generation number field
B0-14(JS%SPC)	Output for all filespec fields named above
B15-17(JS%PRO)	Output for protection field
B18-20(JS%ACT)	Output for account field
B21(JS%TMP)	Return ;T if appropriate
B22(JS%SIZ)	Return size of file in pages
B23(JS%CDR)	Return creation date
B24(JS%LWR)	Return date of last write
B25(JS%LRD)	Return date of last read
B26(JS%PTR)	AC2 contains pointer to the string being returned

B27(JS%ATR)	Return filespec attributes if appropriate
B28(JS%AT1)	Return specification attribute referenced in AC4
B29(JS%OFL)	Return the "OFF-LINE" attribute
B32(JS%PSD)	Punctuate the size and date fields
B33(JS%TBR)	Tab before all fields returned, except for 1st field
B34(JS%TBP)	Tab before all fields (except 1st) with value 1 or 2
B35(JS%PAF)	Punctuate all fields from device through ;T

KFORK JSYS 153

FUNCTION
Kills one or more processes, releasing memory, PSB, and JFNs.

CALLING SEQUENCE
AC1: Process handle

RETURNS +1: Always, unless current process attempts to kill itself

LGOUT JSYS 3

FUNCTION
Kills the specified job and appends an accounting entry to the accounting data file, unless job did not login.

RESTRICTIONS
WHEEL or OPERATOR required to logout job other than current job, job logged in under same username, or PTY job controlled by current job.

CALLING SEQUENCE
AC1: Number of job to be logged out, or -1 for current job

RETURNS +1: Failure, error code in AC1
+2: Success

LNMST JSYS 504

FUNCTION
Translates a logical name to its original definition string.

CALLING SEQUENCE
AC1: Function code
AC2: Pointer to logical name string (without colon)
AC3: Pointer to string for original logical name definition

```
RETURNS    +1:  Failure, error code in AC1
           +2:  Success, updated byte pointer in AC3

FUNCTION CODES
Code   Symbol    Meaning
 0     .LNSJB    Obtain job-wide definition of logical name
 1     .LNSSY    Obtain system-wide definition of logical name
```

LOGIN JSYS 1

```
FUNCTION
Logs a job into the system.

RESTRICTIONS
In non-zero sections, OWGBPs must specify 7-bit bytes.

CALLING SEQUENCE
AC1:  36-bit user number for login
AC2:  Pointer to beginning of password string
AC3:  5B2!<account number>B35  or pointer to account string
      (maximum of 39 characters read)

RETURNS    +1:  Failure, error code in AC1
           +2:  Success, with
                AC1:  Date and time of last login
                AC2:  Updated byte pointer
                AC3:  Updated byte pointer
```

LPINI JSYS 547

```
FUNCTION
Loads the direct access Vertical Formatting Unit (VFU) or
translation Random Access Memory (RAM) for the line printer.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability.

CALLING SEQUENCE
AC1:  JFN of file containing VFU or RAM
AC2:  <status bits>,,<function code>
AC3:  Unit number of line printer

RETURNS    +1:  Always

STATUS BITS
Bit    Symbol    Meaning
B0     MO%LCP    Line printer is lowercase

FUNCTION CODES
Code   Symbol    Meaning
32     .MOLVF    Load VFU from file indicated by JFN
34     .MOLTR    Load translation RAM from file indicated by
```

JFN

MDDT% JSYS 777

FUNCTION
Transfers control to the MDDT program while preserving the context of the process that issued the MDDT% JSYS.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability.

METER% JSYS 766

FUNCTION
Returns the value of the execution accounting meter or the memory reference accounting meter.

RESTRICTIONS
Not available on KS-10 hardware.

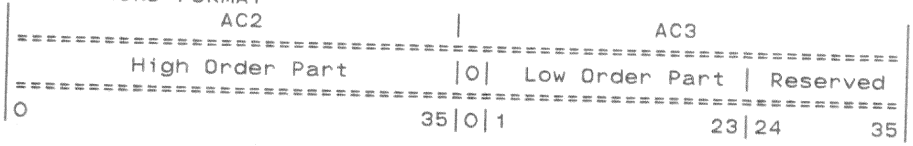
CALLING SEQUENCE
AC1: Function code

RETURNS +1: Always, with 59-bit value in AC2 and AC3

FUNCTION CODES

Code	Symbol	Meaning
1	.MEREA	Read process execution accounting meter doubleword; returns EBOX busy time (number of EBOX ticks)
2	.MERMA	Read process memory-reference accounting meter doubleword; returns count of MBOX references (number of MBOX ticks)

DOUBLE-WORD FORMAT



MRECV JSYS 511

FUNCTION
Retrieves an IPCF (Inter-Process Communication Facility) message from the process's input queue.

RESTRICTIONS
Requires enabled WHEEL, OPERATOR or IPCF capability.

```
CALLING SEQUENCE
AC1:  Length of packet descriptor block
AC2:  Address of packet descriptor block

RETURNS    +1:  Failure, error code in AC1
           +2:  Success, with
                AC1:  B0-17    Length of next entry in queue
                      B18-35   Flags from next packet or 0 if
                               queue empty
```

FORMAT OF PACKET DESCRIPTOR BLOCK

Word	Symbol	Meaning
0	.IPCFL	Flags
1	.IPCFS	PID of sender (RET)
2	.IPCFR	PID of receiver; -1 for any PID in process; -2 for any PID in job
3	.IPCFP	<length of message>,,<destination address>
4	.IPCFD	User number of sender (RET)
5	.IPCFC	Enabled capabilities of sender (RET)
6	.IPCSD	Number of sender's connected directory (RET)
7	.IPCAS	Account string of sender (RET)
10	.IPCLL	Byte pointer for destination of sender's node (optional)

FLAGS FOR WORD .IPCFL OF PACKET DESCRIPTOR BLOCK

Bit	Symbol	Meaning
B0	IP%CFB	Do not block process if no messages in queue; if set, error return if no messages
B1	IP%CFS	Use PID referenced in word .IPCFS as sender's PID
B2	IP%CFR	Use PID referenced in word .IPCFR as receiver's PID
B3	IP%CFO	Allow one send request above quota
B4	IP%TTL	Truncate message if larger than space reserved
B5	IP%CPD	Create PID for sender and return in word .IPCFS
B6	IP%JWP	Make created PID job wide (ignored unless IP%CPD set)
B7	IP%NOA	Do not allow other processes to use created PID (ignored unless IP%CPD set)
B18	IP%CFP	Packet is privileged (requires IPCF)
B19	IP%CFV	Packet is page of data
B21	IP%INT	Reserved
B22	IP%EPN	18-bit page number in word .IPCFP

MSEND JSYS 510

FUNCTION
Sends an IPCF (Inter-Process Communication Facility) message to a specific PID or to <SYSTEM>INFO.

RESTRICTIONS
Some functions require WHEEL, OPERATOR, or IPCF capability

enabled.

CALLING SEQUENCE
AC1: Length of packet descriptor block
AC2: Address of packet descriptor block

RETURNS +1: Failure, error code in AC1
+2: Success

FORMAT OF PACKET DESCRIPTOR BLOCK

Word	Symbol	Meaning
0	.IPCFL	Flags
1	.IPCFS	PID of sender; address of PID if IP%CFS or IP%CFR is set in word .IPCFL; or 0 if no PID exists for sender (RET if creating a PID)
2	.IPCFR	PID of receiver; 0 if receiver is <SYSTEM>INFO
3	.IPCFP	<message length>,,<message starting address>

FLAGS FOR WORD .IPCFL OF PACKET DESCRIPTOR BLOCK

Bit	Symbol	Meaning
B0	IP%CFB	Do not block process if no messages in queue; if set, error return if no messages
B1	IP%CFS	Use PID referenced in word .IPCFS as sender's PID
B2	IP%CFR	Use PID referenced in word .IPCFR as receiver's PID
B3	IP%CFO	Allow one send request above quota
B4	IP%TTL	Truncate message if larger than space reserved
B5	IP%CPD	Create PID for sender and return in word .IPCFS
B6	IP%JWP	Make created PID job wide (ignored unless IP%CPD set)
B7	IP%NOA	Do not allow other processes to use created PID (ignored unless IP%CPD set)
B18	IP%CFP	Packet is privileged (requires IPCF)
B19	IP%CFV	Packet is page of data
B21	IP%INT	Reserved
B22	IP%EPN	18-bit page number in word .IPCFP

FLAGS RETURNED IN WORD .IPCFL

Bit	Symbol	Meaning
B20	IP%CFZ	Zero-length message was sent; packet consists of only packet descriptor block
B24-29	IP%CFE	Error code field for <SYSTEM>INFO errors
	15 .IPCPI	Insufficient privileges
	16 .IPCUF	Invalid function
	67 .IPCSN	<SYSTEM>INFO needs name
	72 .IPCFF	<SYSTEM>INFO free space exhausted
	74 .IPCBP	PID has no name or is invalid
	75 .IPCDN	Duplicate name has been specified
	76 .IPCNN	Unknown name has been specified

		77 .IPCEN	Invalid name has been specified
B30-32	IP%CFC	System and sender code (enabled IPCF to set)	
		1 .IPCCC	Sent by <SYSTEM>IPCF
		2 .IPCCF	Sent by system-wide <SYSTEM>INFO
		3 .IPCCP	Sent by receiver's <SYSTEM>INFO
B33-35	IP%CFM	Special messages field (enabled WHL)	
		1 .IPCFN	Process's input queue contains undeliverable packet

FORMAT OF REQUEST PACKET TO <SYSTEM>INFO

Word	Symbol	Meaning
0	.IPCIO	<user-defined code>,,<<SYSTEM>INFO function>
1	.IPCI1	PID to receive copy of <SYSTEM>INFO's response
2	.IPCI2	Function-specific argument

<SYSTEM>INFO FUNCTION CODES

Function	Argument	Meaning
.IPCIW	Name	Return PID associated with specified name in word .IPCI1
.IPCIG	PID	Return name associated with specified PID in word .IPCI1
.IPCII	ASCIZ name	Assign specified name to PID of process making request
.IPCIJ	ASCIZ name	Same as .IPCII function
.IPCIK	PID	Inform PID when PID in word .IPCI2 is deleted (WHL/OPR)
.IPCIS		Disassociate all PIDs with names (not available to user programs)

MSFRK JSYS 312

FUNCTION
Starts a process in monitor mode.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability, or execution from monitor mode.

AC1: Process handle
AC2: PC word: <user mode flags>,,<virtual address>

RETURNS +1: Always

MSTR JSYS 555

FUNCTION
Performs various structure-dependent functions.

RESTRICTIONS
Some functions require enabled WHEEL, OPERATOR, or MAINTENANCE capability.

CALLING SEQUENCE
AC1: <length of argblk>,,<function code>
AC2: Address of argblk

RETURNS +1: Always, with some functions returning data in argblk

FUNCTION CODES

Code	Symbol	Privileges	Meaning
0	.MSRNU	WH/OPR/MNT	Return status of next disk unit
1	.MSRUS	WH/OPR/MNT	Return status of given disk unit
2	.MSMNT	WH/OPR	Mount structure
3	.MSDIS	WH/OPR	Dismount structure
4	.MSGSS	--	Return status of structure
5	.MSSSS	WH/OPR	Change status of structure
6	.MSINI	WH/OPR	Initialize structure
7	.MSIMC	--	Increment job's mount count for structure
10	.MSDMC	--	Decrement job's mount count for structure
11	.MSGSU	--	Return job #s of structure users
12	.MSHOM	WH/OPR	Modify home block of structure
13	.MSICF	--	Increment fork's mount count for structure
14	.MSDCF	--	Decrement fork's mount count for structure
15	.MSOFL	WH/OPR	Receive interrupt when disk comes on-line
16	.MSIIC	WH/OPR	Ignore increment check for structure use

ARGUMENT BLOCKS

Function	Word	Symbol	Meaning
.MSRNU	0	.MSRCH	Channel number (0-7)
	1	.MSRCT	Controller number
	2	.MSRUN	Unit number (0-7)
	3	.MSRST	Returned software status of unit
		B0(MS%MNT)	Unit part of mounted structure
		B2(MS%DIA)	Unit in on-line diagnostics
		B3(MS%OFL)	Unit is off-line
		B4(MS%ERR)	Unit has read error
		B5(MS%BBB)	Unit has bad BAT block
		B6(MS%HBB)	Unit has bad HOME block

			B7(MS%WLK) Unit is write locked
			B9-17(MS%TYP) Type of disk unit
			1 .MSRP4 RP04
			5 .MSRP5 RP05
			6 .MSRP6 RP06
			7 .MSRP7 RP07
			11 .MSRM3 RM03
			24 .MSR20 RP20
	4	.MSRSN	Byte pointer to ASCIZ structure name string
	5	.MSRSA	Byte pointer to ASCIZ structure alias string
	6	.MSRNS	<unit #>,,<# units in structure>
	7	.MSRSW	Number of pages for swapping on the structure
	10-12	.MSRUI	Unit ID (3 words of 11-formatted ASCII)
	13-15	.MSROI	Owner ID (3 words of 11-formatted ASCII)
	16-20	.MSRFI	File system ID (3 words of 11-formatted ASCII)
	21	.MSRSP	Number of sectors per page
	22	.MSRSC	Number of sectors per cylinder
	23	.MSRPC	Number of pages per cylinder
	24	.MSRCU	Number of cylinders per unit
	25	.MSRSU	Number of sectors per unit
	26	.MSRBT	Number of bit words in bit table per cylinder
	27	.MSRSE	Serial number of CPU for which structure is used in booting system
.MSRUS	0-27	Same as .MSRNU	
.MSMNT	0	.MSTNM	Pointer to ASCIZ string for structure name
	1	.MSTAL	Pointer to ASCIZ string for structure alias
	2	.MSTFL	Flags,,<# units in structure>
			B0(MS%NFH) Do not fix bad HOME blocks
			B1(MS%NFB) Do not fix bad BAT blocks
			B2(MS%XCL) Mount structure for exclusive use by job
			B3(MS%IGN) Ignore correctable errors in bit table and root directory
	3	.MSTUI	3 words of data for each unit in structure
			0 .MSTCH Channel # of unit
			1 .MSTCT Controller # of unit
			2 .MSTUN Unit # of unit
.MSDIS	0	.MSDNM	Device designator, or pointer to ASCIZ structure alias string
.MSGSS	0	.MSGSN	Device designator, or pointer to ASCIZ structure alias string
	1	.MSGST	Returned status word
			B0(MS%PS) Structure is public

```
                                B1(MS%DIS)  Structure is being
                                            dismounted
                                B2(MS%DOM)  Structure is domestic
                                B3(MS%PPS)  Structure is PS:
                                B4(MS%INI)  Structure is being
                                            initialized
                                B5(MS%LIM)  Directory size on
                                            structure limited to
                                            30 pages
                                B6(MS%NRS)  Structure is
                                            non-regulated
          2     .MSGNU  Number of units in structure
          3     .MSGMC  Mount count for this structure
          4     .MSGFC  Open file count for this structure
          5     .MSGSI  Pointer to ASCIZ string for
                        structure's physical ID
.MSSSS    0     .MSSSN  Device designator, or pointer to
                        ASCIZ structure alias string
          1     .MSSST  Word containing new values for bits
                        being changed
          2     .MSSMW  Mask containing bits being changed
                        B1(MS%DIS)  Structure is being
                                    dismounted
                        B2(MS%DOM)  Structure is domestic
                        B6(MS%NRS)  Structure is
                                    non-regulated
                        B7(MS%RWS)  Read-after-write
                                    checking in swapping
                                    area
                        B8(MS%RWD)  Read-after-write
                                    checking in data area
.MSINI    0     .MSINM  Byte pointer to ASCIZ structure name
                        string
          1     .MSIAL  Byte pointer to ASCIZ string
                        containing alias of structure
          2     .MSIFL  BO(MS%NFH)      Do not fix bad HOME
                                        block
                        B1(MS%NFB)      Do not fix bad BAT
                                        block
                        B2(MS%XCL)      Mount structure for
                                        exclusive use by job
                        B3(MS%IGN)      Ignore errors in bit
                                        table and root
                                        directory
                        B12-17(MS%FCN) Function
                              1  .MSCRE  Create new file
                                         system
                              2  .MSRRD  Reconstruct root
                                         directory
                              3  .MSWHB  Write new HOME
                                         blocks
                              4  .MSRIX  Rebuild index
                                         table
                        B18-35(.MSINU) # of units in
                                       structure
        3-5     .MSISU  3 words of data for each unit in
```

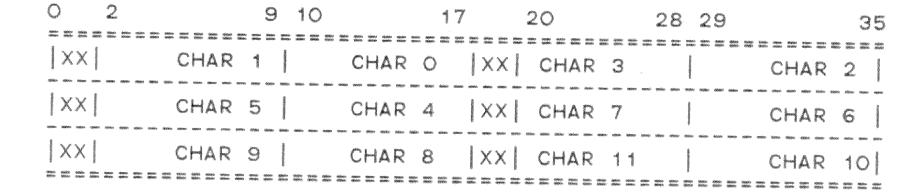
			structure
			0 .MSICH Channel # of unit
			1 .MSICT Controller # of unit
			2 .MSIUN Unit # of unit
	6	.MSIST	Status word (reserved)
	7	.MSISW	Number of pages for swapping on structure
	10	.MSIFE	Number of pages for front-end file system
	11-13	.MSIUI	Unit ID (3 words of ASCII)
	14-16	.MSIOI	Owner ID (3 words of ASCII)
	17-21	.MSIFI	File system ID (3 words of ASCII; reserved)
	22	.MSIFB	Number of pages for file BOOTSTRAP.BIN
	23	.MSISN	Serial number of CPU for which structure is used in booting system
.MSIMC	0	.MSDEV	Device designator, or pointer to ASCIZ structure alias string
	1	.MSJOB	Number of job (if not current job) whose mount count is to be incremented; (optional; enabled WHL/OPR)
.MSDMC	0	.MSDEV	Device designator, or pointer to ASCIZ structure alias string
	1	.MSJOB	Number of job (if not current job) whose mount count is to be decremented; (optional; enabled WHL/OPR)
.MSGSU	0	.MSUAL	Device designator, or pointer to ASCIZ structure alias string
	1	.MSUFL	<flag bits>,,0 B0(MS%GTA) Return users who have accessed structure B1(MS%GTM) Return users who have incremented mount count B2(MS%GTC) Return users who are connected to structure
.MSHOM	0	.MSHNM	Device designator or pointer to ASCIZ structure alias string
	1	.MSHOF	Offset specifying which word should be changed
	2	.MSHVL	Value for new bits
	3	.MSHMK	Mask showing which bits should be changed
.MSICF	0	.MSDEV	Device designator, or pointer to ASCIZ structure alias string
.MSDCF	0	.MSDEV	Device designator, or pointer to ASCIZ structure alias string
.MSOFL	0	.MSCHN	Place process on software interrupt channel; if -1, deassign previously assigned channel

DATA RETURNED BY FUNCTION .MSGSU

Word	Symbol	Meaning
1	.MSUFL	<flag bits from call>,,<# of items returned>

```
 2     .MSUJ1   <flag bits for job>,,<job #>
n+1             <flag bits for job>,,<job #>
                BO(MS%GTA)  Job accessed structure
                B1(MS%GTM)  Job incremented mount count for
                            structure
                B2(MS%GTC)  Job connected to structure
```

11-Formatted ASCII



MTALN JSYS 774

FUNCTION
Associates a given magtape drive with the specified logical unit number.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability.

CALLING SEQUENCE
AC1: <drive type>,,<logical unit # of magtape>
AC2: Decimal serial number of magtape drive

RETURNS +1: Always

MTOPR JSYS 77

FUNCTION
Performs various device-dependent control functions.

RESTRICTIONS
Some functions require enabled WHEEL or OPERATOR capability; or ARPAnet or DECnet software.

CALLING SEQUENCE
AC1: JFN of device
AC2: Function code
AC3: Function arguments or address of argblk
AC4: Function arguments (if required)

RETURNS +1: Always, with
AC2: Requested data
AC3: Requested data or updated byte pointer

ARPANET FUNCTION CODES

```
Code  Symbol   Meaning/Arguments
20    .MOACP   If connection in RFCR state, send RFC to
               accept
21    .MOSND   If connection in buffered send mode, send all
               currently buffered bytes
22    .MOSIN   Send INS/INR command
23    --       Simulate CLS F.S.M. action
24    .MOAIN   Assign interrupt channels for change of state
               or INS/INR message receipt
               AC3:  B0-5      INS/INR PSI channel
                     B12-17    State change PSI channel
25    --       If input, send allocate message; if output,
               wait for allocate message
26    --       Setup 1st I/O buffer and send allocate
               without requiring user I/O

DECnet FUNCTION CODES
Code  Symbol   Meaning/Arguments
24    .MOACN   Allow network task to enable interrupt
               channels for some tasks
               AC3:  B0-8(MO%CDN)      Connect event pending
                     B9-17(MO%INA)     Interrupt message
                                       available
                     B18-26(MO%DAV)    Data available
               Values for AC3 fields
                nn       # of channel to be enabled: 0 to 5,
                         23. to 35.
               .MOCIA    Clear interrupt
               .MONCI    No change
25    .MORLS   Return logical link status
               AC3:  <flag bits>,,<disconnect code> (RET)
                     B0(MO%CON)  Link is connected
                     B1(MO%SRV)  Link is a server
                     B2(MO%WFC)  Link waiting for connection
                     B3(MO%WCC)  Link waiting for connect
                                 confirmation
                     B4(MO%EOM)  Link has entire message to
                                 be read
                     B5(MO%ABT)  Link has been aborted
                     B6(MO%SYN)  Link has been closed
                                 normally
                     B7(MO%INT)  Link has interrupt message
                                 available
                     B8(MO%LWC)  Link has been previously
                                 connected
                     Disconnect codes
                      0   .DCX0   No special error
                      1   .DCX1   Resource allocation
                                  failure
                      2   .DCX2   Destination node does not
                                  exist
                      3   .DCX3   Node shutting down
                      4   .DCX4   Destination process does
                                  not exist
                      5   .DCX5   Invalid name field
                      6   .DCX6   Destination process queue
```

		overflow
7	.DCX7	Unspecified error
8.	.DCX8	Third party aborted link
9.	.DCX9	User abort (asynchronous disconnect)
11.	.DCX11	Undefined error code
21.	.DCX21	Connect initiate with illegal destination address
22.	.DCX22	Connect confirm with illegal destination address
23.	.DCX23	Connect initiate or confirm with zero source address
24.	.DCX24	Flow control violation
32.	.DCX32	Too many connections to node
33.	.DCX33	Too many connections to destination process
34.	.DCX34	Access not permitted
35.	.DCX35	Logical link services mismatch
36.	.DCX36	Invalid account
37.	.DCX37	Segment size too small
38.	.DCX38	Process aborted
39.	.DCX39	No path to destination node
40.	.DCX40	Link aborted due to data loss
41.	.DCX41	Destination process does not exist
42.	.DCX42	Confirmation of disconnect initiate
43.	.DCX43	Image data field too long

26	.MORHN	Return ASCII host node name at other end of logical link AC3: Pointer to string for host name (8-bit)
27	.MORTN	Return unique task name associated with this end of logical link AC3: Pointer to string for task name (8-bit)
30	.MORUS	Return source task user identification supplied in connect initiate message AC3: Pointer to string for user ID (8-bit)
31	MORPW	Return source task's password supplied in connect initiate message AC3: Pointer to string for password (8-bit)
32	.MORAC	Return account string supplied by source task in connect initiate message AC3: Pointer to string for account (8-bit)
33	.MORDA	Return optional data supplied in connect/disconnect messages AC3: Pointer to string for data (8-bit)
34	.MORCN	Return object type used by source task to address connection
35	.MORIM	Read interrupt message

```
                AC3:  Byte pointer to receiving buffer
                      (8-bit)
36    .MOSIM    Send interrupt message
                AC3:  Byte pointer to message (8-bit)
                AC4:  Count of bytes in message (16 maximum)
37    .MOROD    Return unique identification of source task
                AC3:  Pointer to string for source task
                      object-descriptor (8-bit)
40    .MOCLZ    Reject connection either implicitly or
                explicitly
                AC2:  <reject code>,,.MOCLZ
                AC3:  Pointer to string for returned data
                      (8-bit)
                AC4:  Count of bytes in data string (16
                      maximum)
41    .MOCC     Accept connection either implicitly or
                explicitly
                AC3:  Pointer to string for returned data
                AC4:  Count of bytes in data string (16
                      maximum)
42    .MORSS    Return maximum segment size that can be sent
                over this link; (illegal unless link in run
                state)
43    .MOANT    Attach network terminal  T3 = TTY #
44    .MOSNH    Set network host
                AC3:  Address of argblk
                      0            Count including this word
                      1  .SHTTY    ID of TTY controlling local
                                   job
                      2  .SHESC    Flags,,<ASCII escape char>
                                   SH%LPM   Local page mode

FRONT-END FUNCTION CODES
Code  Symbol    Meaning/Arguments
 3    .MOEOF    Flush TOPS-20 buffers and send all data to
                front end
                AC3:  0  Flush buffers and send EOF to FE
                      ≠0 Flush buffers only
 4    .MODTE    Assign specified device to DTE controller on
                front end (enabled WHL/OPR)
                AC3:  Device type (WHL/OPR)

MTA/MT FUNCTION CODES
Code  Symbol    Meaning/Arguments
 0    .MOCLE    Clear any error flags from previous MTOPR
 1    .MOREW    Rewind tape; if labeled, mount 1st volume in
                set and position at BOT
 2    .MOSDR    Set direction of tape motion for reading
                (unlabeled only)
                AC3:  0   Read forwards
                      1   Read backwards
 3    .MOEOF    Write tape mark
 4    .MOSDM    Set hardware data mode for tape data transfer
                AC3:  Hardware data mode
 5    .MOSRS    Set record size
                AC3:  Record size in bytes
```

```
 6      .MOFWR   Advance one record in read direction
 7      .MOBKR   Back up one record from read direction
10      .MOEOT   Advance to EOT (unlabeled) or EOV (labeled)
11      .MORUL   Rewind and unload tape (illegal for MOUNTed
                 tapes)
12      .MORDN   Return density
13      .MOERS   Erase tape gap (unlabeled only)
14      .MORDM   Return hardware data mode
15      .MORRS   Return record size
16      .MOFWF   Advance to next tape mark
17      .MOBKF   Backup to last tape mark or BOT
20      .MOSPR   Set parity
                 AC3:  Desired parity
                       0   .SJPRO  Odd parity
                       1   .SJPRE  Even parity
21      .MORPR   Return parity
22      .MONRB   Return number of bytes remaining in current
                 record
23      .MOFOU   Force output of partial records during
                 sequential write
24      .MOSDN   Set density (unlabeled only)
                 AC3:  Desired density
25      .MOINF   Return tape information
                 AC3:  Address of argblk
                       0   .MOICT  Word count not including
                                   this word
                       1   .MOITP  MTA type code
                       2   .MOIID  MTA reel ID
                       3   .MOISN  LH  Channel/controller/unit
                                   RH  Serial #
                       4   .MOIRD  Number of reads done
                       5   .MOIWT  Number of writes done
                       6   .MOIRC  Record number from BOT
                       7   .MOIFC  Number of files on tape
                      10   .MOISR  Number of soft read errors
                      11   .MOISW  Number of soft write errors
                      12   .MOIHR  Number of hard read errors
                      13   .MOIHW  Number of hard write errors
                      14   .MOIRF  Number of frames read
                      15   .MOIWF  Number of frames written
26      .MORDR   Return read direction
                 AC3:  0  Forwards
                       1  Backwards
27      .MOSID   Set reel ID of mounted tape (enabled WHL/OPR)
                 AC3:  36-bit reel ID
30      .MOIEL   Set error logging for tape
                 AC3:  0   Log errors
                       ≠0  Do not log errors
31      .MONOP   Wait for all activity to stop
32      .MOLOC   Identify 1st volume in MOUNT request or next
                 volume for volume switch (WHL/OPR)
                 AC3:  Pointer to argblk
                       0   .MOCNT  Word count
                       1   .MOMTN  MT unit # to associate with
                                   MTA
                       2   .MOLBT  Label type
```

```
                    3    .MODNS  Density
                    4    .MOAVL  Address of volume labels
                    5    .MONVL  # of volume labels at
                           .MOAVL
                    6    .MOCVN  Volume number in volume set
                    7    .MOVSN  SIXBIT file set identifier
37      .MOSTA  Return current magtape status
                AC3:  Address of argblk
                    0    .MOCNT  Word count including this
                                 word
                    1    .MODDN  Density flags (RET)
                                 B1(SJ%CP2)  200 BPI
                                 B2(SJ%CP5)  556 BPI
                                 B3(SJ%CP8)  800 BPI
                                 B4(SJ%C16)  1600 BPI
                                 B5(SJ%C62)  6250 BPI
                    2    .MODDM  Data mode flags (RET)
                                 B1(SJ%CMC)  Core dump
                                 B2(SJ%CM6)  SIXBIT
                                 B3(SJ%CMA)  ANSI ASCII
                                 B4(SJ%CM8)  Industry
                                             compatible
                                 B5(SJ%CMH)  High density
                                             mode
                    3    .MOTRK  Recording track flags (RET)
                                 B1(SJ%7TR)  7-track drive
                                 B2(SJ%9TR)  9-track drive
                    4    .MOCST  Tape status flags (RET)
                                 B0(SJ%OFS)  Off-line
                                 B1(SJ%MAI)  Maintenance
                                             mode enabled
                                 B2(SJ%MRQ)  Maintenance
                                             mode requested
                                 B3(SJ%BOT)  Beginning of
                                             tape
                                 B4(SJ%REW)  Rewinding
                                 B5(SJ%WLK)  Write locked
                    5    .MODVT  Device type (RET)
40      .MOOFL  Enable interrupts for on-line/off-line
                transition (WHL/OPR)
42      .MOPST  Set interrupt channel to indicate
                availability of UHL(BOV)/UTL(EOV) labels
                AC3:  PSI channel; -1 to clear
43      .MORVL  Rewind current labeled tape volume
44      .MOVLS  Switch volumes for unlabeled multi-volume set
                AC3:  Address of argblk
                    0    Word count including this word
                    1    Flags,,<function code>
                           1 .VSMNV  Mount absolute volume #
                           2 .VSFST  Mount 1st volume in set
                           3 .VSLST  Mount last volume in set
                           4 .VSMRV  Mount relative volume #
                           5 .VSFLS  Force volume switch
                                     (labeled only)
                    2    Volume number (if required)
45      .MONTR  Set translate flag (EBCDIC ==> ASCII; labeled
```

```
                only)
                AC3:  0   Clear translate flag
                      ≠0  Set translate flag
46      .MORDL  Read user header labels
                AC3:  Pointer to string for label
47      .MOWUL  Write user header or trailer labels (labeled
                only)
                AC3:  Byte pointer to label contents (must be
                      76 bytes)
                AC4:  Label identifier code (any ASCII char)
50      .MORLI  Read available fields from volume and header
                labels
                AC3:  Pointer to argblk
                       0  Word count
                       1  Label type (RET)
                          1 .LTUNL  unlabeled
                          2 .LTANS  ANSI
                          3 .LTEBC  EBCDIC
                          4 .LTT20  TOPS-20
                       2  Byte pointer to string for volume
                          name
                       3  Byte pointer to string for owner
                          name
                       4  Tape format (RET)
                       5  Record length (RET)
                       6  Block length (RET)
                       7  Creation date (RET)
                      10  Expiration date (RET)
                      11  Byte pointer to string for file
                          name
                      12  Generation number (RET)
                      13  Version number (RET)
                      14  Form-control value (RET)
                          SP  No line format characters
                          A   FORTRAN format control
                              characters
                          M   All necessary line format
                              characters
                          X   Data in stream mode
51      .MOSMV  Value for form-control field in HDR2 label
                AC3:  Mode
                      0   .TPFST  X
                      1   .TPFCP  M
                      2   .TPFFC  A
                      3   .TPFNC  Space
52      .MOSDS  Set deferred volume switch (labeled only)

PLPT FUNCTION CODES
Code  Symbol    Meaning/Arguments
27      .MOPSI  Enable software interrupt on nonfatal device
                conditions
                AC3:  Address of argblk
                      0  Word count including this word
                      1  Interrupt channel number
                      2  Flags
                         BO(MO%MSG)     Suppress CTY device
```

messages

31 .MONOP Wait for all activity to stop

32 .MOLVF Load line printer's VFU from file referenced in argblk
AC3: Address of argblk
0 Word count including this word
1 JFN of file containing VFU

33 .MORVF Read name of current VFU file in monitor's data base
AC3: Address of argblk
0 Word count including this word
1 Pointer to string for ASCIZ name
2 Number of bytes in string

34 .MOLTR Load line printer's translation RAM from file referenced in argblk
AC3: Address of argblk
0 Word count including this word
1 JFN of file containing translation RAM

35 .MORTR Read name of current translation RAM file in monitor's data base
AC3: Address of argblk
0 Word count including this word
1 Pointer to string for ASCIZ name
2 Number of bytes in string

36 .MOSTS Set status of line printer
AC3: Address of argblk
0 Word count including this word
1 Software status word
B0(MO%LCP) Printer is lowercase
B12(MO%EOF) Set MO%EOF when all data printed
B14(MO%SER) Clear software error condition
2 Value for page counter register

37 .MORST Read line printer status
AC3: Address of argblk
0 Word count including this word
1 Status word
B0(MO%LCP) Printer is lowercase
B1(MO%RLD) FE has been reloaded
B10(MO%FER) Fatal hardware error occurred
B12(MO%EOF) All data sent has been printed
B13(MO%IOP) Output in progress
B14(MO%SER) Software error occurred
B15(MO%HE) Hardware error occurred
B16(MO%OL) Printer is off-line
B17(MO%FNX) Printer does not exist
B30(MO%RPE) RAM parity error occurred
B31(MO%LVU) Printer has optical VFU

```
                        B33(MO%LVF)    VFU error occurred
                        B34(MO%LCI)    Character interrupt
                                       occurred
                        B35(MO%LPC)    Page counter register
                                       overflowed
                    2  Value of page counter register
40      .MOFLO   Flush any output not yet printed

PCDP FUNCTION CODES
Code   Symbol    Meaning/Arguments
27      .MOPSI   Enable software interrupt on nonfatal device
                 conditions
                 AC3:  Address of argblk
                       0  Word count including this word
                       1  Interrupt channel number
                       2  Flags
                          B0(MO%MSG)     Suppress CTY device
                                         messages
37      .MORST   Read card punch status
                 AC3:  Address of argblk
                       0  Word count including this word
                       1  Status word
                          B10(MO%FER)    Fatal error condition
                          B12(MO%EOF)    All pending output
                                         processed
                          B13(MO%IOP)    Output in progress
                          B14(MO%SER)    Software error
                                         occurred
                          B15(MO%HE)     Hardware error
                                         occurred
                          B16(MO%OL)     Card punch is off-line
                          B17(MO%FNX)    Punch doesn't exist
                          B32(MO%HEM)    Hopper empty or
                                         stacker full
                          B33(MO%SCK)    Stack check
                          B34(MO%PCK)    Pick check
                          B35(MO%RCK)    Read check

PCDR FUNCTION CODES
Code   Symbol    Meaning/Arguments
27      .MOPSI   Enable software interrupt on nonfatal device
                 conditions
                 AC3:  Address of argblk
                       0  Word count including this word
                       1  Interrupt channel number
                       2  Flags
                          B0(MO%MSG)     Suppress CTY device
                                         messages
37      .MORST   Read card reader status
                 AC3:  Address of argblk
                       0  Word count including this word
                       1  Status word
                          B0(MO%COL)     Card reader is online
                          B1(MO%RLD)     FE has been reloaded
                          B10(MO%FER)    Fatal hardware error
                                         occurred
```

```
                    B12(MO%EOF)   Card reader at EOF
                    B13(MO%IOP)   Input in progress
                    B14(MO%SER)   Software error
                                  occurred
                    B15(MO%HE)    Hardware error
                                  occurred
                    B16(MO%OL)    Reader is off-line
                    B17(MO%FNX)   Reader does not exist
                    B31(MO%SFL)   Output stacker full
                    B32(MO%HEM)   Input hopper empty
                    B33(MO%SCK)   Stack check
                    B34(MO%PCK)   Pick check
                    B35(MO%RCK)   Read check

PTY FUNCTION CODES
Code  Symbol    Meaning/Arguments
24    .MOAPI    Assign PTY interrupt channels
                AC2:  B0(MO%WFI)       Enable waiting-for-
                                       input interrupt
                      B1(MO%OIR)       Enable output-is-ready
                                       interrupt
                      B12-17(MO%SIC)   Interrupt channel for
                                       PTY output
                      B18-35           Function code
25    .MOPIH    Determine if PTY job needs input
26    .MOBAT    Set batch control bit
                AC3:  0  Job not under batch
                      1  Job under batch

TTY FUNCTION CODES
Code  Symbol    Meaning/Arguments
25    .MOPIH    Determine if TTY job needs input
26    .MOSPD    Set terminal line speed
                AC2:  B0(MO%RMT)  Remote line (WHL/OPR)
                      B1(MO%AUT)  Remote autobaud line
                                  (WHL/OPR)
                      B18-35      Function code
                AC3:  <input speed>,,<output speed>
27    .MORSP    Return terminal line speed
30    .MORLW    Return terminal page width
31    .MOSLW    Set terminal page width
                AC3:  Page width
32    .MORLL    Return terminal page length
33    .MOSLL    Set terminal page length
                AC3:  Page length
34    .MOSNT    Set terminal receive-system-messages code
                AC3:  0(.MOSMY)   Allow messages
                      1(.MOSMN)   Refuse messages
35    .MORNT    Return terminal receive-system-messages code
36    .MOSIG    Set terminal input on inactive line code
                AC3:  0  Do not ignore input
                      1  Ignore input
37    .MORBM    Read 128-character break mask
40    .MOSBM    Set 128-character break mask
                AC3:  Address of argblk
                       0  Word count not including this word
```

```
                        1-4 Break character mask
41      .MORFW  Return current value of field width
42      .MOSFW  Set field width
                AC3:  Field width
43      .MOXOF  Set pause-at-end-of-page mode
                AC3:  0(.MOOFF)   Disable pause-at-end-of-
                                  page mode
                      1(.MOONX)   Enable pause-at-end-of-page
                                  mode
44      .MORXO  Read end-of-page mode
45      .MOSLC  Set terminal's line counter
                AC3:  Line counter value
46      .MORLC  Read terminal's line counter
47      .MOSLM  Set line maximum
                AC3:  Line maximum value
50      .MORLM  Read line maximum
51      .MOTPS  Assign terminal interrupt channels
                AC3:  Address of argblk
                      0  Word count including this word
                      1  B0-17   Output PSI channel
                         B18-35  Input PSI channel
52      .MOPCS  Set terminal pause/unpause characters
                AC3:  <pause character>,,<unpause character>
53      .MOPCR  Read terminal pause/unpause characters
```

MTU% JSYS 600

FUNCTION
Allows privileged programs to perform various utility functions for magnetic-tape MT: devices.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability.

CALLING SEQUENCE
AC1: Function code
AC2: MT unit number
AC3: Address of argblk

RETURNS +1: Always

FUNCTION CODES

```
Code  Symbol    Meaning/Arguments
 1    .MTNNV    Declare volume switch error
                0  .MTCNT Word count
                1  .MTCOD Error code to return to user
                2  .MTPTR Byte pointer to operator response
 2    .MTRAL    Read labels
                0  .MTCNT Word count
                1  .MTVL1 Byte pointer to area for VOL1 label
                2  .MTVL2 Byte pointer to area for VOL2 label
                3  .MTHD1 Byte pointer to area for HDR1 label
                4  .MTHD2 Byte pointer to area for HDR2 label
 3    .MTASI    Return assignment information
```

```
                0   .MTCNT Word count
                1   .MTPHU Returned MTA # associated with MT
 4      .MTCVV   Clear volume ID for specified MT
```

MUTIL JSYS 512

```
FUNCTION
Performs various IPCF (Inter-Process Communication Facility)
functions.

RESTRICTIONS
Some functions require WHEEL, OPERATOR, or IPCF capability
enabled.

CALLING SEQUENCE
AC1:  Length of argblk
AC2:  Address of argblk

RETURNS    +1:  Failure, error code in AC1
           +2:  Success, with requested data in argblk

ARGUMENT BLOCK
Word   Contents
 0     Function code
1-n    Function-specific arguments

FUNCTION CODES
Code   Symbol    Meaning/Arguments
 1     .MUENB    Enable specified PID to receive packets
                 1   PID
 2     .MUDIS    Disable specified PID from receiving packets
                 1   PID
 3     .MUGTI    Return PID associated with <SYSTEM>INFO
                 1   PID or job #
 4     .MUCPI    Create private copy of <SYSTEM>INFO for job
                 (enabled IPCF)
                 1   PID to be assigned to <SYSTEM>INFO
                 2   PID or job # creating private copy
 5     .MUDES    Delete specified PID
                 1   PID
 6     .MUCRE    Create PID for specified process or job
                 1   Flags,,<process handle or job #>
                     B6(IP%JWP)    PID is job-wide
                     B7(IP%NOA)    PID is not available to
                                   other processes
 7     .MUSSQ    Set send/receive quotas for specified PID
                 (enabled IPCF)
                 1   PID
                 2   B18-26        New send quota
                     B27-35        New receive quota
10     .MUCHO    Change job # associated with specified PID
                 (enabled WHL)
                 1   PID
                 2   New job # or PID belonging to new job
```

```
11      .MUFOJ   Return job # associated with specified PID
                 1   PID
12      .MUFJP   Return all PIDs associated with specified job
                 1   Job # or PID belonging to job
13      .MUFSQ   Return send/receive quotas for specified PID
                 1   PID
15      .MUFFP   Return all PIDs associated with same process
                 as given PID
                 1   PID
16      .MUSPQ   Set maximum number of PIDs allowed for job
                 (enabled IPCF)
                 1   Job # or PID
                 2   PID quota
17      .MUFPQ   Return maximum number of PIDs allowed for job
                 1   Job # or PID
20      .MUQRY   Return Packet Descriptor Block for next
                 packet in queue associated with specified PID
                 1   PID
                 2   -1   Next descriptor block for process
                     -2   Next descriptor block for job
21      .MUAPF   Associate PID with specified process
                 1   PID
                 2   Process handle
22      .MUPIC   Place specified PID on software interrupt
                 channel
                 1   PID
                 2   Channel number; -1 to remove PID
23      .MUDFI   Set PID of <SYSTEM>INFO (enabled IPCF)
                 1   PID of <SYSTEM>INFO
24      .MUSSP   Place specified PID into system PID table at
                 offset (enabled WHL/OPR/IPCF)
                 1   Index into system PID table
                 2   PID
25      .MURSP   Return PID from system PID table 1 Index into
                 system PID table
26      .MUMPS   Return system-wide maximum packet size
27      .MUSKP   Set PID to receive deleted PID messages
                 1   Source (subordinate) PID
                 2   Object (controller) PID
30      .MURKP   Return controlling PID for this subordinate
                 PID
                 1   Source (subordinate) PID
                 2   Object (controller) PID (RET)
```

NIN JSYS 225

FUNCTION
Inputs an integer, with leading spaces ignored.

CALLING SEQUENCE
AC1: Source designator
AC3: Radix (2-10) of number being input

RETURNS +1: Failure, with

```
                AC1:  Updated byte pointer
                AC3:  Error code
          +2:   Success, with
                AC1:  Updated byte pointer
                AC2:  Number input
```

NODE JSYS 567

FUNCTION
Performs network utility functions.

RESTRICTIONS
Some functions require WHEEL, OPERATOR, or MAINTENANCE capability.

CALLING SEQUENCE
AC1: Function code
AC2: Address of argblk

RETURNS +1: Always

FUNCTION CODES

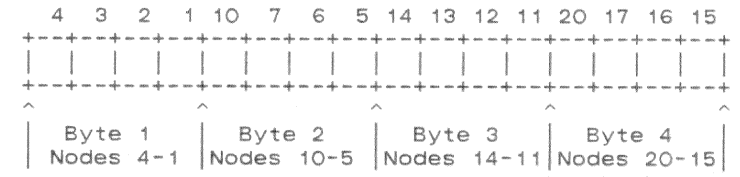
Code	Symbol	Meaning/Arguments
0	.NDSLN	Set local node name (WHL/OPR) 0 .NDNOD Byte pointer to ASCIZ node name
1	.NDGLN	Get local node name 0 .NDNOD Byte pointer to string for ASCIZ node name
2	.NDSNM	Set local node number (WHL/OPR) 0 .NDNOD Number to set (from 1 to .NDMAX)
3	.NDGNM	Get local node number 0 .NDNOD Local node number
4	.NDSLP	Set loopback port (KS-10 only; WHL/OPR/MNT) 0 .NDPRT NSP port number
5	.NDCLP	Clear loopback port (KS-10 only; WHL/OPR/MNT) 0 .NDPRT NSP port number
6	.NDFLP	Find loopback port (KS-10 only) 0 .NDPRT Flags,,<NSP port number> (RET) B0(ND%LPR) Loopback running B1(ND%LPA) Loopback port assigned
7	.NDSNT	Set node table (WHL/OPR) 0 .NDNNO # of nodes in topology message 1 .NDMSK Address of topology message
10	.NDGNT	Get node table 0 .NDNND 0,,<word count> (1/node); On return <# returned nodes>,,<word count> 1 .NDCNT # of words in node block (RET) 2 .NDBK1 Addresses of n node blocks (1/returned node; RET)
11	.NDSIC	Set topology interrupt channel 0 .NDCHN Channel # for topology interrupts
12	.NDCIC	Clear topology interrupt channel
13	.NDGVR	Get NSP version number

```
                0   .NDNVR  Number of versions (RET)
                1   .NDCVR  Address of block for NSP
                            communications version
                2   .NDRVR  Address of block for NSP routing
                            version
14      .NDGLI  Get line information
                0   .NDNLN  0,,<word count> (1/line);
                            On return
                            <# returned lines>,,<word count>
                1   .NDBK1  Addresses of n line blocks
                            (1/returned line; RET)
15      .NDVFY  Verify node name
                0   .NDNOD  Byte pointer to ASCIZ node name
                1   .NDFLG  Flags returned by monitor
                             BO(ND%EXM)   Node exactly matches
                                          name in monitor's
                                          database
16      .NDRNM  Return node name
                0   .NDNOD  Node number
                1   .NDCVR  Byte pointer to string for ASCIZ
                            node name
```

NODE BLOCK

Word	Symbol	Contents
0	.NDNAM	Byte pointer to ASCIZ node name
1	.NDSTA	Node state: .NDSON On, add to table of reachable nodes if not there .NDSOF Off, remove from table if there
2	.NDNXT	Obsolete
3-4	--	ASCIZ node name (word 4 not returned if name .LE. 4 characters)

TOPOLOGY MESSAGE



Value	Meaning
00	Node not reachable
01	Reserved
10	Reachable Phase II node
11	Reachable Phase III node

NSP VERSION BLOCK

Word	Symbol	Contents
0	.NDVER	Version number
1	.NDECO	ECO number
2	.NDCST	Customer change order

```
LINE BLOCK
Word  Symbol    Contents
0     .NDLNM    Line number
1     .NDLST    State of Line
                .NDLON    On
                .NDLOF    Off
                .NDLCN    Controller loopback
                .NDLCB    Cable loopback
                .NDLND    Byte pointer to ASCIZ remote node
                          name
```

NOUT JSYS 224

```
FUNCTION
Outputs an integer number.

CALLING SEQUENCE
AC1:  Destination designator
AC2:  Number to be output
AC3:  B0(NO%MAG)      Output magnitude only
      B1(NO%SGN)      Output + before positive number
      B2(NO%LFL)      Output leading filler
      B3(NO%ZRO)      Output 0's as leading filler
      B4(NO%OOV)      Output on column overflow and return an
                      error
      B5(NO%AST)      Output asterisks on column overflow
      B11-17(NO%COL)  Number of columns to output
      B18-35(NO%RDX)  Radix (2-36) of number being output

RETURNS   +1:  Failure, error code in AC3
          +2:  Success, updated byte pointer in AC1, if
               pertinent
```

NTMAN% JSYS 604

```
FUNCTION
Provides an interface between the DECnet-20 Network
Management layer and lower layers of the Digital Network
Architecture.

RESTRICTIONS
Requires WHEEL or OPERATOR capability.

CALLING SEQUENCE
AC1:  Address of argblk

RETURNS   +1   Always, with error code in AC1

ARGUMENT BLOCK
Word  Symbol    Contents
 0    .NTCNT    Word count including this word
 1    .NTENT    Entity on which to perform function
```

```
                    0    .NTNOD  Node
                    1    .NTLIN  Line
                    2    .NTLOG  Logging
                    3    .NTCKT  Circuit
                    4    .NTMOD  Module
 2      .NTEID      Byte pointer to entity ID
 3      .NTFNC      Function code
                   -2   .NTMAP  Map node number/node name
                   -1   .NTREX  Return local node ID
                    0   .NTSET  Set Parameter
                    1   .NTCLR  Clear Parameter
                    2   .NTZRO  Zero all counters
                    3   .NTSHO  Show selected items
                    4   .NTSZC  Show and zero all counters
                    5   .NTRET  Return list of items
 4      .NTSEL      Selection criterion for function
                    Selectors for .NTSHO
                    0    .NTSUM  Summary
                    1    .NTSTA  Status
                    2    .NTCHA  Characteristics
                    3    .NTCOU  Counters
                    4    .NTEVT  Event
                    Selectors for .NTRET
                   -1   .NTKNO  Known items
                   -2   .NTACT  Active items
                   -3   .NTLOP  Loop
 5      .NTQUA      Byte pointer to function qualifier
 6      .NTBPT      Byte pointer to parameter data buffer
 7      .NTBYT      Parameter data buffer length in bytes
                    (functions .NTMAP, .NTRET, .NTREX, .NTSHO,
                    and .NTSZC)
10      .NTERR      Network management return code
```

ODCNV JSYS 222

FUNCTION

Converts internal date and time format into separate numbers for local weekday, day, month, year, and time and does not convert the numbers to text.

CALLING SEQUENCE

```
AC2:  Internal date/time, or -1 for current date/time
AC4:  B0(IC%DSA)       Apply daylight savings according to
                       B1(IC%ADS)
      B1(IC%ADS)       Apply daylight savings if 1B0(IC%DSA)
      B2(IC%UTZ)       Use time zone in B12-B17(IC%TMZ)
      B3(IC%JUD)       Apply Julian day format
      B12-17(IC%TMZ)   Time zone to use if 1B2(IC%UTZ)
```

```
RETURNS   +1:  Always, with
               AC2:  Year,,<numerical month>  or
                     Year,,<Julian day> if IC%JUD
               AC3:  <day of month>,,<day of week>  or
                     0,,<day of week> if IC%JUD
```

AC4:	B0, B2	On for compatibility with IDCNV
	B1(IC%ADS)	If daylight savings was applied
	B3(IC%JUD)	If Julian day format was applied
	B12-17(IC%TMZ)	Time zone used
	B18-35(IC%TIM)	Local time in seconds since midnight

ODTIM JSYS 220

FUNCTION
Converts the internal date and/or time to text.

CALLING SEQUENCE
AC1: Destination designator
AC2: Internal date/time, or -1 for current date/time
AC3: Format option flags; or
0 for format: dd-mmm-yy hh:mm:ss; or
-1 for format: weekday, month day, year hh:mm:ss

RETURNS +1: Always, with updated byte pointer in AC1

FORMAT OPTION FLAGS

B0(OT%NDA)	Do not output date and ignore B1-8
B1(OT%DAY)	Output day of week according to B2(OT%FDY)
B2(OT%FDY)	Output full text for day of week
B3(OT%NMN)	Output month as numeric and ignore B4(OT%FMN)
B4(OT%FMN)	Output full text for month
B5(OT%4YR)	Output year as a 4-digit number
B6(OT%DAM)	Output day of month after month
B7(OT%SPA)	Output day month year with space delimiter; if 1B6(OT%DAM), output month day, year
B8(OT%SLA)	Output numeric date with slash delimiter; if OB7 and OB8, output day-month-year with dash delimiter
B9(OT%NTM)	Do not output time and ignore B10-13
B10(OT%NSC)	Do not output seconds
B11(OT%12H)	Output time in 12-hour format with AM or PM
B12(OT%NCO)	Output time without colon between hours and minutes
B13(OT%TMZ)	Output time with "-" and time zone
B17(OT%SCL)	Suppress columnization of date and time (omit leading spaces and zeros)

ODTNC JSYS 230

FUNCTION
Outputs the date and/or the time as separate numbers for local year, month, day, or time.

CALLING SEQUENCE
AC1: Destination designator
AC2: Year,,<numerical month>
AC3: <day of month>,,<day of week>
AC4: B1(IC%ADS) Apply daylight savings on output
B12-17(IC%TMZ) Time zone desired
B18-35(IC%TIM) Local time in seconds since midnight
AC5: Format option flags

RETURNS +1: Always, with updated byte pointer in AC1

FORMAT OPTION FLAGS

B0(OT%NDA)	Do not output date and ignore B1-8
B1(OT%DAY)	Output day of week according to B2(OT%FDY)
B2(OT%FDY)	Output full text for day of week
B3(OT%NMN)	Output month as numeric and ignore B4(OT%FMN)
B4(OT%FMN)	Output full text for month
B5(OT%4YR)	Output year as a 4-digit number
B6(OT%DAM)	Output day of month after month
B7(OT%SPA)	Output day month year with space delimiter; if 1B6(OT%DAM), output month day, year
B8(OT%SLA)	Output numeric date with slash delimiter; if OB7 and OB8, output day-month-year with dash delimiter
B9(OT%NTM)	Do not output time and ignore B10-13
B10(OT%NSC)	Do not output seconds
B11(OT%12H)	Output time in 12-hour format with AM or PM
B12(OT%NCO)	Output time without colon between hours and minutes
B13(OT%TMZ)	Output time with "-" and time zone (US zones and Greenwich Mean only)
B17(OT%SCL)	Suppress columnization of date and time (omit leading spaces and zeros)

OPENF JSYS 21

FUNCTION
Opens the given file.

CALLING SEQUENCE
AC1: O,,JFN
AC2: B0-5(OF%BSZ) Byte size (maximum of 36.; 36. default)
B6-9(OF%MOD) Data mode in which to open file
B18(OF%HER) Halt on I/O, device, or data error
B19(OF%RD) Allow read access
B20(OF%WR) Allow write access
B21(OF%EX) Allow execute access
B22(OF%APP) Allow append access

B23(OF%RDU)	Allow unrestricted read access (illegal with OF%THW or OF%WR)
B25(OF%THW)	Allow thawed access
B26(OF%AWT)	Block and wait for access to be granted
B27(OF%PDT)	Do not update access dates of file
B28(OF%NWT)	Do not wait if access disallowed; return error
B29(OF%RTD)	Enforce restricted access
B30(OF%PLN)	Disable line number checking
B31(OF%DUD)	Suppress system updating of modified pages in memory to thawed files on disk unless CLOSF or UFPGS issued
B32(OF%OFL)	Open device even if off-line
B33(OF%FDT)	Force update of .FBREF (last read) in FDB and increment RH of .FBCNT (number of references)
B34(OF%RAR)	Wait if file off-line

RETURNS +1: Failure, error code in AC1
+2: Success

PBIN JSYS 73

FUNCTION
Inputs the next sequential byte from the primary input designator.

RETURNS +1: Always, with the byte R-J in AC1

PBOUT JSYS 74

FUNCTION
Outputs a byte sequentially to the primary output designator.

CALLING SEQUENCE
AC1: Byte to be output, right-justified

RETURNS +1: Always

PDVOP% JSYS 603

FUNCTION
Manipulates program data vectors (PDVs), using program data vector addresses (PDVAs).

CALLING SEQUENCE
AC1: Function code
AC2: Address of argblk
AC3: Byte pointer to string in memory

RETURNS +1: Always, with data returned in the data block, and updated count in .POCT2 if needed

FUNCTION CODES

Code	Symbol	Meaning
0	.POGET	Return all PDVAs within range specified in argblk
1	.POADD	Add PDVAs specified in data block to system's database for process
2	.POREM	Remove PDVAs within range specified in argblk from system's data base for process
3	.PONAM	Return ASCIZ name of program referenced in word .PVNAM of PDV
4	.POVER	Return program version number from word .PVVER of PDV
5	.POLOC	Return all PDVAs of PDVs for program referenced in AC3

ARGUMENT BLOCK

Word	Symbol	Meaning
0	.POCT1	Number of words in argblk
1	.POPHD	Handle of desired process
2	.POCT2	# of words in data block; on return <# of words available>,,<# of words returned>
3	.PODAT	Starting address of data block for returned data
4	.POADR	Starting address of memory range
5	.POADE	Ending address of memory range

FORMAT OF PROGRAM DATA VECTOR

Word	Symbol	Meaning
0	.PVCNT	Length of PDV including this word
1	.PVNAM	Pointer to ASCIZ program name string for this PDV
2	.PVSTR	Program starting address
3	.PVREE	Program reenter address
4	.PVVER	Program version number
5	.PVMEM	Address of block of memory containing length in Word 0 and memory map in remaining words
6	.PVSYM	Address of program symbol table
7	.PVCTM	Time at which program was compiled
10	.PVCVR	Version number of compiler
11	.PVLTM	Time at which program was loaded
12	.PVLVR	Version number of LINK
13	.PVMON	Address of monitor data block (not used currently)
14	.PVPRG	Address of program data block (not used currently)
15	.PVCST	Address of customer-defined data block

PEEK JSYS 311

FUNCTION
Transfers a block of words from the monitor to the user space.

RESTRICTIONS
Requires enabled WHEEL, OPERATOR, or MAINTENANCE capability.

CALLING SEQUENCE
AC1: <word count>,,<1st virtual address of monitor>
AC2: 1st user address

RETURNS +1: Failure, error code in AC1
+2: Success

PLOCK JSYS 561

FUNCTION
Locks physical memory and places a designated section of the process's address space in memory.

RESTRICTIONS
Requires enabled WHEEL, OPERATOR, or MAINTENANCE capabilities.

CALLING SEQUENCE
AC1: Address of 1st page if locking; -1 if unlocking
AC2: <process handle>,,<# of 1st page>
AC3: <control flags>,,<repeat count>
B0(LK%CNT) B18-35 of AC3 contain # of pages to lock
B1(LK%PHY) AC1 contains 1st page desired
B2(LK%NCH) Pages will not be cached
B3(LK%AOL) Off-line pages are to be locked

RETURNS +1: Always

PMAP JSYS 56

FUNCTION
Maps one or more complete pages from a file to a process (Case I), from a process to a file (Case II), or from one process to another process (Case III); or unmaps pages from a process (Case IV) and deletes pages from a file (Case V).

CALLING SEQUENCE
AC1: JFN,,<file page #> (Case I)
<source process handle>,,<process page #> (Cases II & III)
-1 (Cases IV & V)
AC2: <destination process handle>,,<process page #> (Cases I & III)

```
        <destination JFN>,,<file page #> (Case II)
        <process handle>,,<process page #> (Case IV)
        JFN,,<file page #> (Case V)
AC3:    B0(PM%CNT)      B18-35 contain repeat count
        B2(PM%RD)       Permit read access (Cases I - III only)
        B3(PM%WR)       Permit write access (Cases I - III
                        only)
        B4(PM%EX)       Reserved
        B5(PM%PLD)      Preload page being mapped (Cases I -
                        III only)
        B9(PM%CPY)      Create private copy of page (Cases I -
                        III only)
        B10(PM%EPN)     B18-35 of AC2 contain extended (18-bit)
                        process page number (Cases I - III
                        only)
        B11(PM%ABT)     Unmap page and discard changed contents
                        (Cases I - III only)
        B18-35(PM%RPT) # of pages to map if 1B0(PM%CNT)

RETURNS    +1:  Always
```

PMCTL JSYS 560

FUNCTION
Controls physical memory, allowing a privileged program to add or remove most pages of physical memory and to control use of cache memory.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability.

CALLING SEQUENCE
AC1: Function code
AC2: Length of argblk
AC3: Address of argblk

RETURNS +1: Always

FUNCTION CODES

Code	Symbol	Meaning/Arguments
0	.MCRCE	Return status of cache memory
		0 .MCCST If 1B35(MC%CEN), cache is enabled
1	.MCSCE	Set status of cache memory
		0 .MCCST Enable cache if 1B35(MC%CEN)
2	.MCRPS	Return status of specified page
		0 .MCPPN - count,,<physical page #>
		1 .MCPST Returned page status
		0 .MCPSA Page available
		1 .MCPSS Page in transition
		2 .MCPSO Page off-line (nonexistant)
		3 .MCPSE Page off-line due to error
3	.MCSPS	Set status of specified page

```
                0   .MCPPN  Physical page number
                1   .MCPST  Status for page
                            0   .MCPSA  Mark page available
                            1   .MCPSS  Mark page in transition
                            2   .MCPSO  Mark page off line
                            (nonexistant)
                            3   .MCPSE  Mark page off line due
                            to error
4       .MCRME  Return information about MOS memory errors
                0   .PMMTP  <1B8!<count>B17>,,<controller #>
                1   .PMMRG  Error register at error
                2   .PMMSY  Syndrome of error
                3   .PMMBN  Block number of error
                4   .PMMSB  Spare bit number
                5   .PMMEA  Error address
                6   .PMMSN  4 words of 32-bit PROM  serial
                            numbers
```

PPNST JSYS 557

FUNCTION
Translates a project-programmer number (PPN, a TOPS-10 36-bit directory designator) to its corresponding TOPS-20 string.

CALLING SEQUENCE
AC1: Destination designator
AC2: Project-programmer number (36-bit)
AC3: Byte pointer to structure name string for which given PPN applies

RETURNS +1: Always, with updated byte pointer in AC1

PRARG JSYS 545

FUNCTION
Returns or sets up an argument block for the specified process.

CALLING SEQUENCE
AC1: <function code>,,<process handle>
AC2: Address of argblk
AC3: Length of argblk

RETURNS +1: Always, with number of returned words in AC3

FUNCTION CODES

Code	Symbol	Meaning
1	.PRARD	Return arguments in argblk
2	.PRAST	Set arguments from argblk

ARGUMENT BLOCK

Word	Meaning
0	Number of argblks
1 - n	Argument pointers
n+1	Data

ARGUMENT POINTER

Bit	Contents
B0	1
B1-3	0
B4-6	Data structure type 0 Scalar or array without dope vector 1 Array with dope vector 3 Immediate (data in B18-35)
B7-12	Type code 00 Unspecified 02 Integer 04 Real 17 ASCIZ string
B13-17	0
B18-35	Data offset in block or data; -1 for last LOAD-class command

PSOUT JSYS 76

FUNCTION
Outputs a string sequentially to the primary output designator.

CALLING SEQUENCE
AC1: Byte pointer to ASCIZ string

RETURNS +1: Always, with updated byte pointer in AC1

RCDIR JSYS 553

FUNCTION
Translates the given directory string to its corresponding 36-bit directory number.

RESTRICTIONS
In non-zero sections, OWGBPs must specify 7-bit bytes.

CALLING SEQUENCE
AC1: <flag bits>,,0
AC2: Byte pointer to ASCIZ string (to obtain 36-bit directory number)
JFN (to obtain directory number associated with file)
36-bit user number (to obtain logged-in directory)
36-bit directory number (to check validity)
AC3: 36-bit directory number (to use RCDIR to step through directory string with wildcards)

```
RETURNS   +1:  Always, with
               AC1:  <flag bits>,,0
               AC2:  Updated byte pointer (if pointer was
                     supplied)
               AC3:  36-bit directory number

FLAGS SUPPLIED IN RCDIR CALL
Bit   Symbol   Meaning
B14   RC%PAR   Allow partial recognition on directory name
B15   RC%STP   Step to next directory in group and return
               number
B16   RC%AWL   Allow directory name to contain wildcards
B17   RC%EMO   Match given string exactly

FLAGS RETURNED FROM RCDIR CALL
Bit   Symbol   Meaning
B0    RC%DIR   Directory is files-only
B1    RC%ANA   Obsolete
B2    RC%RLM   User sees all messages from <SYSTEM>MAIL.TXT
               on login
B3    RC%NOM   No match was found for string
B4    RC%AMB   String given was ambiguous
B5    RC%NMD   No more directories in group
B6    RC%WLD   Directory name contained wildcards
```

RCM JSYS 134

FUNCTION
Returns the word mask of the activated interrupt channels for the specified process.

CALLING SEQUENCE
AC1: Process handle

```
RETURNS   +1:  Always, with
               AC1:  36-bit word (1Bn indicates channel n
                     activated)
```

RCUSR JSYS 554

FUNCTION
Translates the given user name string to its corresponding 36-bit user number.

RESTRICTIONS
Directory may not be files-only.

CALLING SEQUENCE
AC1: <flag bits>,,0
AC2: Byte pointer to ASCII username string
AC3: 36-bit user number (if stepping to next username in group)

```
RETURNS    +1:  Always, with
                AC1:  <flag bits>,,0
                AC2:  Updated byte pointer
                AC3:  36-bit user number

FLAGS SUPPLIED ON CALL
Bit    Symbol   Meaning
B14    RC%PAR   Allow partial recognition on username string
B15    RC%STP   Step to next username in group
B16    RC%AWL   Allow username to contain wildcards
B17    RC%EMO   Match given string exactly

FLAGS RETURNED FROM CALL
Bit    Symbol   Meaning
B1     RC%ANA   Obsolete
B2     RC%RLM   User sees all messages from <SYSTEM>MAIL.TXT
                on login
B3     RC%NOM   No match was found for string
B4     RC%AMB   String given was ambiguous
B5     RC%NMD   No more usernames in group
B6     RC%WLD   Username given contained wildcards
```

RCVIM JSYS 751

FUNCTION
Retrieves a message from the ARPANET special message queue.

RESTRICTIONS
For ARPANET systems only.

CALLING SEQUENCE

```
AC1:  B0        If on, leader is 96-bit; if off, leader is
                32-bit
      B1        If on, 32-bit data in each word of message
                (high-order); if off, 36-bit data in each
                word
      B18-35    Special queue header
AC2:  Address for storing extended message

RETURNS    +1:  Failure, error code in AC1
           +2:  Success
```

RCVOK% JSYS 575

FUNCTION
Allows installation-supplied access-control program to service an approval request in the GETOK% request queue.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability.

```
CALLING SEQUENCE
AC1:  Address of argblk
AC2:  Length of argblk

RETURNS    +1:  Always

ARGUMENT BLOCK (RET)
Word   Symbol    Contents
 0     .RCFCJ    <GETOK% function code>,,<job # of requestor>
 1     .RCUNO    User number
 2     .RCCDR    Connected directory
 3     .RCRQN    Request number
 4     .RCNUA    B0-17   # arguments passed to RCVOK% block
                 B18-35  # user arguments in user block
 5     .RCARA    Address of user arguments
 6     .RCCAP    Capabilities enabled
 7     .RCTER    Controlling terminal number; or -1 for
                 detached job
10     .RCRJB    Requested job #
11               User arguments
11+n             ..
```

RDTTY JSYS 523

```
FUNCTION
Reads input from the primary input designator into the
caller's address space.

CALLING SEQUENCE
AC1:  Byte pointer to string for input
AC2:  <flag bits>,,<# of bytes in string>
      0,,<# of bytes in string> to break on EOL only
      B0(RD%BRK)   Break on CTRL/Z or ESC
      B1(RD%TOP)   Break on CTRL/G, CTRL/L, CTRL/Z, ESC, CR,
                   LF
      B2(RD%PUN)   Break on punctuation:
                   CTRL/A-CTRL/F  CTRL/H-CTRL/I CTRL/K
                   CTRL/N-CTRL/Q  CTRL/S-CTRL/T
                   CTRL/X-CTRL/Y
                   ASCII codes 34-36, 40-57, 72-100,
                   133-140, 173-176
      B3(RD%BEL)   Break on EOL (CRLF or LF only)
      B4(RD%CRF)   Suppress CR and return LF only
      B5(RD%RND)   Return if attempt to delete past
                   beginning of input buffer
      B7(RD%RIE)   Return if input buffer empty
      B9(RD%BEG)   Return if attempt to edit past beginning
                   of input buffer
      B10(RD%RAI)  Convert lowercase input to uppercase
      B11(RD%SUI)  Suppress CTRL/U indication
AC3:  Byte pointer to CTRL/R buffer; 0 if no reprompt text

RETURNS    +1:  Failure, error code in AC1
           +2:  Success, with
```

AC1: Updated byte pointer
AC2: <flag bits>,,<updated byte count>
B12(RD%BTM) Break character terminated input
B13(RD%BFE) Input buffer empty
B14(RD%BLR) Backup limit for editing reached

RELD JSYS 71

FUNCTION
Releases one or all devices assigned to the job.

CALLING SEQUENCE
AC1: Device designator; -1 to release all assigned devices devices assigned to this job

RETURNS +1: Failure, error code in AC1
+2: Success

RELSQ JSYS 753

FUNCTION
Deassigns the ARPANET special message queue, and discards all pending messages.

RESTRICTIONS
For ARPANET systems only.

CALLING SEQUENCE
AC1: Special queue handle (RET by ASNSQ); -1 to deassign all special queues

RETURNS +1: Always

RESET JSYS 147

FUNCTION
Closes all files at or below the current process and releases all JFNs; kills all inferior processes; clears the PSI for the current process; sets TT%WAK, TT%ECO, and .TTASI of the controlling terminal's JFN mode word; releases all PIDs of the current process; dequeues all ENQ requests for the current process, clears PA1050's entry vector; and, releases all process handles inferior to the current process or killed with KFORK.

RETURNS +1: Always

RFACS JSYS 161

FUNCTION
Returns the ACs of the specified process.

CALLING SEQUENCE
AC1: Process handle
AC2: Address of 20-word block to store AC values of specified process

RETURNS +1: Always

RFBSZ JSYS 45

FUNCTION
Returns the byte size for a specific opening of a file.

CALLING SEQUENCE
AC1: JFN

RETURNS +1: Failure, error code in AC1
+2: Success, byte size R-J in AC2

RFCOC JSYS 112

FUNCTION
Returns the control character output control (CCOC) words for the specified terminal.

CALLING SEQUENCE
AC1: File designator

RETURNS +1: Always, with CCOC words in AC2 and AC3

RFMOD JSYS 107

FUNCTION
Returns the JFN mode word associated with the specified file.

CALLING SEQUENCE
AC1: Source designator

RETURNS +1: Always, with mode word in AC2

RFORK JSYS 155

FUNCTION
Resumes one or more processes that have been directly frozen.

CALLING SEQUENCE
AC1: Process handle

RETURNS +1: Always

RFPOS JSYS 111

FUNCTION
Returns the current position of the specified terminal's cursor.

CALLING SEQUENCE
AC1: Device designator

RETURNS +1: Always, with
AC2: <line number>,,<column number>
0 if designator not terminal

RFPTR JSYS 43

FUNCTION
Returns the current position of the specified file's pointer.

CALLING SEQUENCE
AC1: JFN

RETURNS +1: Failure, error code in AC1
+2: Success, byte number in AC2

RFRKH JSYS 165

FUNCTION
Releases the specified process handle if the process is inferior to at least one other process in the job or has been killed with KFORK.

CALLING SEQUENCE
AC1: Process handle; -1 for all

RETURNS +1: Failure, error code in AC1
+2: Success

RFSTS JSYS 156

FUNCTION
Returns the status of the specified process.

CALLING SEQUENCE
```
AC1:  0,,<process handle> (short form)
      flags,,<process handle> (long form)
      B0    RF%LNG  Long form call
      B1-17         Unused, must be zero
AC2:  Address of status return block (long form only)
```

```
RETURNS   +1:  Always, with
               AC1:  Status word  (short form only)
               AC2:  Process PC flags (short form only)
               AC3:  -1 if process deleted (short form only)
```

PROCESS STATUS WORD

Bit	Symbol	Meaning
B0	RF%FRZ	Process is frozen
B1-17	RF%STS	Status code for process
	0 .RFRUN	Process is runnable
	1 .RFIO	Process is dismissed for I/O
	2 .RFHLT	Process dismissed by HFORK or HALTF or never started
	3 .RFFPT	Process dismissed by forced process termination
	4 .RFWAT	Process dismissed waiting for another process to terminate
	5 .RFSLP	Process dismissed for specified amount of time
	6 .RFTRP	Process dismissed because intercepted by superior
	7 .RFABK	Process dismissed because address break encountered
B18-35	RF%SIC	Number of software interrupt channel causing forced process termination

STATUS-RETURN BLOCK (Long Form Only)

Word	Symbol	Meaning
0	.RFCNT	<returned word count>,,<max. words desired> (RH user specified)
1	.RFPSW	Process status word; -1 if unassigned process handle in AC1
2	.RFPFL	Process PC flags
3	.RFPPC	Process PC
4	.RFSFL	Status flag word B0 RF%EXO Process is execute-only

RFTAD JSYS 533

FUNCTION
Returns the dates and times associated with the specified file.

CALLING SEQUENCE
AC1: Source designator
AC2: Address of argblk
AC3: Length of argblk

RETURNS +1: Always, with dates returned in argblk

ARGUMENT BLOCK

Word	Symbol	Meaning
0	.RSWRT	Internal date and time file was last written
1	.RSCRV	Internal date and time file was created
2	.RSREF	Internal date and time file was last referenced
3	.RSCRE	System date and time of last write by monitor
4	.RSTDT	Tape-write date and time for archived or migrated files
5	.RSNET	Online expiration date and time
6	.RSFET	Offline expiration date and time

RIN JSYS 54

FUNCTION
Inputs a non-sequential (random) byte from the specified file.

RESTRICTIONS
Disk file only.

CALLING SEQUENCE
AC1: JFN
AC3: Byte number within file

RETURNS +1: Always, with byte R-J in AC2; 0 if EOF

RIR JSYS 144

FUNCTION
Returns the channel and priority level table addresses for the specified process.

RESTRICTIONS
Process must run in section zero; for multiple-section processes use XRIR%.

```
CALLING SEQUENCE
AC1:  Process handle

RETURNS    +1:  Always, with
                AC2:  <LEVTAB address>,,<CHNTAB address>
                      0 if no SIR issued for process
```

RIRCM JSYS 143

```
FUNCTION
Returns the mask for reserved software interrupt channels
for the specified process.

CALLING SEQUENCE
AC1:  Process handle

RETURNS    +1:  Always, with channel mask in AC2
```

RLJFN JSYS 23

```
FUNCTION
Releases the specified closed JFNs belonging to the current
process or its inferiors.

CALLING SEQUENCE
AC1:  JFN; -1 for all JFNs

RETURNS    +1:  Failure, error code in AC1
           +2:  Success
```

RMAP JSYS 61

```
FUNCTION
Acquires a handle on a page in a process to determine the
access allowed for that page.

CALLING SEQUENCE
AC1:  <process handle>,,<page # within process>

RETURNS    +1:  Always, with
                AC1:  <process/file designator>,,<page #>
                      -1 if page does not exist
                AC2:  Access bits; 0 if page does not exist
                      B2(RM%RD)  Read access allowed
                      B3(RM%WR)  Write access allowed
                      B4(RM%EX)  Execute access allowed
                      B5(RM%PEX) Page exists
                      B9(RM%CPY) Copy-on-write access allowed
```

RNAMF JSYS 35

FUNCTION
Renames an existing file.

CALLING SEQUENCE
AC1: Source file JFN
AC2: Destination file JFN

RETURNS +1: Failure, error code in AC1
+2: Success, JFN in AC1 is released, and JFN in AC2 is associated with file under its new filespec

ROUT JSYS 55

FUNCTION
Outputs a byte nonsequentially (randomly) to the specified file.

RESTRICTIONS
For disk files only.

CALLING SEQUENCE
AC1: JFN
AC2: Byte to be output, right-justified
AC3: Destination byte number within file

RETURNS +1: Always

RPACS JSYS 57

FUNCTION
Returns the accessibility of a page.

CALLING SEQUENCE
AC1: <process/file designator>,,<process/file page number>

RETURNS +1: Always, with
AC2: Flags

B2(PA%RD)	Read access allowed
B3(PA%WT)	Write access allowed
B4(PA%EX)	Execute access allowed
B5(PA%PEX)	Page exists
B6(PA%IND)	Indirect pointer
B9(PA%CPY)	Copy-on-write
B10(PA%PRV)	Private page
B20(P1%RD)	Read access allowed in 1st pointer
B21(P1%WT)	Write access allowed in 1st pointer
B22(P1%EX)	Execute access allowed in

	1st pointer
B23(P1%PEX)	Page exists in 1st pointer
B27(P1%CPY)	Copy-on-write in 1st pointer

RPCAP JSYS 150

FUNCTION
Returns the capabilities for the specified process.

CALLING SEQUENCE
AC1: Process handle

RETURNS +1: Always, with
AC2: Capabilities possible for this process
AC3: Capabilities enabled for this process

RSCAN JSYS 500

FUNCTION
Places a text string in, or reads a text string from, the job's rescan buffer (an area of storage in the Job Storage Block).

CALLING SEQUENCE
AC1: Byte pointer to new text string (1st call, to store string)
0,,<function code> (2nd call, to read string)

RETURNS +1: Failure, error code in AC1
+2: Success, with
AC1: Updated pointer if one supplied, or Count of characters in rescan buffer, or 0 if rescan buffer empty

FUNCTION CODES

Code	Symbol	Meaning
0	.RSINI	Make rescan buffer available for input
1	.RSCNT	Return count of characters remaining in rescan buffer

RSMAP% JSYS 610

FUNCTION
Reads a section map, and provides information about the mapping of one section of a fork's memory.

CALLING SEQUENCE
AC1: <fork handle>,,<section number>

```
RETURNS    +1:  Always, with
                AC1:  -1 if no current mapping;
                       0 if mapping in private section;
                      <fork handle>,,<section #> if indirect
                      or shared mapping to another fork's
                      section; or JFN,,<section #> if
                      shared mapping to file section
                AC2:  Access bits
                      B2(SM%RD) Read access allowed
                      B3(SM%WR) Write access allowed
                      B4(SM%EX) Execute access allowed
                      B5(PA%PEX) Section exists
                      B6(SM%IND) Section created using
                          indirect pointer
```

RTFRK JSYS 322

FUNCTION
Returns the handle of a process that was suspended because of a monitor call intercept and the monitor call that the process was attempting to execute.

```
RETURNS    +1:  Always, with
                AC1:  Handle of process that generated
                      interrupt
                AC2:  JSYS instruction that caused process
                      suspension
```

RTIW JSYS 173

FUNCTION
Reads the terminal interrupt word for the specified process or the entire job, and returns the terminal interrupt word mask.

```
AC1:  B0(RT%DIM)       Return mask for deferred terminal
                       interrupts
      B18-35(RT%PRH)   Process handle, or -5 for entire job

RETURNS    +1:  Always, with
                AC2:  Terminal interrupt mask
                AC3:  Deferred terminal interrupt mask
```

RUNTM JSYS 15

FUNCTION
Returns the run time of the specified process or of the entire job.

CALLING SEQUENCE
AC1: Process handle

RETURNS +1: Always, with
AC1: Runtime (in mss) right-justified
AC2: Divisor to convert mss to sec (1000)
AC3: Console time (in mss)

RWM JSYS 135

FUNCTION
Returns the word mask for the interrupts waiting on software channels for the specified process.

CALLING SEQUENCE
AC1: Process handle

RETURNS +1: Always, with
AC1: 36-bit word (1B*n* indicates pending interrupt on channel *n*)
AC2: Status of interrupts in progress (1B*n* in LH indicates priority level *n* interrupt occurred in user code; 1B(18+*n*) in RH indicates priority level *n* interrupt occurred in monitor code)

RWSET JSYS 176

FUNCTION
Releases the working set by removing all of the current process's pages from its working set.

RETURNS +1: Always

SACTF JSYS 62

FUNCTION
Sets the account to which the specified file is to be charged.

RESTRICTIONS
In non-zero sections, OWGBPs must specify 7-bit bytes.

CALLING SEQUENCE
AC1: JFN
AC2: <5B2!<account number>B35>; or byte pointer to account string (maximum 39 characters)

RETURNS +1: Failure, error code in AC1
+2: Success, updated byte pointer in AC2

SAVE JSYS 202

FUNCTION
Saves, in nonsharable format, pages of a process in the specified file.

RESTRICTIONS
Legal for single-section processes only.

CALLING SEQUENCE
AC1: <process handle>,,JFN
AC2: table entry; or 0,,<table pointer>

RETURNS +1: Always

TABLE FORMAT

Word	Contents
0 to n	<length of save area>,,<address of 1st word to save>
n+1	0

SCTTY JSYS 324

FUNCTION
Redefines the controlling terminal for the specified process and all of its inferiors.

RESTRICTIONS
Requires SC%SCT capability enabled in the process capability word for some functions; cannot be used to change the job's controlling terminal or the controlling terminal of the current process or its superiors.

CALLING SEQUENCE
AC1: <function code>,,<process handle>
AC2: Terminal designator

RETURNS +1: Always

FUNCTION CODES

Code	Symbol	Meaning
0	.SCRET	Return designator of given process's controlling terminal in AC2
1	.SCSET	Change given process's (and inferiors) controlling terminal to terminal designated

		in AC2 (SC%SCT)
2	.SCRST	Reset given process's (and inferiors) controlling terminal to job's controlling terminal (SC%SCT)

SCVEC JSYS 301

FUNCTION
Sets the entry vector and the UUO locations for the compatibility package.

CALLING SEQUENCE
AC1: Process handle
AC2: <entry vector length>,<entry vector address>;
0 to merge compatibilty package into caller's address space; or
-1 to disable UUO simulation
AC3: <UUO location>,,<PC location>

RETURNS +1: Always

COMPATIBILITY PACKAGE'S ENTRY VECTOR

Word	Symbol	Meaning
0	.SVEAD	Entry address for interpreting UUOs
1	.SVINE	Initial entry for setup and first UUO
2	.SVGET	Entry for GET share file routine (obsolete)
3	.SV40	Address to receive contents of location 40 on UUO call
4	.SVRPC	Address to receive return PC word on UUO call
5	.SVMAK	Entry for MAKE share file routine (obsolete)
6-7	.SVCST	2 word block for handling CTRL/C, START sequences between compatibility package and TOPS-20 Command Processor

SDSTS JSYS 146

FUNCTION
Sets the status of a device.

RESTRICTIONS
No-op for devices that do not have device-dependent status bits.

CALLING SEQUENCE
AC1: JFN
AC2: New status bits

RETURNS +1: Always

SDVEC JSYS 543

FUNCTION
Sets the entry vector for the Record Management System (RMS).

RESTRICTIONS
Requires RMS software (currently available only with BASIC and COBOL)

CALLING SEQUENCE
AC1: process handle
AC2: <entry vector length>,,<entry vector address>

RETURNS +1: Always

RECORD MANAGEMENT SYSTEM'S ENTRY VECTOR

Word	Symbol	Meaning
0	.SDEAD	Entry address for RMS calls
1	.SDINE	Initial entry for first RMS call
2	.SDVER	Pointer to RMS version block
3	.SDDMS	Address in which to store RMS call
4	.SDRPC	Address in which to store return PC word

SETER JSYS 336

FUNCTION
Sets the most recent error condition encountered by a process, stores it in the Process Storage Block.

CALLING SEQUENCE
AC1: Process handle
AC2: Error code to set

RETURNS +1: Always

SETJB JSYS 541

FUNCTION
Sets job parameters for the specified job.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability to set parameters for other than current job.

CALLING SEQUENCE
AC1: Jobno, or -1 for current job
AC2: Function code
AC3: Function value

RETURNS +1: Always

```
FUNCTION CODES
Code   Symbol    Function/Values
 0     .SJDEN    Set default magtape density
 1     .SJPAR    Set default for magtape parity
                 0  .SJPRO  Odd parity
                 1  .SJPRE  Even parity
 2     .SJDM     Set default for magtape data mode
 3     .SJRS     Set default for magtape record size (in
                 bytes)
 4     .SJDFS    Set spooling mode
                 0  .SJSPI  Immediate mode spooling
                 1  .SJSPD  Deferred mode spooling
 5     .SJSRM    Set remark for current job session; pointer
                 to remark in AC3
 6     .SJT2O    Indicate if job is at EXEC or program level
                 -1 Job is at EXEC level
                  0 Job is at program level
 7     .SJDFR    Set job default retrieval
                 0  .SJRFA  OPENF of off-line disk file should
                            fail (default)
                 1  .SJRWA  OPENF of off-line disk file should
                            wait for restoral
10     .SJBAT    Set batch flags and batch stream number
                 B0-1(OB%WTO)    Write to operator capability
                                 0  .OBALL  WTO & WTOR allowed
                                 1  .OBNWR  No WTR allowed
                                 2  .OBNOM  No message allowed
                 B10(OB%BSS)     OB%BSN contains batch stream #
                 B11-17(OB%BSN)  Batch stream #
11     .SJLLO    Set job logical location (node name)
```

SETNM JSYS 210

FUNCTION
Sets the private name of the program being used by the current job.

CALLING SEQUENCE
AC1: SIXBIT name used to identify program

RETURNS +1: Always

SETSN JSYS 506

FUNCTION
Sets either the system name or the private name of the program being used by the current job.

CALLING SEQUENCE
AC1: SIXBIT name to be used as system name
AC2: SIXBIT name to be used as private name

RETURNS +1: Failure
+2: Success

SEVEC JSYS 204

FUNCTION
Sets the entry vector of the specified process.

RESTRICTIONS
The process must run in only one section of memory.

CALLING SEQUENCE
AC1: Process handle
AC2: <entry vector length>,,<entry vector address>; or 0 to remove entry vector

RETURNS +1: Always

SFACS JSYS 160

FUNCTION
Sets the ACs of the specified process.

CALLING SEQUENCE
AC1: Process handle
AC2: Address of 20 word block containing new AC values for process

RETURNS +1: Always

SFBSZ JSYS 46

FUNCTION
Resets the byte size for a specific opening of a file.

CALLING SEQUENCE
AC1: JFN
AC2: Byte size, right-justified

RETURNS +1: Failure, error code in AC1
+2: success

SFCOC JSYS 113

FUNCTION
Sets the control character output control (CCOC) for the specified terminal.

CALLING SEQUENCE
AC1: TTY designator
AC2: CCOC word
AC3: CCOC word

RETURNS +1: Always

SFMOD JSYS 110

FUNCTION
Sets the program-related modes (in the JFN mode word) for the specified terminal.

CALLING SEQUENCE
AC1: TTY designator
AC2: JFN mode word

RETURNS +1: Always

SFORK JSYS 157

FUNCTION
Starts the specified process; if the process is frozen, SFORK changes the PC but does not resume the process. On extended machines, the PC section number is obtained from the process entry vector.

RESTRICTIONS
Requires TOPS-20 Version 5 or later for extended addressing.

CALLING SEQUENCE
AC1: <flags>,,<process handle>
1B0(SF%CON) Ignore address in AC2 and start process where halted
AC2: <flags>,,<process starting address> (PC of process being started)

RETURNS +1: Always

SFPOS JSYS 526

FUNCTION
Sets the position of the specified terminal's pointer.

CALLING SEQUENCE
AC1: TTY designator
AC2: <line number>,,<column number>

RETURNS +1: Always

SFPTR JSYS 27

FUNCTION
Sets the position of the specified file's pointer for subsequent I/O to the file.

CALLING SEQUENCE
AC1: JFN
AC2: Byte number to which pointer is to be set; -1 for current EOF

RETURNS +1: Failure, error code in AC1
+2: Success

SFRKV JSYS 201

FUNCTION
Starts the specified process using the position given in its entry vector.

CALLING SEQUENCE
AC1: Process handle
AC2: Offset in entry vector of start address to use

RETURNS +1: Always

SFTAD JSYS 534

FUNCTION
Sets the dates and times associated with the specified file.

RESTRICTIONS
Some functions require enabled WHEEL or OPERATOR capability.

CALLING SEQUENCE
AC1: Source designator
AC2: Address of argblk
AC3: Length of argblk

RETURNS +1: Always

ARGUMENT BLOCK

Word	Symbol	Meaning
0	.RSWRT	Internal date/time file was last written (enabled WHL/OPR for >current)
1	.RSCRV	Internal date and time file was created (enabled WHL/OPR for >current)
2	.RSREF	Internal date and time file was last read (enabled WHL/OPR for >current)
3	.RSCRE	System date and time of last write by monitor (enabled WHL/OPR)
4	.RSTDT	Tape-write date and time of archived or migrated files (enabled WHL/OPR)
5	.RSNET	On-line expiration date and time (date/time or interval)
6	.RSFET	Offline expiration date and time (date/time or interval)

SFUST JSYS 551

FUNCTION
Sets the name of either the author of the file or the user who last wrote to the file.

RESTRICTIONS
Some functions require enabled WHEEL or OPERATOR capability, or caller must have write or owner access to specified file.

CALLING SEQUENCE
AC1: <function code>,,JFN
AC2: Byte pointer to ASCIZ author/user name string

RETURNS +1: Always, with updated byte pointer in AC2

FUNCTION CODES

Code	Symbol	Meaning
0	.SFAUT	Set name of author of file
1	.SFLWR	Set name of user who last wrote file (enabled WHL/OPR)

SIBE JSYS 102

FUNCTION
Tests to see if the designated file input buffer is empty.

CALLING SEQUENCE
AC1: Source designator

RETURNS +1: if device is active terminal and input buffer not empty; or if device is not terminal, is open for read, and input buffer not empty

AC2: Byte count remaining in input buffer
+2: if device is non-active terminal
AC2: Error code
if device is active terminal and input buffer empty; if device not terminal and not open for read; or if device not terminal, is open for read, and input buffer empty
AC2: 0

SIN JSYS 52

FUNCTION
Reads a string from the specified source.

CALLING SEQUENCE
AC1: Source designator
AC2: Byte pointer address to store string

AC3: 0 to read string that terminates with null byte
+n to read string of n characters, or terminate on byte that matches contents of AC4
-n to read string of n bytes
AC4: Byte (R-J) on which to terminate input (if +n in AC3)

RETURNS +1: Always, with
AC1: Updated byte pointer
AC2: Updated byte pointer
AC3: Updated count of bytes transferred

SINR JSYS 531

FUNCTION
Reads a record from the specified device; the calling program must specify the record size (SET TAPE RECORD-LENGTH of .MOSRS function of MTOPR); default record size is 1000 bytes.

RESTRICTIONS
Will not read across record boundaries.

CALLING SEQUENCE
AC1: Source (device) designator
AC2: Byte pointer to address to store record
AC3: 0 to read record that terminates with null byte
+n to read record of n characters, or terminate on byte that matches contents of AC4
-n to read record of n bytes
AC4: Byte (R-J) on which to terminate input (if +n in AC3)

RETURNS +1: Always, with
AC1: Updated byte pointer
AC2: Updated byte pointer

AC3: 0 if specified record size = actual record size (all bytes read)
# of bytes read if specified record size > actual record size
# of bytes requested if specified record size < actual record size; IOX10 returned and unread bytes discarded

SIR JSYS 125

FUNCTION
Sets the addresses of the channel and priority level tables for the specified process.

RESTRICTIONS
The process must run in section 0 of memory, with channel and priority level tables in that section. (Use XSIR% to set table addresses for multiple-section processes.)

CALLING SEQUENCE
AC1: Process handle
AC2: LEVTAB,,CHNTAB

RETURNS +1: Always

SIRCM JSYS 142

FUNCTION
Sets the mask for reserved software interrupt channels for the specified inferior process, causing conditions occurring on software channels that have the corresponding mask bit set to terminate or freeze the process, rather than generate an interrupt.

CALLING SEQUENCE
AC1: Inferior process handle
AC2: Channel mask with bits set for reserved channels
AC3: Deferred terminal interrupt word

RETURNS +1: Always

SIZEF JSYS 36

FUNCTION
Returns the length of an existing file.

CALLING SEQUENCE
AC1: JFN
RETURNS +1: Failure, error code in AC1

+2: Success, with
AC2: File byte count (byte size from FDB)
AC3: File page count

SJPRI JSYS 245

FUNCTION
Sets the scheduler priority control word.

RESTRICTIONS
This JSYS is reserved for DIGITAL. Requires enabled WHEEL or OPERATOR capability.

CALLING SEQUENCE
AC1: Job #
AC2: Priority word

RETURNS +1: Always

PRIORITY WORD

Bits	Contents
B0-17	Percentage of CPU resources (1 - 99%) guaranteed to job; 0 for no request
B18	System job flag (JP%SYS); higher priority than user jobs with guaranteed runtime
B24-29	Highest priority queue job may run in; 0 for no queue assignment request
B30-35	Lowest priority queue job may run in, specified as desired queue+1; 0 for no queue assignment request

SKED% JSYS 577

FUNCTION
Reads or modifies the monitor's scheduler data base.

RESTRICTIONS
Some functions require enabled WHEEL or OPERATOR capability.

CALLING SEQUENCE
AC1: Function code
AC2: Address of argblk

RETURNS +1: Always

FUNCTION CODES

Code	Symbol	Function
1	.SKRBC	Read bias control knob setting
		0 .SACNT Word count including this word
		1 .SAKNB Bias control knob setting (RET)
2	.SKSBC	Set bias control setting (WHL/OPR)
		0 .SACNT Word count including this word
		1 .SAKNB Bias control setting (1-20)

```
3     .SKRCS   Read class parameters
                0 .SACNT   Word count including this word
                1 .SACLS   Class of job (RET)
                2 .SASHR   Share of CPU allocated to class
                           (RET; 0.0<=n.n<=1.0)
                3 .SAUSE   Amount of CPU used by class
                           (RET; 0.0<=n.n<=1.0)
                4 .SA1ML   1 min load avg for class (RET)
                5 .SA5ML   5 min load avg for class (RET)
                6 .SA15L   15 min load avg for class (RET)
4     .SKSCS   Set class parameters (WHL/OPR)
                0 .SACNT   Word count including this word
                1 .SACLS   Class of job
                2 .SASHR   Share of CPU allocated to class
                           (0.0<=n.n<=1.0)
5     .SKICS   Start or stop class scheduler (WHL/OPR)
                0 .SACNT   Word count including this word
                1 .SACTL   Control flags
                           B0(SK%ACT)  Class by accounts
                           B1(SK%WDF)  Withhold windfall
                           B2(SK%STP)  Class scheduler off
6     .SKSCJ   Set job class (WHL/OPR for other than calling
               job)
                0 .SACNT   Word count including this word
                1 .SAJOB   Job #; -1 for calling job
                2 .SAJCL   Class of job
                3 .SAWA    Windfall allocation
7     .SKRJP   Read class parameters for a job
                0 .SACNT   Word count including this word
                1 .SAJSH   Job's share of CPU (RET;
                           0.0<=n.n<=1.0)
                2 .SAJUS   Job's current CPU use (RET;
                           0.0<=n.n<=1.0)
10    .SKBCR   Read class setting for batch jobs
                0 .SACNT   Word count including this word
                1 .SABCL   Batch class; -1 if none (RET)
11    .SKBCS   Set batch class (WHL/OPR)
                0 .SACNT   Word count including this word
                1 .SABCL   Batch class; -1 for none
12    .SKBBG   Run all batch jobs in "dregs" queue; illegal
               if class scheduling in use (WHL/OPR)
                0 .SACNT   Word count including this word
                1 .SADRG   0    don't run in dregs queue
                           ≠0   run in dregs queue
13    .SKDDC   Reserved
14    .SKRCV   Read status
                0 .SACNT   Word count including this word
                1 .SACTL   Flags
                           B0(SK%ACT)  Class by accounts
                           B1(SK%WDF)  Withhold windfall
                           B2(SK%STP)  Class scheduler off
                           B3(SK%DRG)  Batch jobs being run
                                       in dregs queue
```

SKPIR JSYS 127

FUNCTION
Tests to see if the software interrupt system is enabled for the specified process, and performs a skip return if PSI enabled.

CALLING SEQUENCE
AC1: Process handle

RETURNS +1: Software interrupt system is off
+2: Software interrupt system is on

SMAP% JSYS 767

FUNCTION
Maps one or more contiguous sections of memory. Maps one or more complete sections from a file to a process (Case I) or from a process to another process (Case II), creates new sections (Case III), or deletes sections from a process (Case IV).

CALLING SEQUENCE
AC1: JFN,,<file section number> (Case I)
<fork handle>,,<section number> (Case II)
0 (Case III)
-1 (Case IV)
AC2: <fork handle>,,<process section number> (Cases I-IV)
AC3: 0,,<# (1-37) of contiguous sections to map> (Case IV)
flags,,<# (1-37) of contiguous sections to map> (Cases I-III)
B2(SM%RD) Allow read access (Cases I-III)
B3(SM%WR) Allow write access (Cases I-III)
B4(SM%EX) Allow execute access (Cases I-III)
B6(SM%IND) Map using indirect section pointer (Case II-III)

RETURNS +1: Always

SMON JSYS 6

FUNCTION
Sets various flags and parameters in the monitor's data base.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability; some functions are for ARPANET systems only.

CALLING SEQUENCE
AC1: Function code
AC2: New value for function

RETURNS +1: Always

FUNCTION CODES

Code	Symbol	Function
0	.SFFAC	Allow FACT file entries AC2: 1(SF%FAC) to set; 0 to clear
1	.SFCDE	CHECKD found errors AC2: 1(SF%CDE) to set; 0 to clear
2	.SFCDR	CHECKD is running AC2: 1(SF%CDR) to set; 0 to clear
3	.SFMST	Manual start in progress AC2: 1(SF%MST) to set; 0 to clear
4	.SFRMT	Allow remote LOGINs AC2: 1(SF%RMT) to set; 0 to clear
5	.SFPTY	Allow PTY LOGINs AC2: 1(SF%PTY) to set; 0 to clear
6	.SFCTY	Allow CTY LOGINs AC2: 1(SF%CTY) to set; 0 to clear
7	.SFOPR	Operator in attendance AC2: 1(SF%OPR) to set; 0 to clear
10	.SFLCL	Allow local LOGINs AC2: 1(SF%LCL) to set; 0 to clear
11	.SFBTE	Bit table errors found on startup AC2: 1(SF%BTE) to set; 0 to clear
12	.SFCRD	Users can change directory parameters AC2: 1(SF%CRD) to set; 0 to clear
13	.SFNVT	Allow ARPANET terminal LOGINs (ARPA) AC2: 1(SF%NVT) to set; 0 to clear
21	.SFUSG	Allow USAGE file entries AC2: 1(SF%USG) to set; 0 to clear
22	.SFFLO	Set full disk latency optimization (requires KL10-E revision level 2 and RH20 board M8555 revision level D) AC2: 1(SF%FLO) to set; 0 to clear
23	.SFMTA	Enable MOUNTR magtape allocation AC2: 1(SF%MTA) to set; 0 to clear
24	.SFMSO	Set system message level 0 AC2: 1(SF%MSO) to set; 0 to clear
25	.SFMS1	Set system message level 1 AC2: 1(SF%MS1) to set; 0 to clear
44	.SFNTN	Turn ARPANET on (ARPA) AC2: 1 to set; 0 to clear
45	.SFNDU	Reinitialize ARPANET if down (ARPA) AC2: 1 to set; 0 to clear
46	.SFNHI	Initialize ARPANET host table (ARPA) AC2: 1 to set; 0 to clear
47	.SFTMZ	Set local time zone AC2: time zone
50	.SFLHN	Set local ARPANET host number (ARPA) AC2: ARPANET host number
51	.SFAVR	Enable account validation AC2: 1 to set; 0 to clear
52	.SFSTS	Enable status reporting AC2: 1 to set; 0 to clear
53	.SFSOK	Set GETOK% defaults AC2: flags,,<GETOK% function code>

```
                        BO(SF%EOK)  0 to disable access
                                      checking
                                    1 to enable access checking
                        B1(SF%DOK)  0 to deny access if
                                      checking disabled
                                    1 to allow access if
                                      checking disabled
54      .SFMCY    Set maximum offline expiration period
                  AC2:  expiration period in days
55      .SFRDU    Update last access read time for directories
                  AC2:  1 to set; 0 to clear
56      .SFACY    Set maximum offline expiration period for
                  archive files
                  AC2:  expiration period in days
57      .SFRTW    Set no-retrieval-waits flag
                  AC2:  1 to set; 0 to clear
60      .SFTDF    Set tape mount controls
                  AC2:  BO(MT%UUT)  1 to unload unrecognizable
                                      tapes
                                    0 to treat unrecognizable
                                      tapes as unlabeled
61      .SFWSP    Enable working set preloading
                  AC2:  1 to set; 0 to clear
```

SNDIM JSYS 750

FUNCTION
Places a message in a previously assigned ARPANET special message queue.

RESTRICTIONS
For ARPANET systems only.

CALLING SEQUENCE

```
AC1:  BO        If on, message contains 96-bit leader; if
                off, message contains 32-bit leader
      B1        If on, data in high-order 32 bits of each
                word of message; if off, data in all 36 bits
                of each word of message
      B18-35    Special queue header
AC2:  Address of extended message

RETURNS   +1:  Failure, error code in AC1
          +2:  Success, message queued
```

See BBN Report #1822 for the format of the extended message.

SNOOP JSYS 516

FUNCTION
Performs system performance analysis.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability.

CALLING SEQUENCE
AC1: Function code
AC2: Function-specific argument
AC3: Function-specific argument
AC4: Function-specific argument

RETURNS +1: Failure, error code in AC1
+2: Success

FUNCTION CODES

Code	Symbol	Function/Arguments
0	.SNPLC	Declare and lock code into monitor's address space AC2: number of pages desired AC3: user page number of start of breakpoint routines to be locked On return AC2: monitor page # corresponding to user page #
1	.SNPLS	Lock swappable monitor
2	.SNPDB	Define a breakpoint AC2: Number of breakpoint AC3: Address in monitor space to be patched AC4: Instruction to be executed before patched instruction
3	.SNPIB	Insert all breakpoints and start analyzing
4	.SNPRB	Remove all breakpoints and stop analyzing
5	.SNPUL	Unlock and release all storage; remove all breakpoints
6	.SNPSY	Obtain address of monitor symbol AC2: Radix-50 symbol AC3: Radix-50 program name if local address desired; 0 to search entire symbol table On return AC2: Monitor address or value of symbol
7	.SNPAD	Obtain monitor symbol AC2: 36-bit value of symbol to be looked up in monitor's symbol table AC3: Radix-50 program name if local value desired; 0 to search entire symbol table On return AC2: Radix-50 symbol closest to and less than given value AC3: Difference between returned value and given value

SOBE JSYS 103

FUNCTION
Tests to see if the designated file output buffer is empty.

CALLING SEQUENCE
AC1: Destination designator

RETURNS +1: Output buffer is not empty
AC2: number of bytes remaining in output buffer
+2: Output buffer is empty
AC2: 0
Error return
AC2: Error code

SOBF JSYS 175

FUNCTION
Tests to see if the designated file's output buffer is full.

CALLING SEQUENCE
AC1: File designator

RETURNS +1: Output buffer is not full
AC2: Count of bytes in buffer
Error return
AC2: 0
+2: Output buffer is full or error
AC2: Count of bytes in buffer if no error

SOUT JSYS 53

FUNCTION
Writes a string to the specified destination.

CALLING SEQUENCE
AC1: Destination designator
AC2: Byte pointer to string to be written
AC3: 0 to write string that terminates with null byte
+n to write string of n characters, or terminate on byte that matches contents of AC4
-n to write string of n bytes
AC4: Byte (R-J) on which to terminate output (if +n in AC3)

RETURNS +1: Always, with
AC1: Updated byte pointer
AC2: Updated byte pointer
AC3: Updated count of bytes transferred

SOUTR JSYS 532

FUNCTION
Writes a variable-length record to the specified device; the calling program must specify the record size (SET TAPE RECORD-LENGTH of .MOSRS function of MTOPR); default record size is 1000 bytes.

CALLING SEQUENCE
AC1: Destination designator
AC2: Byte pointer to string to be written
AC3: 0 to write record that terminates with null byte
+n to write record of n characters, or terminate on byte that matches contents of AC4
-n to write record of n bytes
AC4: Byte (R-J) on which to terminate input (if +n in AC3)

RETURNS +1: Always, with
AC2: Last non-zero byte written
AC3: (# bytes written) - (# bytes requested)

SPACS JSYS 60

FUNCTION
Sets the accessibility of a page.

CALLING SEQUENCE
AC1: <process/file designator>,,<process/file page number>
AC2: Access flags
B2(PA%RD) Permit read access
B3(PA%WT) Permit write access
B4(PA%EX) Permit execute access
B9(PA%CPY) Permit copy-on-write

SPJFN JSYS 207

FUNCTION
Sets the primary JFNs (.PRIIN and .PRIOU) for the specified process.

CALLING SEQUENCE
AC1: Process handle
AC2: <primary input JFN>,,<primary output JFN>; or -1 in appropriate half to set to process's controlling terminal

RETURNS +1: Always

SPLFK JSYS 314

FUNCTION
Splices a process structure.

RESTRICTIONS
The new superior must be either the calling process or an inferior of it; the new inferior process must be an inferior of the calling process. The new superior and new inferior processes must not be the same process.

CALLING SEQUENCE
AC1: Process handle of new superior process
AC2: Process handle of new inferior process

RETURNS +1: Failure, error code in AC1
+2: Success, with
AC1: process handle of new superior
AC2: process handle of new inferior

SPOOL JSYS 517

FUNCTION
Defines and initializes a device to be used for input spooling, or sets and reads the directory for a spooled device.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability.

CALLING SEQUENCE
AC1: <length of argblk>,,<function code>
AC2: Address of argblk

RETURNS +1: Failure, error code in AC1
+2: Success

FUNCTION CODES

Code	Symbol	Function/Arguments
0	.SPLDI	Define an input spooling device
		0 .SPLDV Device designator of input device
		1 .SPLNA Pointer to input file string
		2 .SPLGN Generation number of first file
1	.SPLSD	Set directory of spooled device (enabled WHL/OPR)
		0 .SPLDV Device designator of spooled device
		1 .SPLDR Directory number of user who opened spooled device
2	.SPLRD	Read directory of spooled device
		0 .SPLDV Designator of spooled device

SPRIW JSYS 243

FUNCTION
Sets the priority word for the specified process.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability.

CALLING SEQUENCE
AC1: Process handle
AC2: Priority word

RETURNS +1: Always

PRIORITY WORD

Bits	Contents
B0-17	Percentage of CPU resources (1 - 99%) guaranteed to job; 0 for no request
B18	System job flag (JP%SYS); higher priority than user jobs with guaranteed runtime
B24-29	Highest priority queue job may run in; 0 for no queue assignment request
B30-35	Lowest priority queue job may run in, specified as desired queue+1; 0 for no queue assignment request

SSAVE JSYS 203

FUNCTION
Creates a sharable, save-format file for the given JFN by copying (not sharing) pages from the given process.

CALLING SEQUENCE
AC1: <process handle>,,JFN
AC2: One table entry; or 0,,<table address>
AC3: 2nd word of 2-word table entry (if SS%EPN set); or 0

RETURNS +1: Always

TABLE ENTRY

Word	Contents	
0	Flags:	
	B0-17(SS%NNP)	-(# of pages) in each group
	B18(SS%CPY)	Allow copy-on-write access
	B19(SS%UCA)	Limit access according to user's current page access ANDed with table word access
	B20(SS%RD)	Allow read access
	B21(SS%WR)	Allow write access
	B22(SS%EXE)	Allow execute access
	B23(SS%EPN)	Table entry is 2 words long; 2nd word contains page # of 1st page of group
	B27-35(SS%FPN)	If 0B23(SS%EPN), page # of 1st page in group; if 1B23(SS%EPN), 0
1	Page number of 1st page in group (for pages in	

```
       non-zero section)
 2     0
```

STAD JSYS 226

FUNCTION
Sets the system's date.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability if the system's date is already set.

CALLING SEQUENCE
AC1: day,,<fraction of day>

RETURNS +1: Failure, error code in AC1
         +2: success

STCMP JSYS 540

FUNCTION
Compares two ASCIZ strings.

RESTRICTIONS
Alphabetics are compared in upper case, regardless of case in string.

CALLING SEQUENCE
AC1: Byte pointer to test string
AC2: Byte pointer to base string

RETURNS +1: always, with
              AC1: 0 if strings are equal; or flags
                    B0(SC%LSS) Test string is less than
                               base string
                    B1(SC%SUB) Test string is subset of
                               base string
                    B2(SC%GTR) Test string is greater than
                               base string
              AC2: Base byte pointer, pointing before 1st
                    non-matching byte

STDEV JSYS 120

FUNCTION
Translates the given device name string to its corresponding device designator.

CALLING SEQUENCE
AC1: Byte pointer to device name string

RETURNS +1: Failure, error code in AC2
+2: Success, device designator in AC2

STI JSYS 114

FUNCTION
Simulates terminal input.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability if specified terminal not assigned or opened by calling process, or is not accepting advice.

CALLING SEQUENCE
AC1: TTY designator
AC2: Character to be input, right-justified

RETURNS +1: Always

STIW JSYS 174

FUNCTION
Sets the terminal interrupt word for the entire job or a specific process.

RESTRICTIONS
Requires enabled SC%CTY capability in process capability word.

CALLING SEQUENCE
AC1: BO(ST%DIM) Set deferred terminal interrupt mask given in AC3
B18-35(ST%PRH) process handle
AC2: Terminal interrupt word mask (1B*n* enables terminal code *n*)
AC3: Deferred terminal interrupt word mask (1B*n* defers terminal code *n*)

RETURNS +1: Always

STO JSYS 246

FUNCTION
Simulates terminal output.

CALLING SEQUENCE
AC1: TTY designator

RETURNS +1: Always, with character right-justified in AC2

STPAR JSYS 217

FUNCTION
Sets the device-related modes for the specified terminal.

CALLING SEQUENCE
AC1: TTY designator
AC2: JFN mode word

RETURNS +1: Always

STPPN JSYS 556

FUNCTION
Translates the given directory name string to its corresponding project-programmer number (a TOPS-10 36-bit directory designator).

RESTRICTIONS
In non-zero sections, DWGBPs must specify 7-bit bytes.

CALLING SEQUENCE
AC1: JFN; 36-bit directory number; or byte pointer to ASCIZ directory name string

RETURNS +1: Always, with project-programmer number in AC2

STSTS JSYS 25

FUNCTION
Clears the status of a file.

CALLING SEQUENCE
AC1: 0,,JFN
AC2: Flags

B9(GS%ERR)	File may be in error
B13(GS%HLT)	I/O errors are terminating conditions
B17(GS%FRK)	JFN is restricted

RETURNS +1: Failure, error code in AC1

+2: success

STTYP JSYS 302

FUNCTION
Sets the terminal type number for the specified terminal line.

CALLING SEQUENCE
AC1: TTY designator
AC2: TTY type

RETURNS +1: Always

SWJFN JSYS 47

FUNCTION
Swaps the association of two JFNs by exchanging all information cells of each JFN.

CALLING SEQUENCE
AC1: JFN
AC2: Another JFN

RETURNS +1: Always

SWTRP% JSYS 573

FUNCTION
Provides a process with the ability to intercept arithmetic overflow or underflow conditions.

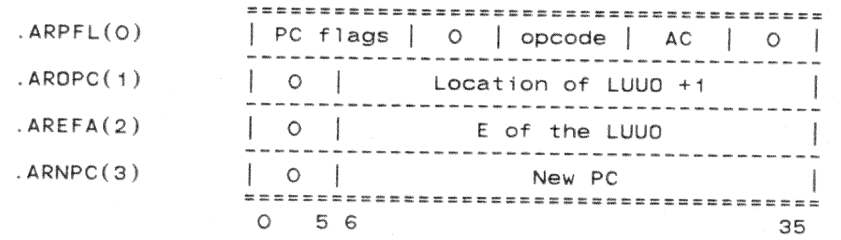
CALLING SEQUENCE
AC1: Process handle
AC2: Function code
AC3: Function-dependent argument

RETURNS +1: Always

FUNCTION CODES

Code	Symbol	Function/Arguments
0	.SWART	Set arithmetic trap location
		AC3: Address of arithmetic trap block; 0 to clear
1	.SWRAT	Read arithmetic trap location
		AC3: Trap block address; 0 if none set (RET)
2	.SWLUT	Set LUUO block address for non-zero sections
		AC3: LUUO address; 0 to clear
3	.SWRLT	Read LUUO block address
		AC3: LUUO address; 0 if none set (RET)

LUUO BLOCK FORMAT



SYERR JSYS 527

FUNCTION
Places information in the system error file.

RESTRICTIONS
Requires enabled WHEEL, OPERATOR, or MAINTENANCE capability.

CALLING SEQUENCE
AC1: Address of argblk
AC2: Length of argblk

RETURNS +1: Always

SYSGT JSYS 16

FUNCTION
Returns the table number, table length, and word 0 of the specified system table.

CALLING SEQUENCE
AC1: SIXBIT table name

RETURNS +1: Always, with
AC1: Word 0 of table
AC2: - <# of words in table>,,<table #>;
0 if table not found

TBADD JSYS 536

FUNCTION
Adds an entry to a standard-formatted command table used for user program command recognition.

CALLING SEQUENCE
AC1: Address of table
AC2: Entry to be added to table

RETURNS +1: Always, with address of new entry in AC1

TBDEL JSYS 535

FUNCTION
Deletes an entry from a standard-formatted command table used for user program command recognition.

CALLING SEQUENCE
AC1: Address of table
AC2: Address of entry to be deleted

RETURNS +1: Always

TBLUK JSYS 537

FUNCTION
Compares the specified string with strings indicated by a command table.

CALLING SEQUENCE
AC1: Address of table
AC2: Byte pointer to string to be compared with string in table

RETURNS +1: Always, with
AC1: Address of entry that matches input string, or address where entry would be if in table
AC2: Recognition flags
B0(TL%NOM) Input string has no match in table
B1(TL%AMB) Input string has more than one match in table
B2(TL%ABR) Input string is valid abbreviation
B3(TL%EXM) Input string has exact match in table
AC3: Pointer to remainder of string in table if 1B2

COMMAND TABLE FORMAT
Word Contents
0 <# of remaining words>,,<max # of remaining words>
1 - n <address of argblk>,,<available to user>

ARGUMENT BLOCK
Word Contents
0 If 0(B0-6) and 1B7(CM%FW), <flags>B18-35 and string begins in next word; if ¬0(B0-6) or 0B7,string starts in this word
B34(CM%NOR) Do not recognize this string

1 Start of string if O(BO-6) and 1B7(CM%FW) in word O

TEXTI JSYS 524

FUNCTION
Reads input from a terminal or a file.

CALLING SEQUENCE
AC1: Address of argblk

RETURNS +1: Failure, error code in AC1
+2: Success, updated pointer in word .RDDBP, appropriate bits set in word .RDFLG, and updated count in word .RDDBC of argblk

ARGUMENT BLOCK

Word	Symbol	Contents	
0	.RDCWB	Word count not including this word	
1	.RDFLG	Flags	
		B0(RD%BRK)	Break on CTRL/Z or ESC
		B1(RD%TOP)	Break on CTRL/G, CTRL/K, CTRL/L, CTRL/Z, ESC, CR, LF
		B2(RD%PUN)	Break on punctuation
		B3(RD%BEL)	Break on EOL (CRLF or LF only)
		B4(RD%CRF)	Suppress CR and return LF only
		B5(RD%RND)	Return if user tries to delete past beginning of buffer
		B6(RD%JFN)	JFNs in word .RDIOJ
		B7(RD%RIE)	Return if input buffer empty
		B8(RD%BBG)	Not used
		B9(RD%BEG)	Return when .RDBKL pointer is reached
		B10(RD%RAI)	Convert lowercase input to UPPERCASE
		B11(RD%SUI)	Suppress CTRL/U indication
		B12(RD%BTM)	Break character terminated input (RET)
		B13(RD%BFE)	Returned because user tried to delete past beginning of buffer (RET)
		B14(RD%BLR)	Backup limit for editing reached (RET)
2	.RDIOJ	Byte pointer to string; or <input JFN>,,<output JFN>	
3	.RDDBP	Byte pointer to destination string buffer	
4	.RDDBC	Number of bytes in destination string	
5	.RDBFP	Byte pointer to beginning of destination buffer	
6	.RDRTY	Byte pointer to beginning of CTRL/R buffer	
7	.RDBRK	Address of 4-word break character mask block	
10	.RDBKL	Byte pointer to backup limit in destination buffer	

TFORK JSYS 321

FUNCTION
Sets and removes monitor call intercepts (JSYS traps) for the given inferior processes.

RESTRICTIONS
Requires enabled WHEEL, OPERATOR, or MAINTENANCE capability for use on execute-only processes.

CALLING SEQUENCE
AC1: <function code>,,<process handle>
AC2: <interrupt channel>,,<size of monitor call bit table> (in bits)
AC3: Address of monitor call bit table

RETURNS +1: Always

FUNCTION CODES

Code	Symbol	Function
0	.TFSET	Set JSYS traps for given process (illegal for execute-only processes)
1	.TFRAL	Remove all JSYS traps for given process (illegal for execute-only processes)
2	.TFRTP	Remove JSYS traps indicated in monitor call bit table for given process (illegal for execute-only processes)
3	.TFSPS	Set interrupts on given software channel
4	.TFRPS	Return interrupt channel in left half of AC2
5	.TFTST	Test if caller is to be intercepted when it attempts to execute monitor calls; On return AC2: -1 intercept; 0 no intercept
6	.TFRES	Remove intercepts for all inferiors and clear assigned software channels
7	.TFUUO	Set JSYS traps for TOPS-10 UUOs for given process (illegal for execute-only processes)
10	.TFSJU	Set JSYS traps for both TOPS-10 UUOs indicated in monitor call bit table (illegal for execute-only processes)
11	.TFRUU	Remove JSYS traps for TOPS-10 UUOs

THIBR JSYS 770

FUNCTION
Blocks the current process for the specified elapsed time or until awakened by a TWAKE monitor call.

RESTRICTIONS
This call is temporary and may not be defined in future releases.

CALLING SEQUENCE
AC1: 0,,<maximum number of seconds to block>

RETURNS +1: Never
+2: Always

TIME JSYS 14

FUNCTION
Returns the amount of time since the system was last restarted.

RETURNS +1: Always, with
AC1: Time in milliseconds, right-justified
AC2: 1000 (divisor for conversion to seconds)

TIMER JSYS 522

FUNCTION
Controls the amount of time either a process within a job or the entire job can run.

CALLING SEQUENCE
AC1: <process handle>,,<function code>
AC2: Time at which to generate interrupt
AC3: Software channel number on which to generate interrupt

RETURNS +1: Failure, error code in AC1
+2: Success

FUNCTION CODES

Code	Symbol	Function/Arguments
0	.TIMRT	Set total runtime of entire job AC2: Total runtime in mss
1	.TIMEL	Set elapsed time for process AC2: Elapsed time in mss before interrupt
2	.TIMDT	Set exact time to generate interrupt for process AC2: Time of interrupt in internal format
3	.TIMDD	Remove any pending interrupts at given time AC2: Time of interrupt in internal format
4	.TIMBF	Remove any pending interrupts before given time AC2: Time of interrupt in internal format
5	.TIMAL	Remove all pending requests for given process

TLINK JSYS 216

FUNCTION
Controls terminal linking.

RESTRICTIONS
Some functions require enabled WHEEL or OPERATOR capability.

CALLING SEQUENCE

AC1:	B0(TL%CRO)	Clear link from remote to object designator
	B1(TL%COR)	Clear link from object to remote designator
	B2(TL%EOR)	Establish link from object to remote designator
	B3(TL%ERO)	Establish link from remote to object designator
	B4(TL%SAB)	Examine B5(TL%ABS) to determine setting of object designator's accept link bit
	B5(TL%ABS)	Set object designator's accept link bit
	B6(TL%STA)	Examine B7(TL%AAD) to determine setting of object designator's accept advice bit
	B7(TL%AAD)	Set object designator's accept advice bit
	B18-35(TL%OBJ)	Object designator
AC2:	0,,<remote designator>	

RETURNS +1: Failure, error code in AC1
+2: Success

TMON JSYS 7

FUNCTION
Returns various flags and parameters in the monitor's data base.

CALLING SEQUENCE
AC1: Function code
AC2: Function-specific arguments

RETURNS +1: Always, with
AC2: Value of function
normally, 1 if set; 0 if clear

FUNCTION CODES

Code	Symbol	Function
0	.SFFAC	FACT file entries are allowed
1	.SFCDE	CHECKD found errors
2	.SFCDR	CHECKD is running
3	.SFMST	Manual start is in progress
4	.SFRMT	Remote LOGINs are allowed
5	.SFPTY	PTY LOGINs are allowed
6	.SFCTY	CTY LOGINs are allowed

7	.SFOPR	Operator is in attendance
10	.SFLCL	Local LOGINs are allowed
11	.SFBTE	Bit table errors found on startup
12	.SFCRD	Users can change nonprivileged directory parameters
13	.SFNVT	ARPANET terminal LOGINs are allowed
21	.SFUSG	USAGE file entries are allowed
22	.SFFLO	Disk latency optimization using RH20 backup register is enabled
23	.SFMTA	MOUNTR magtape allocation is enabled
24	.SFMSO	System message level 0 is enabled
25	.SFMS1	System message level 1 is enabled
44	.SFNTN	ARPANET is on
45	.SFNDU	ARPANET will be reinitialized if it is down
46	.SFNHI	ARPANET host table will be initialized
47	.SFTMZ	Local time zone
50	.SFLHN	ARPANET local host number
51	.SFAVR	Account validation is running
52	.SFSTS	Status reporting is enabled
53	.SFSOK	GETOK% defaults AC2: flags,,<GETOK% function code> BO(SF%EOK) 0 Access checking is disabled 1 Access checking is enabled B1(SF%DOK) 0 Access is denied if checking disabled 1 Access is allowed if checking disabled
54	.SFMCY	Maximum offline expiration period in days for ordinary files
55	.SFRDU	Update last access read time for directories
56	.SFACY	Maximum offline expiration period in days for archive files
57	.SFRTW	File-retrieval requests should fail
60	.SFTDF	Tape mount controls BO(MT%UUT) 1 unload unrecognizable tapes 0 treat unrecognizable tapes as unlabeled
61	.SFWSP	Enable working set preloading

TTMSG JSYS 775

FUNCTION
Sends a message to a specified terminal or to all terminals.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability to send to all terminals. Messages sent by privileged users may have a maximum of 581. characters; messages sent by non-privileged users may have a maximum of 526. characters. This call is temporary and may not be defined in future releases.

CALLING SEQUENCE
AC1: 400000 + TTY number; or -1 for all terminals
AC2: Byte pointer to message string

RETURNS +1: Always

TWAKE JSYS 771

FUNCTION
Wakes the specified job that is blocked because of the execution of a THIBR call.

RESTRICTIONS
This call is temporary and may not be defined in future releases.

CALLING SEQUENCE
AC1: 0,,<number of job to be awakened>

RETURNS +1: Failure, error code in AC1
+2: Success

UFPGS JSYS 525

FUNCTION
Updates pages of the specified file.

CALLING SEQUENCE
AC1: JFN,,<file page # of 1st page to be updated>
AC2: Flags,,<# of sequential pages to update>
BO(UF%NOW) Perform UFPGS without blocking

RETURNS +1: Failure, error code in AC1
+2: Success

USAGE JSYS 564

FUNCTION
Controls accounting on the system by writing entries into the system's data file.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability.

CALLING SEQUENCE
AC1: Function code
AC2: Function argument; or address of record descriptor block

RETURNS +1: Always

FUNCTION CODES

Code	Symbol	Function/Arguments
0	.USENT	Write entry into system's data file AC2: Address of record descriptor block
1	.USCLS	Close system's data file
2	.USCKP	Perform checkpoint of all jobs
3	.USLGI	Initialize checkpoint entry for job AC2: Address of record descriptor block
4	.USLGO	Terminate checkpoint entry for job and write entry to system's data file AC2: Address of record descriptor block
5	.USSEN	Terminate current session, write entry to system's data file, and initialize new checkpoint entry for job AC2: Address of record descriptor block
6	.USCKI	Set checkpoint time interval AC2: Time interval in minutes
7	.USENA	Install accounting data base into running monitor from PS:<SYSTEM>ACCOUNTS-TABLE.BIN
10	.USCAS	Change accounting shift
11	.USSAS	Set accounting shifts AC2: Pointer to argblk of format: 0 <# table entries>,,<max # entries> 1-n B0-6(US%DOW) Days-of-week entry in effect (0=Monday) B7-17 Not used, must be zero B18-35 (US%SSM) Time in seconds since midnight for accounting shift change
12	.USRAS	Read accounting shifts AC2: Pointer to argblk (see .USSAS for format)

USRIO USYS 310

FUNCTION
Places the user program into user I/O mode for executing various hardware I/O instructions.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability.

RETURNS +1: Failure, error code in AC1
+2: Success, user IOT flag set

UTEST JSYS 563

FUNCTION
Provides a method for determining if every instruction in a section of monitor code actually gets executed.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability.

CALLING SEQUENCE
AC1: <function code>,,<length of argblk>
AC2: Address of argblk

RETURNS +1: Always

FUNCTION CODES

Code	Symbol	Function
0	.UTSET	Start testing code
1	.UTCLR	Stop testing code and update bit map in argblk

ARGUMENT BLOCK

Word	Symbol	Contents
0	.UTADR	Address of beginning of code section to be tested
1	.UTLEN	Length of code section to be tested
2	.UTMAP	Start of bit map representing instructions to be tested in code section

UTFRK JSYS 323

FUNCTION
Resumes the execution of a process that was suspended because of a monitor call intercept.

CALLING SEQUENCE
AC1: Flags,,<process handle>
BO(UT%TRP) Cause failure return for suspended process

RETURNS +1: Always

VACCT JSYS 566

FUNCTION
Verifies accounts by validating the supplied account for the given user.

CALLING SEQUENCE
AC1: user number; directory number; or -1 for current user
AC2: Byte pointer to account string

RETURNS +1: Always, with updated pointer in AC2

WAIT JSYS 306

FUNCTION
Dismisses the current process indefinitely and does not return.

WFORK JSYS 163

FUNCTION
Causes the current process to wait for an inferior process to terminate.

CALLING SEQUENCE
AC1: Inferior process handle

RETURNS +1: Always, when specified processes terminates

WILD% JSYS 565

FUNCTION
Compares a possibly wild string against a non-wild string to see if the latter matches the wild string.

CALLING SEQUENCE
AC1: Flags,,<function code>
AC2: Wild argument: JFN or byte pointer to string
AC3: Non-wild argument: JFN or byte pointer to string

RETURNS +1: Always

FUNCTION CODES

Code	Symbol	Function
0	.WLSTR	Compare non-wild string against wild string AC1: BO(WL%LCD) Lowercase characters are distinct from uppercase On return AC1: 0 Strings matched BO(WL%NOM) If on, non-wild string did not match wild string B1(WL%ABR) If on, non-wild string is abbreviation of wild string
1	.WLJFN	Compare non-wild filespec against wild filespec On return AC1: 0 Filespecs matched B1(WL%DEV) Device field does not match B2(WL%DIR) Directory field does not match

```
B3(WL%NAM)   Name field does not match
B4(WL%EXT)   File type does not match
B5(WL%GEN)   Generation number does not
             match
```

XGSEV% JSYS 614

FUNCTION
Gets an extended special entry vector that has been set to allow use of TOPS-10 Compatibility and RMS entry vectors in non-zero sections.

CALLING SEQUENCE
AC1: <vector type code>,,<fork handle>

RETURNS +1: Always, with
              AC2: Length of entry vector
              AC3: BO-5   Flags
                    B6-35   Address of entry vector

XGTPW% JSYS 612

FUNCTION
Returns the page-fail words of a process that runs in more than one section of memory.

CALLING SEQUENCE
AC1: Process handle
AC2: Address of argblk

RETURNS +1: Always

ARGUMENT BLOCK

Word	Contents
0	Length of block, including this word
1	0 On return Flags BO(PF%USR) Page failure on user-mode reference B1(PF%WTF) Page failure on write reference
2	0 On return Address that referenced page
3	0 On return MUUO opcode and AC
4	0 On return 30-bit effective address of MUUO

XGVEC% JSYS 606

FUNCTION
Returns the entry vector of a specified process which runs in more than one section of memory.

CALLING SEQUENCE
AC1: Process handle

RETURNS +1: Always, with
              AC2: Length of entry vector
              AC3: address of entry vector

XRIR% JSYS 601

FUNCTION
Reads the addresses of the channel and priority level tables for a process running in more than one section of memory.

CALLING SEQUENCE
AC1: Process handle
AC2: Address of argblk

RETURNS +1: Always

ARGUMENT BLOCK

Word	Contents
0	Length of argblk, including this word
1	Address of interrupt level table
2	Address of channel table

XRMAP% JSYS 611

FUNCTION
Acquires a handle on a page in an extended process to determine the access allowed for that page.

CALLING SEQUENCE
AC1: <process handle>,,0
AC2: Address of argblk

RETURNS +1: Always

ARGUMENT BLOCK

Word	Contents
0	Length of argblk, including this word
1	Number of pages in this group on which to return data On return <process/file designator>,,<page number> (page handle)
2	Number of first page in this group On return Access flags; or -1 if page non-existant

```
 3     Address of block for returned data
n      Number of pages in this group on which to return data
n+1    Number of first page in this group
n+2    Address of block for returned data

ACCESS FLAGS
Bit    Symbol    Meaning
B2     RM%RD     Read access allowed
B3     RM%WR     Write access allowed
B4     RM%EX     Execute access allowed
B5     RM%PEX    Page exists
B9     RM%CPY    Copy-on-write access allowed
```

XSFRK% JSYS 605

```
FUNCTION
Starts the specified process in a non-zero section of
memory.

CALLING SEQUENCE
AC1:   Flags,,<process handle>
       BO(SF%CON)      Continue process that has halted
AC2:   <PC flags>,,0
AC3:   Address to set PC to (ignored if SC%CON on)

RETURNS   +1:  Always
```

XSIR% JSYS 602

```
FUNCTION
Sets the addresses of the channel and priority level tables
for a process running in one or more sectons of memory.

CALLING SEQUENCE
AC1:   Process handle
AC2:   Address of argblk

RETURNS   +1:  Always

ARGUMENT BLOCK
Word   Contents
 0     Length of argblk, including this word
 1     Address of interrupt level table
 2     Address of channel table
```

XSSEV% JSYS 613

FUNCTION
Allows setting of extended special entry vector for use with TOPS-10 Compatibility Package and RMS entry vectors in non-zero sections.

CALLING SEQUENCE

```
AC1:  <vector type code>,,<fork handle>
      0  .XSEVC   TOPS-10 Compatibility
      1  .XSEVD   RMS
AC2:  Length of entry vector
AC3:  B1(XS%EEV)  Extended entry vector; if on, entry vector
                  points to 2-word extended PC and extended
                  format UUO word
      B6-35       Address of entry vector
```

RETURNS +1: Always

XSVEC% JSYS 607

FUNCTION
Sets or clears the entry vector of a process that runs in one or more sections of memory.

CALLING SEQUENCE

```
AC1:  Process handle
AC2:  Length of entry vector; or 0 to clear
AC3:  Address of entry vector
```

RETURNS +1: Always

CONTROL CHARACTER OUTPUT CONTROL (CCOC) WORD

ASCII Code	Wake-up Class	CCOC Word	Character or Control Character
0	C	1B1	CTRL/@ null,break
1	C	1B3	CTRL/A
2	C	1B5	CTRL/B
3	C	1B7	CTRL/C
4	C	1B9	CTRL/D
5	C	1B11	CTRL/E
6	C	1B13	CTRL/F
7	C	1B15	CTRL/G bell
10	F	1B17	CTRL/H backspace
11	P	1B19	CTRL/I horizontal tab
12	F	1B21	CTRL/J line feed
13	C	1B23	CTRL/K vertical tab
14	F	1B25	CTRL/L form feed
15	F	1B27	CTRL/M carriage return
16	C	1B29	CTRL/N
17	C	1B31	CTRL/O
20	C	1B33	CTRL/P
21	C	1B35	CTRL/Q
22	C	2B1	CTRL/R
23	C	2B3	CTRL/S
24	C	2B5	CTRL/T
25	C	2B7	CTRL/U
26	C	2B9	CTRL/V
27	C	2B11	CTRL/W
30	C	2B13	CTRL/X
31	C	2B15	CTRL/Y
32	C	2B17	CTRL/Z
33	all	2B19	ESCAPE (altmode)
34	C	2B21	CTRL/backslash
35	C	2B23	CTRL/right square bracket
36	C	2B25	CTRL/uparrow
37	F	2B27	CTRL/backarrow
40	P		SPACE
41	P		!
42	P		"
43	P		#
44	P		$
45	P		%
46	P		&
47	P		'
50	P		(
51	P		)
52	P		*
53	P		+
54	P		,
55	P		-
56	P		.
57	P		/
60-71	A		0-9
72	P		:
73	P		;
74	P		<

75	P	=
76	P	>
77	P	?
100	P	@
101-132	A	UPPERCASE LETTERS A-Z
133	P	[
134	P	\
135	P	]
136	P	^
137	P	_
140	P	accent grave
141-172	A	lowercase letters a-z
173(1)	P	{
174(1)	P	|
175(1)	P	}
176(1)	P	~
177	all	DELETE (RUBOUT)

A Alphanumeric character	0(00) Ignore (send nothing)
C Non-formatting CTRL/char	1(01) Indicate by ^X
F Formatting CTRL/char	2(10) Send character code
P Punctuation character	3(11) Simulate format action

COMMUNICATIONS PROTOCOLS

Code	Symbol	Meaning
0	.VN20F	RSX20F protocol
1	.VNMCB	MCB DECnet protocol
2	.VND60	DN60 (IBMCOM) protocol
	.VNDDC	DDCMP (DECnet) protocol
3	.VNMOP	MOP (DDCMP maintenance) protocol
4	.VNCNL	Controller loopback
5	.VNCBL	Cable loopback

DEVICE TYPES

Name	Description	Type	Symbol	Units
DSK:	disk structure	0	.DVDSK	no
MTA:	magtape	2	.DVMTA	yes
MT:	logical magtape	2	.DVMTA	yes
LPT:	spooled line printer	7	-	yes
PLPT:	physical line printer	7	.DVLPT	yes
CDR:	spooled card reader	10	-	yes
PCDR:	physical card reader	10	.DVCDR	yes
FE:	front-end pseudo-device	11	.DVFE	no
TTY:	terminal	12	.DVTTY	yes
PTY:	pseudo-terminal	13	.DVPTY	yes
NUL:	null device	15	.DVNUL	no
NET:	ARPA network	16	.DVNET	no
CDP:	spooled card punch	21	-	yes
PCDP:	physical card punch	21	.DVCDP	yes
DCN:	DECnet active component	22	.DVDCN	no
SRV:	DECnet passive component	23	.DVSRV	no

```
+-------------------------------------------------------------+
| Device designator = <600000(.DVDES)+type>,,<unit number>    |
|                     <600000(.DVDES)+type>,,-1 if no units   |
| Terminal designator = 0,,<400000(.TTDES) + TTY number>      |
+-------------------------------------------------------------+
```

DIRECTORY PROTECTION FIELDS

Value	Symbol	Meaning
40	DP%RD	Directory access controlled by individual file access
10	DP%CN	Connecting to directory and changing protection/account allowed
4	DP%CF	Creating files in directory allowed

FILE PROTECTION FIELDS

Value	Symbol	Meaning
40	FP%RD	Read access
20	FP%WR	Write access
10	FP%EX	Execute access
4	FP%APP	Append access
2	FP%DIR	Directory listing access

FILE DESCRIPTOR BLOCK (FDB)

Word	Symbol	Contents
0	.FBHDR	FDB header word
		B0-28 Reserved for DIGITAL
		B29-35(FB%LEN) Length of this file's FDB
1	.FBCTL	B0(FB%TMP) File is temporary
		B1(FB%PRM) File is permanent
		B2(FB%NEX) File does not exist (no file type)
		B3(FB%DEL) File is deleted
		B4(FB%NXF) File does not exist (not yet closed)
		B5(FB%LNG) File is longer than 512 pages
		B6(FB%SHT) Reserved for DIGITAL
		B7(FB%DIR) File is a directory
		B8(FB%NOD) File is not to be backed-up
		B9(FB%BAT) File may have one or more bad pages
		B10(FB%SDR) Directory has subdirectories
		B11(FB%ARC) File has archive status
		B12(FB%INV) File is invisible
		B13(FB%OFF) File is offline
		B14-17(FB%FCF) File class field
		0(.FBNRM) not RMS file
		1(.FBRMS) RMS file
		B18(FB%NDL) File cannot be deleted
2	.FBEXL	Link to FDB of next file with same name but different type
3	.FBADR	Disk address of file index block
4	.FBPRT	File access code: 500000,,<access flags>
5	.FBCRE	Date/time that file was closed after last write
6	.FBAUT	Pointer to file author string
7	.FBGEN	<generation #>,,<internal directory #> if 1B7 of .FBCTL
10	.FBACT	Pointer to alphanumeric account designator string
11	.FBBYV	File I/O flags
		B0-5(FB%RET) Generation retention count
		B6-11(FB%BSZ) File byte size
		B14-17(FB%MOD) Data mode of last file open
		B18-35(FB%PGC) File page count
12	.FBSIZ	Number of bytes in file
13	.FBCRV	File creation date/time
14	.FBWRT	Date/time of last user write
15	.FBREF	Date/time of last non-write access
16	.FBCNT	<# of file writes>,,<# of file references>
17	.FBBKO	Used by DUMPER
20	.FBBK1	Reserved for DIGITAL
21	.FBBK2	Reserved for DIGITAL
22	.FBBBT	Flags,,<# file pages when deleted>
		B1(AR%RAR) User request for file archive
		B2(AR%RIV) System request for involuntary file migration

	B3(AR%NDL)	Do not delete contents of file when archiving
	B4(AR%NAR)	Resist involuntary migration
	B5(AR%EXM)	File exempt from involuntary migration
	B6(AR%1ST)	1st pass of archival-collection run in progress
	B7(AR%RFL)	Restore failed
	B10(AR%WRN)	Warn user of approaching on-line expiration
	B11-17(AR%RSN)	Reason file was moved offline
		.AREXP(1) File expired
		.ARRAR(2) Archiving was requested
		.ARRIR(3) Migration was requested
	B18-35(AR%PSZ)	0,,<# file pages when archived>
23 .FBNET	On-line expiration date/time	
24 .FBUSW	User-settable word	
25 .FBGNL	Address of FDB for next generation of file	
26 .FBNAM	Pointer to filename block	
27 .FBEXT	Pointer to file type block	
30 .FBLWR	Pointer to user-who-last-wrote string	
31 .FBTDT	Archive or collection tape-write date/time	
32 .FBFET	Offline expiration date/time	
33 .FBTP1	Tape ID for first archive or collection run	
34 .FBSS1	<1st tape saveset #>,,<1st tape file #>	
35 .FBTP2	Tape ID for second archive or collection run	
36 .FBSS2	<2nd tape saveset #>,,<2nd tape file #>	

FORK (PROCESS) HANDLES

Value	Symbol	Meaning
400000	.FHSLF	Current process
400000+n	--	Process n, inferior to current process (relative fork handle)
-1	.FHSUP	Superior process
-2	.FHTOP	Top-level process
-3	.FHSAI	Current process and all inferiors
-4	.FHINF	All inferiors of current process
-5	.FHJOB	All processes in job

FLOATING-POINT FORMAT CONTROL

Bit	Symbol	Meaning
B0-1	FL%SGN	Sign control for 1st field; 1st character position used for minus for negative numbers; for positive numbers, 1st character position defined according to: 0 .FLDIG 1st character is digit

		1 .FLSPC 1st character is space
		2 .FLPLS 1st character is plus sign
		3 .FLSPA 1st character is space
B2-3	FL%JUS	Justification control for 1st field
		0 .FLLSP Right justify with leading spaces
		1 .FLLZR Right justify with leading 0's
		2 .FLLAS Right justify with leading asterisks
		3 .FLTSP Left justify with trailing spaces
B4	FL%ONE	Output at least 1 digit in 1st field
B5	FL%DOL	Prefix number with dollar sign ($)
B6	FL%PNT	Output decimal point
B7-8	FL%EXP	3rd (exponent) field control
		0 .FLEXN No exponent field
		1 .FLEXE Output E as 1st character of exponent field
		2 .FLEXD Output D as 1st character of exponent field
		3 .FLEXM Output *10^ as 1st characters of exponent field
B9-10	FL%ESG	Exponent sign control; 1st character position used for minus for negative exponents; for positive exponents, 1st character position defined according to:
		0 .FLDGE 1st character after exponent prefix is digit
		1 .FLPLE 1st character after prefix is plus sign
		2 .FLSPE 1st character after prefix is space
		3 .FLDGT 1st character after exponent prefix is digit
B11	FL%OVL	Use free format on overflow of 1st field and expand exponent on overflow of 3rd field
B13-17	FL%RND	Digit position at which rounding will occur; if 0, rounding occurs at 12th digit; if 37, no rounding occurs
B18-23	FL%FST	Number of characters in 1st field, including $ if FL%DOL set
B24-29	FL%SND	Number of characters in 2nd field
B30-35	FL%THD	Number of characters in 3rd field

I/O IDENTIFIERS

Symbol	LH,,RH	Meaning
--	0,,JFN	Job File Number (file handle)
.PRIIN	0,,100	Primary input designator
.PRIOU	0,,101	Primary output designator
.NULIO	0,,377777	Null designator
.TTDES	0,,400000	Universal terminal designator
.CTTRM	0,,777777	Process's controlling terminal
.DVDES	600000,,xxxxxx	Universal device designator
	777777,,address	Implicit byte pointer
	777777,,777777	Universal default

JFN MODE WORD

Bit	Symbol	Function
B0	TT%OSP	Output suppress control (1 = ignore output; 0 = allow output)
B1	TT%MFF	Has mechanical form feed
B2	TT%TAB	Has mechanical tab
B3	TT%LCA	Has lower case
B4-10	TT%LEN	Page length
B11-17	TT%WID	Page width
B18-23	TT%WAK	Wakeup control on:
	B18	not used
	B19 TT%IGN	Ignore other TT%WAK bits
	B20 TT%WKF	Formatting control character
	B21 TT%WKN	Non-formatting control character
	B22 TT%WKP	Punctuation character
	B23 TT%WKA	Alphanumeric character
B24	TT%ECO	Echo on
B25	TT%ECM	Echo mode
B26	TT%ALK	Accept links
B27	TT%AAD	Accept advice
B28-29	TT%DAM	Terminal data mode
	00 .TTBIN	No translation
	01 .TTASC	Translate both echo and output
	10 .TTATO	Translate output only
	11 .TTATE	Translate echo only
B30	TT%UOC	Upper case output control
	0	Do not indicate
	1	Indicate by 'X
B31	TT%LIC	Lower case input control
	0	No conversion
	1	Convert lower to upper
B32-33	TT%DUM	Duplex mode
	00 .TTFDX	Full duplex
	01	Reserved for DIGITAL
	10 .TTHDX	Character half duplex
	11 .TTLDX	Line half duplex
B34	TT%PGM	Pause-on-command mode
	0	Disable pause-on-command mode
	1	Enable pause-on-command mode
B35	TT%CAR	System carrier state; on if line is dataset and carrier is on

JOB CAPABILITY WORD

Bit	Symbol	Meaning
		B0-8 Process Capabilities
B0	SC%CTC	Process can enable for CTRL/C interrupts
B1	SC%GTB	Process can examine monitor tables with GETAB
B3	SC%LOG	Process can execute protected log functions
B6	SC%SCT	Process can change source of terminal interrupts for other processes

B9-17 Inferior Process Capabilities

B9	SC%SUP	Process can manipulate its superior process
B17	SC%FRZ	Unprocessed software interrupts can cause process to be frozen instead of terminated

B18-35 User Capabilities

B18	SC%WHL	User has WHEEL capability
B19	SC%OPR	User has OPERATOR capability
B20	SC%CNF	User has CONFIDENTIAL INFORMATION ACCESS capability
B21	SC%MNT	User has MAINTENANCE capability
B22	SC%IPC	User has IPCF capability
B23	SC%ENQ	User has ENQ/DEQ capability
B24	SC%NWZ	User has NET WIZARD (ARPAnet) capability
B25	SC%NAS	User has ARPANET ABSOLUTE SOCKET capability
B26	SC%DNA	User has access to DECNET
B27	SC%ANA	User has access to ARPANET

MAGTAPE DEVICE TYPES

Code	Symbol	Type
3	.MTT45	TU45 (default)
17	.MTT70	TU70
20	.MTT71	TU71
21	.MTT72	TU72
13	.MTT77	TU77
19	.MTT78	TU78

MAGTAPE DRIVE TYPES

Code	Symbol	Meaning
1	.TMDR9	9-track tape drive
2	.TMDR7	7-track tape drive

MAGTAPE HARDWARE DATA MODES

Code	Symbol	Meaning
0	.SJDDM	Default system data mode
1	.SJDMC	Dump mode
2	.SJDM6	SIXBIT mode
3	.SJDMA	ANSI ASCII mode
4	.SJDM8	Industry compatible mode
5	.SJDMH	High-density mode (TU70, TU72 only)

MAGTAPE LABEL STATES

Code	Symbol	Meaning
0	.LSUNL	Unlabeled tape
1	.LSPRI	Private tape
2	.LSSCR	Scratch tape
3	.LSUSC	User scratch tape

MAGTAPE LABEL TYPES

Code	Symbol	Meaning
1	.LTUNL	Unlabeled
2	.LTANS	ANSI Standard label
3	.LTEBC	EBCDIC Standard label
4	.LTT20	TOPS-20 Standard label

MAGTAPE RECORD SIZES

Data Mode	Maximum Record Size
System-default	--
Dump	8192. bytes
SIXBIT	49152. bytes
ANSI ASCII	40960. bytes
Industry compatible	32768. bytes
High density	8192. bytes

MAGTAPE RECORDING DENSITIES

Code	Symbol	Meaning
0	.SJDDN	Default system density
1	.SJDN2	200 BPI
2	.SJDN5	556 BPI
3	.SJDN8	800 BPI
4	.SJD16	1600 BPI
5	.SJD62	6250 BPI

PHYSICAL CARD PUNCH (PCDP:) STATUS BITS

Bit	Symbol	Meaning
B10	MO%FER	Fatal error condition
B12	MO%EOF	All pending output has been processed
B13	MO%IOP	Output in progress
B14	MO%SER	Software error
B15	MO%HE	Hardware error
B16	MO%OL	Card-punch off-line
B17	MO%FNX	Card punch doesn't exist
B32	MO%HEM	Stacker full or hopper empty
B33	MO%SCK	Stack check

B34	MO%PCK	Pick check

PHYSICAL CARD READER (PCDR:) STATUS BITS

Bit	Symbol	Meaning
B0	MO%COL	Device is on line
B10	MO%FER	Fatal hardware error
B12	MO%EOF	Card reader at EOF
B13	MO%IOP	I/O in progress
B14	MO%SER	Software error
B15	MO%HE	Hardware error
B16	MO%OL	Device is off line
B17	MO%FNX	Device is nonexistent
B31	MO%SFL	Output stacker full
B32	MO%HEM	Input hopper empty
B33	MO%SCK	Stack check
B34	MO%PCK	Pick check
B35	MO%RCK	Read check

PHYSICAL LINE PRINTER (PLPT:) CONTROL CHARACTERS

ASCII Code	Default Channel	Name	Default Action
11		HT (^I)	Skips to beginning of every 8th column on same line
12	8	LF (^J)	Skips to column 1 on next line; skips last 6 lines of page
13	7	VT (^K)	Skips to column 1 on line at next third of page
14	1	FF (^L)	Skips to column 1 on top of next page
15		CR (^M)	Returns to column 1 of current line; no paper advance
20	2	Half page	Skips to column 1 on next half page
21	3	Alternate lines	Skips to column 1 on next even line
22	4	Three lines	Skips to column 1 on every third line
23	5	Next line	Skips to column 1 on next line; fills last 6 lines of page
24	6	Sixth page	Skips to column 1 on next sixth of page

PHYSICAL LINE PRINTER (PLPT:) STATUS BITS

Bit	Symbol	Meaning
B0	MO%LCP	Lower case printer
B10	MO%FER	Fatal hardware error
B12	MO%EOF	All data sent to printer has been printed
B13	MO%IOP	I/O in progress
B14	MO%SER	Software error
B15	MO%HE	Hardware error
B16	MO%OL	Device is off line
B17	MO%FNX	Device is nonexistent
B30	MO%RPE	RAM parity error
B31	MO%LVU	Optical VFU
B33	MO%LVF	VFU error
B34	MO%LCI	Character interrupt
B35	MO%LPC	Page counter register overflow

PHYSICAL MAGTAPE (MTA:) STATUS BITS

Bit	Symbol	Meaning
B18	MT%ILW	Drive is write protected
B19	MT%DVE	Device error (hung or data late)
B20	MT%DAE	Data error
B21	MT%SER	Suppress automatic error recovery procedures
B22	MT%EOF	Device EOF (file) mark
B23	MT%IRL	Incorrect record length
B24	MT%BOT	Beginning of tape
B25	MT%EOT	End of tape
B26	MT%EVP	Even parity
B29-31	MT%CCT	Character counter if MT%IRL on
B32	MT%NSH	Selected data mode or density not supported by hardware

SOFTWARE DATA MODES

Mode	Symbol	Explanation
0	.GSNRM	Normal mode - allows unit-record output
1	.GSSMB	Small Buffer mode - allows small data segments to be transmitted to terminals
10	.GSIMG	Image mode - sends an "image" of each byte (12-bit)
17	.GSDMP	Dump mode - unbuffered by default

SOFTWARE INTERRUPT CHANNELS

Channel	Symbol	Meaning
0-5		Assignable by user program
6	.ICAOV	Arithmetic overflow (includes NODIV)
7	.ICFOV	Arithmetic floating point overflow (includes FXU)
8.		Reserved for DIGITAL
9.*	.ICPOV	Pushdown list (PDL) overflow
10.	.ICEOF	End of file condition
11.*	.ICDAE	Data error file condition
12.*	.ICQTA	Disk full or quota exceeded when creating new page
13.-14.		Reserved for DIGITAL
15.*	.ICILI	Illegal instruction
16.*	.ICIRD	Illegal memory read
17.*	.ICIWR	Illegal memory write
18.		Reserved for DIGITAL
19.	.ICIFT	Inferior process termination or forced freeze
20.*	.ICMSE	System resources exhausted
21.		Reserved for DIGITAL
22.	.ICNXP	Reference to non-existent page
23.-35.		Assignable by user program

* Channels are panic channels and cannot be completely deactivated

SYSTEM PIDS

PID	Symbol	Meaning
0	.SPIPC	Reserved
1	.SPINF	PID of <SYSTEM>INFO
2	.SPQSR	PID of QUASAR
3	.SPMDA	PID of QSRMDA
4	.SPOPR	PID of ORION
5	.SPNSR	PID of NETSER

SYSTEM TABLES

Name	Index	Contents
APRID		Processor serial number
BLDTD		Date and time system was generated
DBUGSW		Debugging information
	0	State of operator coverage
		0 Unattended
		1 Attended
		2 Debugging
	1	State of BUGCHK handling
		0 Proceed
		1 Breakpoint
DEVCHR	(P1)	Device characteristics word
DEVNAM	(P1)	SIXBIT device name including unit number
DEVUNT	(P1)	B0-17 Job # to which device is assigned; -1 if device is not assigned; or -2 if reserved for device allocator
		B18-35 Unit #; -1 if device has no units
DRMERR		Information on drum errors
	0	Number of recoverable errors
	1 to n	Varies depending on type of drum being used
DSKERR		Information on disk errors
	0	Number of recoverable disk errors
	1 to n	Varies depending on type of disk being used
DWNTIM		Downtime information
	0	Date/time of next scheduled system shutdown
	1	Date/time when system will return
HQLAV		High queue load averages
IMPLT1	i(P2)	ARPANET; 1 fullword for each link
		B0-17 -1 if control link; or internal connection index for NETAWD, NETBAL, NETBTC, NETBUF, NETFSK, NETLSK, NETSTS
		B18-19 00 receive, 10 send, 11 free, 01 delete
		B20-27 Host number
		B28-35 Link number
IMPLT2	i(P2)	ARPANET: 1 fullword for each link
		B0-9 Flags
		B10-17 Byte size of buffer
		B18-35 Address of input buffer
IMPLT3	i(P2)	ARPANET: 1 fullword for each link
		B0-17 Address of output buffer
		B18-35 Message saved for retransmission
IMPLT4	i(P2)	ARPANET: 1 fullword for each link
		B0-17 Address of current buffer
		B18-35 Message allocation in bits
JBONT	Job #	Owning job for CRJOB-created jobs

JOBNAM	Job #	B0-17 Reserved for DIGITAL B18-35 Index into system program tables of system program being used by job
JOBPNM	Job #	SIXBIT name of program running in this job
JOBRT	Job #	CPU time used by job (negative if no such job)
JOBTTY	Job #	B0-17 Controlling terminal line #; or -1 if none B18-35 Reserved for DIGITAL
LOGDES		Logging information
	0	Designator for logging information
	1	Designator for job 0 and error information
LQLAV		Low queue load averages
NETHST	i(P2)	ARPANET: 1 full word for each internal connection; -1 if no foreign host
NETAWD	j(P2)	ARPANET: 1 full word for each internal connection B0-8 Link number B9-17 Unused B18-23 Timeout countdown B24-35 Index to link tables
NETBAL	j(P2)	ARPANET: number of bits allocated to each internal connection
NETBTC	j(P2)	ARPANET: byte count statistics
NETBUF	j(P2)	ARPANET: 1 fullword for each internal connection B0-17 Bytes per buffer B18-35 (Buffer location)-1
NETFSK	j(P2)	ARPANET: foreign socket number (32 bits) for each internal connection
NETLSK	j(P2)	ARPANET: local socket number for each internal connection
NETRDY		ARPANET: operational status table
	0	0 IMP down >0 IMP going down -1 IMP up
	1	0 Network off ≠0 Network on
	2	Flags for NETSER
	3	Time of last NCP cycle up
	4	Last IMP GOING DOWN message B0-15 Reserved for DIGITAL B16-17 0 Panic 1 Scheduled hardware PM 2 Software reload 3 Emergency restart B18-21 # of 5-minute intervals before IMP goes down B22-31 # of 5-minute intervals IMP will be down
	5	Time of last IMP ready drop
	6	Time of last IMP ready up
	7	Time of IMP GOING DOWN message
NCPGS		Number of pages of real (physical) user core available in system (1 word)
NSWPGS		Default swapping pages

PTYPAR		Pseudo-TTY parameter information
	0	B0-17 Number of PTYs in system B18-35 TTY number of first PTY
QTIMES	0 to n	Accumulated runtime of jobs on n scheduler queues
SNAMES	(P3)	SIXBIT name of system program; 0 if entry unused
SNBLKS	(P3)	Number of samples in working set size integral
SPFLTS	(P3)	Total number of page faults of system program
SSIZE	(P3)	Time integral of working set size
STIMES	(P3)	Total runtime of system program
SYMTAB		SIXBIT table names of all GETAB tables
SYSTAT		Monitor statistics
	0	Time with no runnable jobs
	1	Waiting time with 1 or more runnable jobs
	2	Time spent in scheduler
	3	Time spent processing pager traps
	4	Number of drum reads
	5	Number of drum writes
	6	Number of disk reads
	7	Number of disk writes
	10	Number of terminal wakeups
	11	Number of terminal interrupts
	12	Time integral of number of processes in balance set
	13	Time integral of number of runnable processes
	14	Exponential 1-minute average of number of runnable processes
	15	Exponential 5-minute average of number of runnable processes
	16	Exponential 15-minute average of number of runnable processes
	17	Time integral of number of processes waiting for disk
	20	Time integral of number of processes waiting for drum
	21	Number of terminal input characters
	22	Number of terminal output characters
	23	Number of system core management cycles
	24	Time spent doing postpurging
	25	Number of forced balance set process removals
	26	Time integral of number of processes in swap wait
	27	Scheduler overhead time in high precision units
	30	Idle time in high precision units
	31	Lost time in high precision units
	32	User time
	33	Time integral of number of processes on high queue
	34	Time integral of number of processes on low queue

	35		Sum of process disk-write waits
	36		Number of forced adjustments to balance set
	37		Integral of number of reserve pages of all processes in memory
	40		Integral of number of pages on replaceable queue
	41		High precision pager trap time
	42		Number of context switches
	43		Time spent on background tasks
	44		Total system page traps
	45		Total saves from replacement queue
	46		Number of pages removed from memory during system-wide garbage collection
	47		Integral of number of working sets in memory
	50		Integral of number of wait time without swap waits
	51		Count of working set loads
	52		Count of runable processes removed from balance set
	53		Number of pages removed from memory during process-wide garbage collection
SYSVER			ASCIZ string identifying system name, version, and date
TICKPS			Number of clock ticks per second
TTYJOB	line #	B0-17	Job # for which this is controlling terminal; -1 for unassigned line; -2 for line currently being assigned; or job # to which line is assigned
		B18-35	-1 if no process is waiting for input from terminal ≠ -1 if some process is waiting for input

(P<u>n</u>) specifies a set of parallel tables where <u>n</u> is a unique identifier of the set
<u>i</u> specifies an index into a table derived from B24-35 of NETAWD
<u>j</u> specifies an index into a table derived from B0-17 of IMPLT1

TERMINAL CHARACTERISTICS

Number	Terminal	Symbol	Characteristics
0	TTY model 33	.TT33	UPPERCASE only; padding after TAB and FF; page width 72., page length 66.
1	TTY model 35	.TT35	Mechanical FF and TAB; UPPERCASE only; padding after TAB and FF; page

			width 72., page length 66.
2	TTY model 37	.TT37	Lowercase; padding after TAB and FF; page width 72., page length 66.
3	TI/EXECUPORT	.TTEXE	Lowercase; padding after CR only; page width 80., page length 66.
4-7			Reserved for customer
8.	Default	.TTDEF	Lowercase; full padding; page width 72., page length 66.
9.	Ideal	.TTIDL	Mechanical FF and TAB; lowercase; no padding; no specified width or length
10.	VT05	.TTV05	Mechanical TAB; UPPERCASE only; padding after LF and FF; page width 72., page length 20.; cursor control
11.	VT50	.TTV50	UPPERCASE only; no padding; page width 80., page length 12.; cursor control
12.	LA30	.TTL30	UPPERCASE only; full padding; page width 80., page length 66.
13.	GT40	.TTG40	Lowercase; no padding; page width 80., page length 30.
14.	LA36	.TTL36	Lowercase; no padding; page width 132., page length 66.
15.	VT52	.TTV52	Mechanical TAB; lowercase; no padding; page width 80., page length 24.
16.	VT100	.TT100	Mechanical TAB; lowercase; no padding; page width 80., page length 24.; cursor control
17.	LA38	.TTL38	Mechanical TAB; lowercase; no padding; page width 132., page length 66.
18.	LA120	.TT120	Mechanical FF and TAB; lowercase; no padding; page width 132., page length 60.
35.	VT125	.TT125	Mechanical TAB; lowercase; no padding; page width 80., page length 24.; cursor control; graphics capabilities
36.	VK100	.TTK10	Mechanical TAB; lowercase; no padding; page width 80., page length 24.; cursor control; color graphics capabilities

Terminal Characteristics	
Mechanical FF and TAB	Page width and length
Lowercase	Padding after mechanical FF
Padding after CR	Cursor control
Padding after LF	Graphics capabilities
Padding after mechanical TAB	

TERMINAL INTERRUPT CODES

Code	Symbol	Character/Condition
0	.TICBK	CTRL/@ or BREAK
1	.TICCA	CTRL/A
2	.TICCB	CTRL/B
3	.TICCC	CTRL/C
4	.TICCD	CTRL/D
5	.TICCE	CTRL/E
6	.TICCF	CTRL/F
7	.TICCG	CTRL/G
8.	.TICCH	CTRL/H
9.	.TICCI	CTRL/I (TAB)
10.	.TICCJ	CTRL/J (LF)
11.	.TICCK	CTRL/K (vertical TAB)
12.	.TICCL	CTRL/L (FF)
13.	.TICCM	CTRL/M (CR)
14.	.TICCN	CTRL/N
15.	.TICCO	CTRL/O
16.	.TICCP	CTRL/P
17.	.TICCQ	CTRL/Q
18.	.TICCR	CTRL/R
19.	.TICCS	CTRL/S
20.	.TICCT	CTRL/T
21.	.TICCU	CTRL/U
22.	.TICCV	CTRL/V
23.	.TICCW	CTRL/W
24.	.TICCX	CTRL/X
25.	.TICCY	CTRL/Y
26.	.TICCZ	CTRL/Z
27.	.TICES	ESCAPE (altmode)
28.	.TICRB	DELETE (RUBOUT)
29.	.TICSP	SPACE
30.	.TICRF	Dataset carrier off
31.	.TICTI	Typein
32.	.TICTO	Typeout
33.-35.		Reserved for DIGITAL

TIME ZONES

Zone Name	Abbreviation	Left Half
GREENWICH DAYLIGHT TIME	GDT	700000
GREENWICH MEAN TIME	GMT	500000
GREENWICH STANDARD TIME	GST	500000
ATLANTIC DAYLIGHT TIME	ADT	700004
ATLANTIC STANDARD TIME	AST	500004
EASTERN DAYLIGHT TIME	EDT	700005
EASTERN STANDARD TIME	EST	500005
CENTRAL DAYLIGHT TIME	CDT	700006
CENTRAL STANDARD TIME	CST	500006
MOUNTAIN DAYLIGHT TIME	MDT	700007
MOUNTAIN STANDARD TIME	MST	500007
PACIFIC DAYLIGHT TIME	PDT	700010
PACIFIC STANDARD TIME	PST	500010
YUKON DAYLIGHT TIME	YDT	700011
YUKON STANDARD TIME	YST	500011
ALASKA-HAWAII DAYLIGHT TIME	HDT	700012
ALASKA-HAWAII STANDARD TIME	HST	500012
BERING DAYLIGHT TIME	BDT	700013
BERING STANDARD TIME	BST	500013
LOCAL DAYLIGHT TIME	DAYLIGHT	600000

TOPS-20 JSYS ERROR CODES

Note

See TOPS-20 JSYS ERROR MNEMONICS for error strings.

Code	Mnemonic	Code	Mnemonic	Code	Mnemonic
600010	LGINX1	600011	LGINX2	600012	LGINX3
600013	LGINX4	600014	LGINX5	600020	CRJBX1
600021	CRJBX2	600023	CRJBX4	600024	CRJBX5
600025	CRJBX6	600035	LOUTX1	600036	LOUTX2
600045	CACTX1	600046	CACTX2	600055	GJFX1
600056	GJFX2	600057	GJFX3	600060	GJFX4
600061	GJFX5	600062	GJFX6	600063	GJFX7
600064	GJFX8	600065	GJFX9	600066	GJFX10
600067	GJFX11	600070	GJFX12	600071	GJFX13
600072	GJFX14	600073	GJFX15	600074	GJFX16
600075	GJFX17	600076	GJFX18	600077	GJFX19
600100	GJFX20	600101	GJFX21	600102	GJFX22
600103	GJFX23	600104	GJFX24	600107	GJFX27
600110	GJFX28	600112	GJFX30	600113	GJFX31
600114	GJFX32	600115	GJFX33	600116	GJFX34
600117	GJFX35	600120	OPNX1	600121	OPNX2
600122	OPNX3	600123	OPNX4	600124	OPNX5
600125	OPNX6	600126	OPNX7	600127	OPNX8
600130	OPNX9	600131	OPNX10	600133	OPNX12
600134	OPNX13	600135	OPNX14	600136	OPNX15
600137	OPNX16	600140	OPNX17	600141	OPNX18
600142	OPNX19	600143	OPNX20	600144	OPNX21
600145	OPNX22	600150	DESX1	600151	DESX2
600152	DESX3	600153	DESX4	600154	DESX5
600155	DESX6	600156	DESX7	600157	DESX8
600160	CLSX1	600161	CLSX2	600165	RJFNX1
600166	RJFNX2	600167	RJFNX3	600170	DELFX1
600175	SFPTX1	600176	SFPTX2	600177	SFPTX3
600200	CNDIX1	600204	CNDIX5	600210	SFBSX1
600211	SFBSX2	600215	IOX1	600216	IOX2
600217	IOX3	600220	IOX4	600221	IOX5
600222	IOX6	600240	PMAPX1	600241	PMAPX2
600245	SPACX1	600250	FRKHX1	600251	FRKHX2
600252	FRKHX3	600253	FRKHX4	600254	FRKHX5
600255	FRKHX6	600260	SPLFX1	600261	SPLFX2
600262	SPLFX3	600267	GTABX1	600270	GTABX2
600271	GTABX3	600273	RUNTX1	600275	STADX1
600276	STADX2	600300	ASNDX1	600301	ASNDX2
600302	ASNDX3	600320	ATACX1	600321	ATACX2
600322	ATACX3	600323	ATACX4	600324	ATACX5
600332	STDVX1	600335	DEVX1	600336	DEVX2
600337	DEVX3	600350	TERMX1	600351	TLNKX1
600352	ATIX1	600353	ATIX2	600356	TLNKX2
600357	TLNKX3	600360	TTYX1	600361	RSCNX1
600362	RSCNX2	600363	CFRKX3	600365	KFRKX1

600366	KFRKX2	600370	HFRHX1	600371	GFRKX1
600373	GETX1	600374	GETX2	600375	TFRKX1
600376	TFRKX2	600377	SFRVX1	600407	NOUTX1
600410	NOUTX2	600411	TFRKX3	600414	IFIXX1
600415	IFIXX2	600416	IFIXX3	600424	GFDBX1
600425	GFDBX2	600426	GFDBX3	600430	CFDBX1
600431	CFDBX2	600432	CFDBX3	600433	CFDBX4
600440	DUMPX1	600441	DUMPX2	600442	DUMPX3
600443	DUMPX4	600450	RNAMX1	600451	RNAMX2
600452	RNAMX3	600453	RNAMX4	600454	BKJFX1
600460	TIMEX1	600461	ZONEX1	600462	ODTNX1
600464	DILFX1	600465	TILFX1	600466	DATEX1
600467	DATEX2	600470	DATEX3	600471	DATEX4
600472	DATEX5	600473	DATEX6	600516	SMONX1
600530	SACTX1	600531	SACTX2	600532	SACTX3
600533	SACTX4	600540	GACTX1	600541	GACTX2
600544	FFUFX1	600545	FFUFX2	600546	FFUFX3
600555	DSMX1	600570	SIRX1	600600	SSAVX1
600601	SSAVX2	600610	SEVEX1	600614	WHELX1
600615	CAPX1	600617	PEEKX2	600620	CRDIX1
600621	CRDIX2	600622	CRDIX3	600623	CRDIX4
600624	CRDIX5	600626	CRDIX7	600640	GTDIX1
600641	GTDIX2	600650	FLINX1	600651	FLINX2
600652	FLINX3	600653	FLINX4	600660	FLOTX1
600661	FLOTX2	600662	FLOTX3	600670	HPTX1
600704	GTHSX1	600705	GTHSX2	600707	GTHSX3
600710	ATNX1	600711	ATNX2	600712	ATNX3
600713	ATNX4	600714	ATNX5	600715	ATNX6
600716	ATNX7	600717	ATNX8	600720	ATNX9
600721	ATNX10	600722	ATNX11	600723	ATNX12
600724	ATNX13	600727	CVHST1	600730	CVSKX1
600731	CVSKX2	600732	SNDIX1	600733	SNDIX2
600734	SNDIX3	600735	SNDIX4	600736	SNDIX5
600737	NTWZX1	600740	ASNSX1	600741	ASNSX2
600742	SQX1	600743	SQX2	600746	GTNCX1
600747	GTNCX2	600750	RNAMX5	600751	RNAMX6
600752	RNAMX7	600753	RNAMX8	600754	RNAMX9
600755	RNMX10	600756	RNMX11	600757	RNMX12
600760	GJFX36	600770	ILINS1	600771	ILINS2
600772	ILINS3	601000	CRLNX1	601001	INLNX1
601002	LNSTX1	601010	RDTX1	601011	GFKSX1
601013	GTJIX1	601014	GTJIX2	601015	GTJIX3
601016	IPCFX1	601017	IPCFX2	601020	IPCFX3
601021	IPCFX4	601022	IPCFX5	601023	IPCFX6
601024	IPCFX7	601025	IPCFX8	601026	IPCFX9
601027	IPCF10	601030	IPCF11	601031	IPCF12
601032	IPCF13	601033	IPCF14	601034	IPCF15
601035	IPCF16	601036	IPCF17	601037	IPCF18
601040	IPCF19	601041	IPCF20	601042	IPCF21
601043	IPCF22	601044	IPCF23	601045	IPCF24
601046	IPCF25	601047	IPCF26	601050	IPCF27
601051	IPCF28	601052	IPCF29	601053	IPCF30
601054	GNJFX1	601055	ENQX1	601056	ENQX2
601057	ENQX3	601060	ENQX4	601061	ENQX5
601062	ENQX6	601063	ENQX7	601064	ENQX8
601065	ENQX9	601066	ENQX10	601067	ENQX11

601070	ENQX12	601071	ENQX13	601072	ENQX14
601073	ENQX15	601074	ENQX16	601075	ENQX17
601076	ENQX18	601077	ENQX19	601100	ENQX20
601101	ENQX21	601102	IPCF31	601103	IPCF32
601104	PMAPX3	601105	PMAPX4	601106	PMAPX5
601107	PMAPX6	601110	SNOPX1	601111	SNOPX2
601112	SNOPX3	601113	SNOPX4	601114	SNOPX5
601115	SNOPX6	601116	SNOPX7	601117	SNOPX8
601120	SNOPX9	601121	SNOP10	601122	SNOP11
601123	SNOP12	601124	SNOP13	601125	SNOP14
601126	SNOP15	601127	SNOP16	601130	IPCF33
601131	SNOP17	601132	OPNX23	601133	GJFX37
601134	CRLNX2	601135	INLNX2	601136	LNSTX2
601137	ALCX1	601140	ALCX2	601141	ALCX3
601142	ALCX4	601143	ALCX5	601144	SPLX1
601145	SPLX2	601146	SPLX3	601147	SPLX4
601150	SPLX5	601151	CLSX3	601152	CRLNX3
601153	ALCX6	601154	CKAX1	601155	CKAX2
601156	CKAX3	601157	TIMX1	601160	TIMX2
601161	TIMX3	601162	TIMX4	601163	SNOP18
601164	GJFX38	601165	GJFX39	601166	CRDIX8
601167	CRDIX9	601170	CRDI10	601171	DELDX1
601172	DELDX2	601173	GACTX3	601174	DIAGX1
601175	DIAGX2	601176	DIAGX3	601177	DIAGX4
601200	DIAGX5	601201	DIAGX6	601202	DIAGX7
601203	DIAGX8	601204	DIAGX9	601205	DIAG10
601206	SYEX1	601207	SYEX2	601210	MTOX1
601211	IOX7	601212	IOX8	601213	MTOX5
601214	DUMPX5	601215	DUMPX6	601216	IOX9
601217	CLSX4	601220	MTOX2	601221	MTOX3
601222	MTOX4	601223	MTOX6	601224	OPNX25
601225	GJFX40	601226	MTOX7	601227	LOUTX3
601230	LOUTX4	601231	CAPX2	601232	SSAVX3
601233	SSAVX4	601234	TDELX1	601235	TADDX1
601236	TADDX2	601237	TLUKX1	601240	IOX10
601244	SJBX1	601245	SJBX2	601246	SJBX3
601247	TMONX1	601250	SMONX2	601251	SJBX4
601252	SJBX5	601253	SJBX6	601254	GTJIX4
601255	ILINS4	601256	ILINS5	601257	COMNX1
601260	COMNX2	601261	COMNX3	601263	PRAX1
601264	PRAX2	601265	COMNX5	601270	PRAX3
601271	CKAX4	601272	GACCX1	601273	GACCX2
601274	MTOX8	601275	DBRKX1	601276	SJPRX1
601277	GJFX41	601300	GJFX42	601301	GACCX3
601302	TIMEX2	601303	DELFX2	601304	DELFX3
601305	DELFX4	601306	DELFX5	601307	DELFX6
601310	DELFX7	601311	DELFX8	601312	FRKHX7
601313	DIRX1	601314	DIRX2	601315	DIRX3
601316	UFPGX1	601317	LNGFX1	601320	IPCF34
601321	COMNX8	601322	MTOX9	601323	MTOX10
601324	MTOX11	601325	MTOX12	601326	MTOX13
601327	MTOX14	601330	SAVX1	601331	MTOX15
601332	MTOX16	601333	LPINX1	601334	LPINX2
601335	LPINX3	601336	MTOX17	601337	LGINX6
601340	DESX9	601341	ACESX1	601343	DSKOX1
601344	DSKOX2	601345	MSTRX1	601346	MSTRX2

601347	MSTRX3	601350	MSTRX4	601351	MSTRX5
601352	MSTRX6	601353	MSTRX7	601354	MSTRX8
601355	MSTRX9	601356	MSTX10	601357	MSTX11
601360	MSTX12	601361	MSTX13	601362	MSTX14
601363	MSTX15	601364	MSTX16	601365	DSKX01
601367	DSKX03	601371	GFUSX1	601372	GFUSX2
601373	SFUSX1	601374	SFUSX2	601376	RCDIX1
601377	RCDIX2	601400	RCDIX3	601401	RCDIX4
601402	RCUSX1	601403	TDELX2	601404	TIMX5
601405	LSTRX1	601406	SWJFX1	601407	MTOX18
601410	OPNX26	601411	DELFX9	601412	CRDIX6
601413	COMNX9	601414	STYPX1	601415	PMAPX7
601416	DSKOX3	601417	DESX10	601420	DSKOX4
601421	MSTX17	601422	MSTX18	601423	MSTX19
601424	MSTX20	601425	MSTX21	601426	MSTX22
601427	CRDI11	601430	MSTX23	601431	ACESX3
601432	ACESX4	601433	ACESX5	601435	ACESX6
601436	STRXO1	601437	STRXO2	601440	IOX11
601441	IOX12	601442	STRXO3	601443	STRXO4
601444	PPNX1	601445	PPNX2	601450	SPLX6
601451	CRDI12	601452	GFUSX3	601453	GFUSX4
601454	RNMX13	601455	SJBX8	601456	DECRSV
601460	WILDX1	601461	MSTX41	601462	MSTX42
601475	LCBDBP	601477	LCNOND	601500	SSAVX5
601502	ATACX6	601503	ATACX7	601533	DSKOX5
601534	DSKOX6	601535	TIMX6	601536	TIMX7
601537	TIMX8	601540	TIMX9	601541	TIMX10
601550	SCTX1	601551	SCTX2	601552	SCTX3
601553	SCTX4	601554	PDVXO1	601555	PDVXO2
601556	PDVXO3	601557	GETX4	601560	GETX5
601700	SFUSX4	601701	SFUSX5	601702	SFUSX6
601703	GETX3	601706	CAPX3	601713	ARGXO2
601715	ARGXO4	601716	ARGXO5	601717	ARGXO6
601720	ARGXO7	601721	ARGXO8	601722	ARGXO9
601723	ARGX10	601725	ARGX12	601726	ARGX13
601727	MONXO1	601730	MONXO2	601731	MONXO3
601732	MONXO4	601733	ARGX14	601734	ARGX15
601741	ARGX16	601742	ARGX17	601743	ARGX18
601744	DEVX5	601747	STRXO6	601750	MSTX24
601751	MSTX25	601752	MSTX26	601753	LOUTX5
601754	GJFX43	601755	MTOX19	601756	MTOX20
601757	MSTX27	601760	MSTX28	601761	MSTX29
601763	DSKXO5	601764	DSKXO6	601765	DSKXO7
601766	DSKXO8	601767	COMX10	601770	MSTX30
601771	LOCKX1	601772	LOCKX2	601774	ILLXO1
601775	ILLXO2	601776	ILLXO3	601777	ILLXO4
602000	MSTX31	602001	MSTX32	602002	MSTX33
602003	STDIX1	602004	CNDIX7	602005	PMCLX1
602006	PMCLX2	602007	PMCLX3	602010	DLFX10
602011	DLFX11	602012	GJFX44	602013	UTSTX1
602014	UTSTX2	602015	UTSTX3	602016	BOTXO1
602017	BOTXO2	602020	DCNX1	602021	DCNX5
602022	DCNX3	602023	DCNX4	602024	DCNX9
602025	DCNX8	602026	DCNX11	602027	DCNX12
602030	TTYXO1	602031	BOTXO3	602032	MONXO5
602033	ARGX19	602035	COMX11	602036	COMX12

602037	COMX13	602040	COMX14	602041	COMX15
602042	COMX16	602043	COMX17	602044	NPXAMB
602045	NPXNSW	602046	NPXNOM	602047	NPXNUL
602050	NPXINW	602051	NPXNC	602052	NPXICN
602053	NPXIDT	602054	NPXNQS	602055	NPXNMT
602056	NPXNMD	602057	NPXCMA	602060	GJFX45
602061	GJFX46	602062	GJFX47	602063	MSTX34
602064	GJFX48	602065	GJFX49	602077	SJBX7
602100	DELF10	602101	CRDI13	602102	CRDI14
602103	CRDI15	602104	CRDI16	602105	ENACX1
602106	ENACX2	602107	ENACX3	602110	ENACX4
602111	VACCX0	602112	VACCX1	602113	USGX01
602114	BOTX04	602116	USGX02	602117	CRDI17
602120	ENQX23	602121	ENQX22	602122	DCNX2
602123	ABRKX1	602124	USGX03	602125	IPCF35
602126	VACCX2	602127	CRDI18	602130	CRDI19
602132	BOTX05	602133	CRDI20	602134	COMX18
602135	COMX19	602136	CRDI21	602137	ACESX7
602140	CRDI22	602141	CRDI23	602142	STRX07
602143	STRX08	602144	CRDI24	602165	PMCLX4
602170	FRKHX8	602171	ARGX20	602172	ARGX21
602173	ARGX22	602177	ARGX23	602200	ARGX24
602201	MSTX35	602202	DCNX13	602203	DCNX14
602204	DCNX15	602205	GJFX50	602206	KDPX01
602207	NODX02	602210	NODX03	602211	GJFX51
602212	COMX20	602220	GOKER1	602221	GOKER2
602222	STRX09	602223	MSTX36	602224	MSTX37
602225	MSTX40	602227	IOX13	602230	IOX14
602231	IOX15	602233	IOX17	602234	IOX20
602235	IOX21	602236	IOX22	602237	IOX23
602240	IOX24	602241	IOX25	602242	SWJFX2
602243	IOX26	602245	IOX30	602246	ARGX25
602247	SKDX1	602275	DEVX6	602310	DATEX7
602312	ARCFX2	602313	ARCFX3	602314	ARCFX4
602315	ARCFX5	602316	ARCFX6	602317	ARCFX7
602320	ARCFX8	602321	ARCFX9	602322	ARCX10
602323	ARCX11	602324	ARCX12	602325	ARCX13
602326	OPNX30	602327	OPNX31	602330	DELX11
602331	DELX12	602332	ARCX14	602333	ARCX15
602334	ARCX16	602335	ARCX17	602336	ARCX18
602337	ARCX19	602340	ARGX26	602341	ARGX27
602342	DIRX5	602343	IOX31	602347	LTLBLX
602352	METRX1	602353	NSPX00	602354	NSPX01
602355	NSPX02	602356	NSPX03	602357	NSPX04
602360	NSPX05	602361	NSPX06	602362	NSPX07
602363	NSPX08	602364	NSPX09	602365	NSPX10
602366	NSPX11	602367	NSPX12	602370	NSPX13
602371	NSPX14	602372	NSPX15	602373	NSPX16
602374	NSPX17	602375	NSPX18	602376	NSPX19
602377	NSPX20	602400	NSPX21	602401	NSPX22
602406	DIAG11	602407	DIAG12	602410	DESX11
602411	NSPX23	602412	ARGX28	602413	NPX2CL
602414	ARGX29	602415	ARGX30	602416	ARGX31
602417	DEVX7	602420	GJFX52	602421	GOKER3
602422	IOX32	602423	IOX33	602424	XSIRX1
602425	SIRX2	602426	RIRX1	602427	XSIRX2

602431	SMAPX1	602432	TTMSX1	602433	MONXO6
602434	BOTXO6	602435	BOTXO7	602436	BOTXO8
602437	BOTXO9	602440	BOTX10	602441	BOTX11
602442	BOTX12	602443	BOTX13	602444	BOTX14
602445	BOTX15	602446	BOTX16	602447	BOTX17
602450	BOTX18	602451	NTMX1	602452	COMX21
602453	DELX13	602454	ANTXO1	602455	TTYXO2
602456	NSPX24	602457	NSPX25	602460	NSPX26
602461	GJFX53	602462	IOX34	602463	IOX35
602464	PMAPX8	602465	SMAPX2	602467	BOTX19
602470	BOTX20	602471	ILLXO5	602472	XSEVX1
602473	XSEVX2	602474	XSEVX3	602475	ABRKX2
602477	ABRKX4				

TOPS-20 JSYS ERROR MNEMONICS

Note

See TOPS-20 JSYS ERROR CODES for a list of error codes sorted numerically.

JSYS names ([JSYS]) are listed for those error mnemonics that are called from within a particular JSYS module. Error mnemonics not followed by [JSYS] are not called from within any particular JSYS module, but may be returned while a JSYS is executing.

Mnemonic	Code	Text String [JSYS]
ABRKX1:	602123	Address break not available on this system [ADBRK]
ACESX1:	601341	Argument block too small [ACCES]
ACESX3:	601431	Password is required [ACCES; CRDIR]
ACESX4:	601432	Function not allowed for another job [ACCES]
ACESX5:	601433	No function specified for ACCES [ACCES]
ACESX6:	601435	Directory is not accessed [ACCES]
ACESX7:	602137	Directory is "files-only" and cannot be accessed [ACCES]
ALCX1:	601137	Invalid function [ALLOC]
ALCX2:	601140	WHEEL or OPERATOR capability required [ALLOC]
ALCX3:	601141	Device is not assignable [ALLOC]
ALCX4:	601142	Invalid job number [ALLOC]
ALCX5:	601143	Device already assigned to another job [ALLOC]
ALCX6:	601153	Device assigned to user job, but will be given to allocator when released [ALLOC]
ANTXO1:	602454	No more network terminals available [MTOPR]
ARCFX2:	602312	File already has archive status [ARCF]
ARCFX3:	602313	Cannot perform ARCF functions on nonmultiple directory devices [ARCF]
ARCFX4:	602314	File is not on line [ARCF]
ARCFX5:	602315	Files are not on the same device or structure [ARCF]
ARCFX6:	602316	File does not have archive status [ARCF]
ARCFX7:	602317	Invalid parameter for .ARSST [ARCF]
ARCFX8:	602320	Archive not complete [ARCF]
ARCFX9:	602321	File not off line [ARCF]
ARCX10:	602322	Archive prohibited [ARCF]
ARCX11:	602323	Archive requested, modification prohibited

		[ARCF]
ARCX12:	602324	Archive requested, delete prohibited [ARCF]
ARCX13:	602325	Archive system request not completed [ARCF]
ARCX14:	602332	Restore failed [ARCF]
ARCX15:	602333	Migration prohibited [ARCF]
ARCX16:	602334	Cannot exempt off-line, archived, or archive-pending files [ARCF]
ARCX17:	602335	FDB improper format for ARCF [ARCF]
ARCX18:	602336	Retrieval wait cannot be fulfilled for waiting process [ARCF]
ARCX19:	602337	Migration already pending [ARCF]
ARGX02:	601713	Invalid function [ADBRK; BOOT; DSKAS; GTHST%; GTNCP%; NODE; PMCTL; SKED%; USAGE; WILD%; ARCF; METER%]
ARGX04:	601715	Argument block too small [GETOK%; MTU%; SKED%; USAGE; XSIR%]
ARGX05:	601716	Argument block too long [GETOK%; MTU%; USAGE; XSIR%]
ARGX06:	601717	Invalid page number [PMAP; PMCTL; RPACS]
ARGX07:	601720	Invalid job number [ACCES]
ARGX08:	601721	No such job [ACCES; SKED%]
ARGX09:	601722	Invalid byte size [CRLNM; NTMAN%]
ARGX10:	601723	Invalid access requested [TTMSG]
ARGX12:	601725	Invalid process handle [PLOCK]
ARGX13:	601726	Invalid software interrupt channel number [NODE]
ARGX14:	601733	Invalid account identifier
ARGX15:	601734	Job is not logged in [SKED%]
ARGX16:	601741	Password is required
ARGX17:	601742	Invalid argument block length [NTMAN%; TEXTI; XRMAP%]
ARGX18:	601743	Invalid structure name [MSTR]
ARGX19:	602033	Invalid unit number [NODE]
ARGX20:	602171	Invalid arithmetic trap argument [SWTRP%]
ARGX21:	602172	Invalid LUUO trap argument [SWTRP%]
ARGX22:	602173	Invalid flags [PLOCK; WILD%]
ARGX23:	602177	Invalid section number [RSMAP%; SKPIR]
ARGX24:	602200	Invalid count [PLOCK; SKPIR]
ARGX25:	602246	Invalid class [SKED%]
ARGX26:	602340	File is off line [DELDF; GETOK%]
ARGX27:	602341	Offline expiration time cannot exceed system maximum [SFTAD]
ARGX28:	602412	Not available on this system [RSMAP%]
ARGX29:	602414	Invalid class share [SKED%]
ARGX30:	602415	Invalid KNOB value [SKED%]
ARGX31:	602416	Class scheduler already enabled [SKED%]
ASNDX1:	600300	Device is not assignable [ASND]
ASNDX2:	600301	Illegal to assign this device [ASND]
ASNDX3:	600302	No such device [ASND]
ASNSX1:	600740	Insufficient system resources (All special queues in use) [ASNSQ]
ASNSX2:	600741	Link(s) assigned to another special queue [ASNSQ]
ATACX1:	600320	Invalid job number [ATACH; TWAKE]

ATACX2:	600321	Job already attached [ATACH]
ATACX3:	600322	Incorrect user number [ATACH]
ATACX4:	600323	Invalid password [ATACH]
ATACX5:	600324	This job has no controlling terminal [ATACH]
ATACX6:	601502	Terminal is already attached to a job [ATACH]
ATACX7:	601503	Illegal terminal number [ATACH]
ATIX1:	600352	Invalid software interrupt channel number [ATI]
ATIX2:	600353	Control-C capability required [ATI]
ATNX10:	600721	Send JFN is not a network connection [ATNVT]
ATNX11:	600722	Send JFN has been used [ATNVT]
ATNX12:	600723	Send connection has been refused [ATNVT]
ATNX13:	600724	Insufficient system resources (no NVTs) [ATNVT]
ATNX1:	600710	Invalid receive JFN [ATNVT]
ATNX2:	600711	Receive JFN is not open for read [ATNVT]
ATNX3:	600712	Receive JFN is not open [ATNVT]
ATNX4:	600713	Receive JFN is not a network connection [ATNVT]
ATNX5:	600714	Receive JFN has been used [ATNVT]
ATNX6:	600715	Receive connection has been refused [ATNVT]
ATNX7:	600716	Invalid send JFN [ATNVT]
ATNX8:	600717	Send JFN is not open for write [ATNVT]
ATNX9:	600720	Send JFN is not open [ATNVT]
BKJFX1:	600454	Illegal to back up terminal pointer twice [BKJFN]
BOTXO1:	602016	Invalid DTE-20 number [BOOT]
BOTXO2:	602017	Invalid byte size [BOOT]
BOTXO3:	602031	Invalid protocol version number [BOOT]
BOTXO4:	602114	Byte count is not positive [BOOT]
BOTXO5:	602132	Protocol initialization failed [BOOT]
BOTXO6:	602434	GTJFN failed for dump file [BOOT]
BOTXO7:	602435	OPENF failed for dump file [BOOT]
BOTXO8:	602436	Dump failed [BOOT]
BOTXO9:	602437	To -10 error on dump [BOOT]
BOTX10:	602440	To -11 error on dump [BOOT]
BOTX11:	602441	Failed to assign page on dump [BOOT]
BOTX12:	602442	Reload failed [BOOT]
BOTX13:	602443	-11 didn't power down [BOOT]
BOTX14:	602444	-11 didn't power up [BOOT]
BOTX15:	602445	ROM did not ACK the -10 [BOOT]
BOTX16:	602446	-11 boot program did not make it to -11 [BOOT]
BOTX17:	602447	-11 took more than 1 minute to reload; will cause retry [BOOT]
BOTX18:	602450	Unknown BOOT error [BOOT]
BOTX19:	602467	Overdue TO-11 transfer aborted [BOOT]
BOTX20:	602470	Overdue TO-10 transfer aborted [BOOT]
CACTX1:	600045	Invalid account identifier [CACCT]
CACTX2:	600046	Job is not logged in [CACCT; MSTR]
CAPX1:	600615	WHEEL or OPERATOR capability required [ARCF; ACCES; BOOT; GIVOK%; MSFRK; MTU%;

		NTMAN%; PEEK; RCVOK%; SFTAD; SFUST; SKED%; SYERR; USAGE]
CAPX2:	601231	WHEEL, OPERATOR, or MAINTENANCE capability required [HSYS; MSTR; NODE; PMCTL; USRIO]
CAPX3:	601706	WHEEL capability required [UTEST]
CFDBX1:	600430	Invalid displacement [CHFDB]
CFDBX2:	600431	Illegal to change specified bits [CHFDB; SFTAD]
CFDBX3:	600432	Write or owner access required [CHFDB]
CFDBX4:	600433	Invalid value for specified bits [CHFDB]
CFRKX3:	600363	Insufficient system resources [CFORK; PMAP]
CKAX1:	601154	Argument block too small [CHKAC]
CKAX2:	601155	Invalid directory number [CHKAC]
CKAX3:	601156	Invalid access code [CHKAC]
CKAX4:	601271	File is not on disk [CHKAC]
CLSX1:	600160	File is not open [CLOSF]
CLSX2:	600161	File cannot be closed by this process [CLOSF]
CLSX3:	601151	File still mapped [CLOSF]
CLSX4:	601217	Device still active [CLOSF]
CNDIX1:	600200	Invalid password [ACCES]
CNDIX5:	600204	Job is not logged in [ACCES]
CNDIX7:	602004	The CNDIR JSYS has been replaced by ACCES
COMNX1:	601257	Invalid COMND function code [COMND]
COMNX2:	601260	Field too long for internal buffer [COMND]
COMNX3:	601261	Command too long for internal buffer [COMND]
COMNX5:	601265	Invalid string pointer argument [COMND]
COMNX8:	601321	Number base out of range 2-10 [COMND]
COMNX9:	601413	End of input file reached [COMND]
COMX10:	601767	Invalid default string [COMND]
COMX11:	602035	Invalid CMRTY pointer [COMND]
COMX12:	602036	Invalid CMBFP pointer [COMND]
COMX13:	602037	Invalid CMPTR pointer [COMND]
COMX14:	602040	Invalid CMABP pointer [COMND]
COMX15:	602041	Invalid default string pointer [COMND]
COMX16:	602042	Invalid help message pointer [COMND]
COMX17:	602043	Invalid byte pointer in function block [COMND]
COMX18:	602134	Invalid character in node name [COMND]
COMX19:	602135	Too many characters in node name [COMND]
COMX20:	602212	Invalid node name [COMND]
COMX21:	602452	Node name doesn't contain an alphabetic character [COMND]
CRDI10:	601170	Maximum directory number exceeded; index table needs expanding [CRDIR]
CRDI11:	601427	Invalid terminating bracket on directory [CRDIR]
CRDI12:	601451	Structure is not mounted [CRDIR]
CRDI13:	602101	Request exceeds superior directory working quota [CRDIR]
CRDI14:	602102	Request exceeds superior directory permanent quota [CRDIR]
CRDI15:	602103	Request exceeds superior directory subdirectory quota [CRDIR]

CRDI16:	602104	Invalid user group [CRDIR]
CRDI17:	602117	Illegal to create nonfiles-only subdirectory under files-only directory [CRDIR]
CRDI18:	602127	Illegal to delete logged-in directory [CRDIR]
CRDI19:	602130	Illegal to delete connected directory [CRDIR]
CRDI20:	602133	WHEEL, OPERATOR, or requested capability required [CRDIR]
CRDI21:	602136	Working space insufficient for current allocation [CRDIR]
CRDI22:	602140	Subdirectory quota insufficient for existing subdirectories [CRDIR]
CRDI23:	602141	Superior directory does not exist [CRDIR]
CRDI24:	602144	Invalid subdirectory quota [CRDIR]
CRDIX1:	600620	WHEEL or OPERATOR capability required [CRDIR]
CRDIX2:	600621	Illegal to change number of old directory [CRDIR]
CRDIX3:	600622	Insufficient system resources (Job Storage Block full) [CRDIR]
CRDIX4:	600623	Superior directory full [CRDIR]
CRDIX5:	600624	Directory name not given [CRDIR]
CRDIX6:	601412	Directory file is mapped [CRDIR]
CRDIX7:	600626	File(s) open in directory [CRDIR]
CRDIX8:	601166	Invalid directory number [CRDIR]
CRDIX9:	601167	Internal format of directory is incorrect [CRDIR]
CRJBX1:	600020	Invalid parameter or function bit combination [CRJOB]
CRJBX2:	600021	Illegal for created job to enter MINI-EXEC [CRJOB]
CRJBX4:	600023	Terminal is not available [CRJOB]
CRJBX5:	600024	Unknown name for LOGIN [CRJOB]
CRJBX6:	600025	Insufficient system resources [CRJOB]
CRLNX1:	601000	Logical name is not defined [CRLNM]
CRLNX2:	601134	WHEEL or OPERATOR capability required [CRLNM]
CRLNX3:	601152	Invalid function [CRLNM]
CVHST1:	600727	No string for that host number [CVHST; CVSKT]
CVSKX1:	600730	Invalid JFN [CVSKT]
CVSKX2:	600731	Local socket invalid in this context [CVSKT]
DATEX1:	600466	Year out of range [IDCNV; IDTIM; ODTNC]
DATEX2:	600467	Month is not less than 12 [IDCNV; ODTNC]
DATEX3:	600470	Day of month too large [IDCNV; IDTIM; ODTNC]
DATEX4:	600471	Day of week is not less than 7 [ODTNC]
DATEX5:	600472	Date out of range [IDCNV; IDTIM]
DATEX6:	600473	System date and time are not set [ODCNV; ODTIM; SFTAD]
DATEX7:	602310	Julian day is out of range [IDCNV]
DBRKX1:	601275	No interrupts in progress [DEBRK]
DCNX1:	602020	Invalid network file name

DCNX2:	602122	Interrupt message must be read first
DCNX3:	602022	Invalid object
DCNX4:	602023	Invalid task name
DCNX5:	602021	No more logical links available
DCNX8:	602025	Invalid network operation
DCNX9:	602024	Object is already defined
DCNX11:	602026	Link aborted
DCNX12:	602027	String exceeds 16 bytes
DCNX13:	602202	Node not accessible
DCNX14:	602203	Previous interrupt message outstanding
DCNX15:	602204	No interrupt message available
DECRSV:	601456	DEC-reserved bits not zero [DSKOP; RFSTS]
DELDX1:	601171	WHEEL or OPERATOR capability required [DELDF]
DELDX2:	601172	Invalid directory number [DELDF]
DELF10:	602100	Directory still contains subdirectory [DELF]
DELFX1:	600170	Delete access required [DELF; DELNF]
DELFX2:	601303	File cannot be expunged because it is currently open [DELDF; DELF]
DELFX3:	601304	System scratch area depleted; file not deleted [DELF]
DELFX4:	601305	Directory symbol table could not be rebuilt [DELDF; DELF]
DELFX5:	601306	Directory symbol table needs rebuilding [DELDF; DELF]
DELFX6:	601307	Internal format of directory is incorrect [DELDF; DELF; DIRST; GFUST; GTDAL; PPNST; VACCT]
DELFX7:	601310	FDB formatted incorrectly; file not deleted [DELDF; DELF]
DELFX8:	601311	FDB not found; file not deleted [DELDF; DELF]
DELFX9:	601411	File is not a directory file [DELF]
DELX11:	602330	File has archive status, delete is not permitted
DELX12:	602331	File has no pointer to offline storage [DELF]
DELX13:	602453	File is marked "Never Delete" [DELF; DELNF]
DESX1:	600150	Invalid source/destination designator [BIN; BKJFN; BOUT; CFIBF; CFOBF; CHFDB; CLOSF; DELF; DELNF; DEVST; DFIN; DFOUT; DIBE; DIRST; DOBE; DUMPI; DUMPO; DVCHR; ERSTR; FFFFP; FFUFP; FLIN; FLOUT; GACTF; GDSTS; GFUST; GNJFN; GTFDB; GTTYP; JFNS; MTOPR; MTU%; NIN; NOUT; OPENF; PBIN; PBOUT; PMAP; PPNST; PSOUT; RCDIR; RFBSZ; RFPOS; RFPTR; RFTAD; RIN; RLJFN; RNAMF; ROUT; RPACS; SACTF; SCTTY; SDSTS; SFBSZ; SFCOC; SFMOD; SFPTR; SFTAD; SFUST; SIBE; SIN; SINR; SIZEF; SOBE; SOUT; SOUTR; SPACS; STPAR; STPPN; STSTS; STTYP; SWJFN; TLINK]
DESX2:	600151	Terminal is not available to this job [RCDIR; BIN; BKJFN; BOUT; CLOSF; DEVST;

		DFIN; DFOUT; DIRST; DUMPI; DUMPO; FLIN; FLOUT; GACTF; GDSTS; GFUST; GNJFN; JFNS; MTOPR; NIN; NOUT; PBIN; PBOUT; PPNST; PSOUT; RFBSZ; RFPTR; RIN; ROUT; SDSTS; SFBSZ; SFCOC; SFPTR; SFUST; SIN; SIZEF; SOUT; SPACS; STI; STO; STPPN; STSTS; STTYP; SWJFN]
DESX3:	600152	JFN is not assigned [BIN; BKJFN; BOUT; CFIBF; CFOBF; CHFDB; CLOSF; DELF; DELNF; DEVST; DFIN; DFOUT; DIBE; DIRST; DOBE; DUMPI; DUMPO; DVCHR; FFFFP; FFUFP; FLIN; FLOUT; GACTF; GDSTS; GFUST; GNJFN; GTFDB; JFNS; MTOPR; NIN; NOUT; OPENF; PBOUT; PMAP; PPNST; PSOUT; RCDIR; RFBSZ; RFPOS; RFPTR; RFTAD; RIN; RLJFN; RNAMF; ROUT; RPACS; SACTF; SDSTS; SFBSZ; SFCOC; SFMOD; SFPTR; SFTAD; SFUST; SIBE; SIN; SINR; SIZEF; SOBE; SOUT; SOUTR; SPACS; SPJFN; STPAR; STPPN; STSTS; SWJFN; UFPGS; WILD%]
DESX4:	600153	Invalid use of terminal designator or string pointer [CHFDB; CLOSF; DELF; DELNF; DUMPI; DUMPO; DVCHR; FFFFP; FFUFP; GACTF; GDSTS; GFUST; GNJFN; GTFDB; JFNS; MTOPR; OPENF; RCDIR; RFBSZ; RFPTR; RIN; RLJFN; RNAMF; ROUT; RPACS; SACTF; SDSTS; SFBSZ; SFPTR; SFUST; SIZEF; SPACS; STPPN; STSTS; SWJFN; UFPGS]
DESX5:	600154	File is not open [BIN; BKJFN; BOUT; CFIBF; CFOBF; DEQ; DFIN; DFOUT; DIBE; DIRST; DOBE; DUMPI; DUMPO; ENQ; FFFFP; FLIN; FLOUT; GDSTS; GFUST; MTOPR; NIN; NOUT; PBIN; PBOUT; PMAP; PPNST; PSOUT; RFBSZ; RFPOS; RFPTR; RIN; ROUT; RPACS; SDSTS; SFBSZ; SFCOC; SFMOD; SIBE; SIN; SINR; SOBE; SOUT; SOUTR; SPACS; STPAR]
DESX6:	600155	Device is not a terminal [BKJFN, SIBE]
DESX7:	600156	Illegal use of parse-only JFN or output wildcard-designators [CHFDB; DELF; DELNF; FFUFP; GACTF; GFUST; GTFDB; OPENF; PMAP; RCDIR; RFTAD; RNAMF; SFTAD; SFUST; STPPN; UFPGS]
DESX8:	600157	File is not on disk [DEQ; ENQ; ENQC; GFUST; RCDIR; RPACS; SFBSZ; SFPTR; SFUST; SPACS; STPPN; UFPGS]
DESX9:	601340	Invalid operation for this device [DELF; GTJFN; MTOPR; MTU%; SDSTS]
DESX10:	601417	Structure is dismounted [GFUST; RCDIR; SFUST; STPPN]
DESX11:	602410	Invalid operation for this label type [MTOPR]
DEVX1:	600335	Invalid device designator [ALLOC; ASND; DEVST; DVCHR; GDSKC; RELD]
DEVX2:	600336	Device already assigned to another job [ASND; CFIBF; CFOBF; DIBE; DOBE; MTOPR; RELD; RFPOS; SCTTY; SFCOC; SFMOD; SIBE; SOBE; STI; STO; STPAR]

DEVX3:	600337	Device is not on-line
DEVX5:	601744	No such device [BOOT]
DEVX6:	602275	Job has open JFN on device [RELD]
DEVX7:	602417	Null device name given [COMND]
DIAG10:	601205	Subunit does not exist [DIAG]
DIAG11:	602406	Device is already on-line [DIAG]
DIAG12:	602407	Unit not on-line [DIAG]
DIAGX1:	601174	Invalid function [DIAG]
DIAGX2:	601175	Device is not assigned [DIAG]
DIAGX3:	601176	Argument block too small [DIAG]
DIAGX4:	601177	Invalid device type [DIAG]
DIAGX5:	601200	WHEEL, OPERATOR, or MAINTENANCE capability required [DIAG]
DIAGX6:	601201	Invalid channel command list [DIAG]
DIAGX7:	601202	Illegal to do I/O across page boundary [DIAG]
DIAGX8:	601203	No such device [DIAG]
DIAGX9:	601204	Unit does not exist [DIAG]
DILFX1:	600464	Invalid date format [IDTIM; IDTNC]
DIRX1:	601313	Invalid directory number [DIRST; GTDAL, PPNST; VACCT]
DIRX2:	601314	Insufficient system resources [DIRST; GFUST; PPNST]
DIRX3:	601315	Internal format of directory is incorrect [DIRST; GFUST; PPNST; VACCT]
DIRX5:	602342	Directory too large
DLFX10:	602010	Cannot delete directory; file still mapped [DELF]
DLFX11:	602011	Cannot delete directory file in this manner [DELF]
DSKOX1:	601343	Channel number too large [DSKOP]
DSKOX2:	601344	Unit number too large [DSKOP]
DSKOX3:	601416	Invalid structure number [DSKOP]
DSKOX4:	601420	Invalid address type specified [DSKOP]
DSKOX5:	601533	Invalid word count
DSKOX6:	601534	Invalid buffer address
DSKXO1:	601365	Invalid structure number [DSKAS]
DSKXO3:	601367	Bit table has not been initialized [DSKAS]
DSKXO5:	601763	Disk assignments and deassignments are currently prohibited [DSKAS]
DSKXO6:	601764	Invalid disk address [DSKAS]
DSKXO7:	601765	Address cannot be deassigned because it has not been assigned [DSKAS]
DSKXO8:	601766	Address cannot be assigned because it is already assigned [DSKAS]
DSMX1:	600555	File(s) not closed [ASND]
DUMPX1:	600440	Command list error [DUMPI; DUMPO]
DUMPX2:	600441	JFN is not open in dump mode [DUMPI; DUMPO]
DUMPX3:	600442	Address error (too big or crosses end of memory) [DUMPI; DUMPO]
DUMPX4:	600443	Access error (cannot read or write data in memory) [DUMPI; DUMPO]
DUMPX5:	601214	No-wait dump mode not supported for this device [DUMPI; DUMPO]
DUMPX6:	601215	Dump mode not supported for this device

		[DUMPI; DUMPO]
ENACX1:	602105	Account validation data base not completely closed
ENACX2:	602106	Cannot get a JFN for <SYSTEM>ACCOUNTS-TABLE.BIN
ENACX3:	602107	Account validation data base too long
ENACX4:	602110	Cannot get an OFN for <SYSTEM>ACCOUNTS-TABLE.BIN
ENQX10:	601066	Invalid argument block length [DEQ; ENQ; ENQC]
ENQX11:	601067	Invalid software interrupt channel number [DEQ; ENQ; ENQC]
ENQX12:	601070	Invalid number of resources requested [ENQ; ENQC]
ENQX13:	601071	Indirect or indexed byte pointer not allowed [DEQ; ENQ; ENQC]
ENQX14:	601072	Invalid byte size [DEQ; ENQ; ENQC]
ENQX15:	601073	ENQ/DEQ capability required [DEQ; ENQ; ENQC]
ENQX16:	601074	WHEEL or OPERATOR capability required [DEQ; ENQ; ENQC]
ENQX17:	601075	Invalid JFN [DEQ; ENQ; ENQC]
ENQX18:	601076	Quota exceeded [DEQ; ENQ; ENQC]
ENQX19:	601077	String too long [DEQ; ENQ; ENQC]
ENQX1:	601055	Invalid function [DEQ; ENQ; ENQC]
ENQX20:	601100	Locked JFN cannot be closed [CLOSF; DEQ; ENQ; ENQC]
ENQX21:	601101	Job is not logged in [DEQ; ENQC]
ENQX22:	602121	Invalid mask block length [ENQ]
ENQX23:	602120	Mismatched mask block lengths [ENQ]
ENQX2:	601056	Level number too small [DEQ; ENQ; ENQC]
ENQX3:	601057	Request and lock level numbers do not match [DEQ; ENQ; ENQC]
ENQX4:	601060	Number of pool and lock resources do not match [DEQ; ENQ; ENQC]
ENQX5:	601061	Lock already requested [ENQ; ENQC]
ENQX6:	601062	Requested locks are not all locked [DEQ; ENQ; ENQC]
ENQX7:	601063	No ENQ on this lock [DEQ; ENQC]
ENQX8:	601064	Invalid access change requested [ENQ; ENQC]
ENQX9:	601065	Invalid number of blocks specified [DEQ; ENQ; ENQC]
FFUFX1:	600544	File is not open [FFUFP]
FFUFX2:	600545	File is not on multiple-directory device [FFUFP]
FFUFX3:	600546	No used page found [FFUFP]
FLINX1:	600650	First character is not blank or numeric [DFIN; FLIN]
FLINX2:	600651	Number too small [DFIN; FLIN]
FLINX3:	600652	Number too large [DFIN; FLIN]
FLINX4:	600653	Invalid format [DFIN; FLIN]
FLOTX1:	600660	Column overflow in field 1 or 2 [DFOUT; FLOUT]
FLOTX2:	600661	Column overflow in field 3 [DFOUT; FLOUT]
FLOTX3:	600662	Invalid format specified [DFOUT; FLOUT]

FRKHX1:	600250	Invalid process handle [ADBRK; AIC; CLZFF; DIC; DIR; EIR; EPCAP; ERSTR; FFORK; GCVEC; GET; GEVEC; GFRKH; GFRKS; GPJFN; GTRPI; GTRPW; GTRPW; HFORK; IIC; KFORK; MSFRK; PMAP; RCM; RFACS; RFORK; RFRKH; RFSTS; RIR; RIRCM; RMAP; RPACS; RPCAP; RTIW; RUNTM; RWM; SAVE; SCTTY; SCVEC; SETER; SEVEC; SFACS; SFORK; SFRKV; SIR; SIRCM; SKPIR; SPACS; SPJFN; SPLFK; SSAVE; STIW; UTFRK; WFORK; XGVEC%; XRMAP%; XSFRK%; XSIR%; XSVEC%]
FRKHX2:	600251	Illegal to manipulate a superior process [ADBRK; AIC; CLZFF; DIC; DIR; EIR; EPCAP; FFORK; GCVEC; GET; GEVEC; GFRKH; GFRKS; GPJFN; GTRPI; GTRPW; GTRPW; HFORK; IIC; KFORK; MSFRK; PMAP; RCM; RFACS; RFORK; RFRKH; RFSTS; RIR; RIRCM; RPACS; RTIW; RWM; SAVE; SCTTY; SCVEC; SETER; SEVEC; SFACS; SFORK; SFRKV; SIR; SIRCM; SKPIR; SPACS; SPJFN; SSAVE; STIW; UTFRK; WFORK; XGVEC%; XSFRK%; XSIR%; XSVEC%]
FRKHX3:	600252	Invalid use of multiple process handle [ADBRK; AIC; CLZFF; DIC; DIR; EIR; FFORK; GCVEC; GET; GEVEC; GFRKH; GFRKS; GPJFN; GTRPI; GTRPW; GTRPW; IIC; KFORK; MSFRK; PMAP; RCM; RFACS; RFORK; RFRKH; RFSTS; RIR; RIRCM; RPACS; RPCAP; RTIW; RWM; SAVE; SCVEC; SEVEC; SFACS; SFORK; SFRKV; SIR; SIRCM; SKPIR; SPACS; SPJFN; SSAVE; STIW; UTFRK; XGVEC%; XSFRK%; XSIR%; XSVEC%]
FRKHX4:	600253	Process is running [RFACS; SCVEC; SETER; SFACS; SFRKV]
FRKHX5:	600254	Process has not been started [SFORK; XSFRK%]
FRKHX6:	600255	All relative process handles in use [CFORK; GFRKH; GFRKS; RTFRK]
FRKHX7:	601312	Process page cannot exceed 777 [PMAP]
FRKHX8:	602170	Illegal to manipulate an execute-only process [ADBRK; AIC; CFORK; DIC; DIR; EIR; GET; IIC; PMAP; RFACS; SAVE; SCVEC; SDVEC; SETER; SEVEC; SFACS; SFORK; SFRKV; SIR; SIRCM; SPACS; STIW; TFORK; UTFRK; XSFRK%; XSIR%; XSVEC%]
GACCX1:	601272	Invalid job number [GACCT]
GACCX2:	601273	No such job [GACCT]
GACCX3:	601301	Confidential Information Access capability required [GACCT]
GACTX1:	600540	File is not on multiple-directory device [GACTF]
GACTX2:	600541	File expunged [GACTF]
GACTX3:	601173	Internal format of directory is incorrect [GACTF]
GETX1:	600373	Invalid save file format [GET]
GETX2:	600374	System Special Pages Table full [GET]
GETX3:	601703	Illegal to overlay existing pages [GET]
GETX4:	601557	Illegal to specify .GBASE for multisection

		file [GET]
GETX5:	601560	EXE file directory entry specifies a section-crossing [GET]
GFDBX1:	600424	Invalid displacement [GTFDB]
GFDBX2:	600425	Invalid number of words [GTFDB]
GFDBX3:	600426	List access required [GTFDB]
GFKSX1:	601011	Area too small to hold process structure [GFRKS]
GFRKX1:	600371	Invalid process handle [GFRKH]
GFUSX1:	601371	Invalid function [GFUST]
GFUSX2:	601372	Insufficient system resources [GFUST]
GFUSX3:	601452	File expunged [GFUST]
GFUSX4:	601453	Internal format of directory is incorrect [GFUST]
GJFX1:	600055	Desired JFN invalid [GTJFN]
GJFX2:	600056	Desired JFN not available [GTJFN]
GJFX3:	600057	No JFNs available [GTJFN]
GJFX4:	600060	Invalid character in file name [CRLNM; GTJFN]
GJFX5:	600061	Field cannot be longer than 39 characters [CRLNM; GTJFN]
GJFX6:	600062	Device field not in a valid position [CRLNM; GTJFN]
GJFX7:	600063	Directory field not in a valid position [CRLNM; GTJFN]
GJFX8:	600064	Directory terminating delimiter is not preceded by a valid beginning delimiter [CRLNM; GTJFN]
GJFX9:	600065	More than one name field is not allowed [CRLNM; GTJFN]
GJFX10:	600066	Generation number is not numeric [CRLNM; GTJFN]
GJFX11:	600067	More than one generation number field is not allowed [CRLNM; GTJFN]
GJFX12:	600070	More than one account field is not allowed [CRLNM; GTJFN]
GJFX13:	600071	More than one protection field is not allowed [CRLNM; GTJFN]
GJFX14:	600072	Invalid protection [CRLNM; GTJFN]
GJFX15:	600073	Invalid confirmation character [CRLNM; GTJFN]
GJFX16:	600074	No such device [GTJFN]
GJFX17:	600075	No such directory name [GTJFN]
GJFX18:	600076	No such filename [GTJFN]
GJFX19:	600077	No such file type [GTJFN]
GJFX20:	600100	No such generation [GTJFN]
GJFX21:	600101	File was expunged [GTJFN]
GJFX22:	600102	Insufficient system resources (Job Storage Block full) [CRLNM; GTJFN; LNMST; PPNST]
GJFX23:	600103	Exceeded maximum number of files per directory [GTJFN]
GJFX24:	600104	File not found [GTJFN]
GJFX27:	600107	File already exists (new file required) [GTJFN]
GJFX28:	600110	Device is not on-line [GTJFN]
GJFX30:	600112	Account is not numeric [GTJFN]

GJFX31:	600113	Invalid wildcard designator [CRLNM; GTJFN]
GJFX32:	600114	No files match this specification [GTJFN]
GJFX33:	600115	Filename was not specified [GTJFN]
GJFX34:	600116	Invalid character "?" in file specification [GTJFN]
GJFX35:	600117	Directory access privileges required [GTJFN]
GJFX36:	600760	Internal format of directory is incorrect [GTJFN]
GJFX37:	601133	Input deleted [GTJFN]
GJFX38:	601164	File not found because output-only device was specified [GTJFN]
GJFX39:	601165	Logical name loop detected [GTJFN]
GJFX40:	601225	Undefined attribute in file specification [GTJFN]
GJFX41:	601277	File name must not exceed 6 characters [GTJFN]
GJFX42:	601300	File type must not exceed 3 characters [GTJFN]
GJFX43:	601754	More than one ;T specification is not allowed [GTJFN]
GJFX44:	602012	Account string does not match [GTJFN]
GJFX45:	602060	Illegal to request multiple specifications for the same attribute [GTJFN]
GJFX46:	602061	Attribute value is required [GTJFN]
GJFX47:	602062	Attribute does not take a value [GTJFN]
GJFX48:	602064	GTJFN input buffer is empty [GTJFN]
GJFX49:	602065	Invalid attribute for this device [GTJFN]
GJFX50:	602205	Invalid argument for attribute [GTJFN]
GJFX51:	602211	Byte count too small [GTJFN]
GJFX52:	602420	End of tape encountered while searching for file
GJFX53:	602461	Tape label filename specification exceeds 17 characters [GTJFN]
GNJFX1:	601054	No more files in this specification [GNJFN]
GOKER1:	602220	Illegal function [GETOK%]
GOKER2:	602221	Request denied by Access Control Facility [GETOK%]
GOKER3:	602421	JSYS not executed within ACJ fork [GIVOK%; RCVOK%]
GTABX1:	600267	Invalid table number [GETAB]
GTABX2:	600270	Invalid table index [GETAB]
GTABX3:	600271	GETAB privileges required [GETAB]
GTDIX1:	600640	WHEEL or OPERATOR capability required [GTDIR; TTMSG]
GTDIX2:	600641	Invalid directory number [GTDIR]
GTHSX1:	600704	Unknown host number [GTHST%]
GTHSX2:	600705	No number for that host name [GTHST%]
GTHSX3:	600707	No string for that host number [GTHST%]
GTJIX1:	601013	Invalid index [GETJI; GTHST%; GTNCP%]
GTJIX2:	601014	Invalid terminal line number [GETJI]
GTJIX3:	601015	Invalid job number [GETJI]
GTJIX4:	601254	No such job [GETJI]
GTNCX1:	600746	Invalid network JFN [GTNCP%]
GTNCX2:	600747	Invalid or inactive NVT [GTNCP%]

HFRKX1:	600370	Illegal to halt self with HFORK [HFORK]
HPTX1:	600670	Undefined clock number [HPTIM]
IFIXX1:	600414	Radix is not in range 2 to 10 [NIN]
IFIXX2:	600415	First nonspace character is not a digit [NIN]
IFIXX3:	600416	Overflow (number is greater than 2**35) [NIN]
ILINS1:	600770	Undefined operaton code
ILINS2:	600771	Undefined JSYS [ASNSQ; ATNVT; CVHST; CVSKT; FLHST; GTHST; GTNCP; RCVIM; RELSQ; SNDIM]
ILINS3:	600772	UUO simulation facility not available
ILINS4:	601255	UUO simulation is disabled
ILINS5:	601256	RMS facility is not available [GDVEC; SDVEC]
ILLXO1:	601774	Illegal memory read [SFTAD]
ILLXO2:	601775	Illegal memory write
ILLXO3:	601776	Memory data parity error
ILLXO4:	601777	Reference to non-existent page [PLOCK]
ILLXO5:	602471	Illegal memory reference, section greater than 37
INLNX1:	601001	Index is beyond end of logical name table [INLNM]
INLNX2:	601135	Invalid function [INLNM]
IOX1:	600215	File is not open for reading [BIN; PBIN; RIN; SIN; SINR; DUMPI]
IOX2:	600216	File is not open for writing [BOUT; PBOUT; PSOUT; SOUTR; DUMPO; ROUT; SOUT]
IOX3:	600217	Illegal to change pointer for this opening of file [RIN; ROUT]
IOX4:	600220	End of file reached [BIN; DUMPI; PBIN; RIN; SIN; SINR; MTOPR]
IOX5:	600221	Device or data error [BIN; BOUT; DUMPI; DUMPO; MTOPR; PBIN; PBOUT; PSOUT; RIN; ROUT; SIN; SINR; SOUT; SOUTR]
IOX6:	600222	Illegal to write beyond absolute end of file [PBOUT; PSOUT; ROUT; SOUT; SOUTR; BOUT]
IOX7:	601211	Insufficient system resources (Job Storage Block full) [SIN; SINR; SOUT; SOUTR]
IOX8:	601212	Monitor internal error [MTU%; SIN; SINR; SOUT; SOUTR]
IOX9:	601216	Function legal for sequential write only [SOUTR]
IOX10:	601240	Record is longer than user requested [SINR]
IOX11:	601440	Quota exceeded [BOUT; CLOSF; CLZFF; DEVST; DFOUT; DIRST; DUMPO; ERSTR; FLOUT; GTJFN; JFNS; NOUT; PBOUT; PMAP; PPNST; PSOUT; RDTTY; ROUT; SAVE; SOUT; SOUTR; SSAVE; TEXTI; UFPGS]
IOX12:	601441	Insufficient system resources (swapping space full)
IOX13:	602227	Invalid segment type
IOX14:	602230	Invalid segment size
IOX15:	602231	Illegal tape format for dump mode [MTOPR]

IOX17:	602233	Invalid tape label [MTOPR]
IOX20:	602234	Illegal tape record size [MTPOR]
IOX21:	602235	Tape HDR1 missing [MTOPR]
IOX22:	602236	Invalid tape HDR1 sequence number [MTOPR]
IOX23:	602237	Tape label read error [MTOPR]
IOX24:	602240	Logical end of tape encountered [MTOPR]
IOX25:	602241	Invalid tape format [MTOPR]
IOX26:	602243	Tape write date has not expired [MTOPR]
IOX30:	602245	Tape has invalid access character [MTOPR]
IOX31:	602343	Invalid record descriptor in labeled tape [MTOPR]
IOX32:	602422	Tape position is indeterminate [MTOPR]
IOX33:	602423	TTY input buffer full [BOUT; SOUT]
IOX34:	602462	Disk full [BOUT; CLOSF; CLZFF; DEVST; DFOUT; DIRST; DUMPO; ERSTR; FLOUT; GTJFN; JFNS; NOUT; PBOUT; PMAP; PPNST; PSOUT; RDTTY; ROUT; SAVE; SOUT; SOUTR; SSAVE; TEXTI; UFPGS]
IOX35:	602463	Unable to allocate disk - structure damaged [BOUT; CLOSF; CLZFF; DEVST; DFOUT; DIRST; DUMPO; ERSTR; FLOUT; GTJFN; JFNS; NOUT; PBOUT; PMAP; PPNST; PSOUT; RDTTY; ROUT; SAVE; SOUT; SOUTR; SSAVE; TEXTI; UFPGS]
IPCF10:	601027	WHEEL capability required [MUTIL]
IPCF11:	601030	WHEEL or IPCF capability required [MRECV; MSEND; MUTIL]
IPCF12:	601031	No free PID's available [MSEND; MUTIL]
IPCF13:	601032	PID quota exceeded [MSEND; MUTIL]
IPCF14:	601033	No PID's available to this job [MRECV; MSEND; MUTIL]
IPCF15:	601034	No PID's available to this process [MRECV; MSEND; MUTIL]
IPCF16:	601035	Receive and message data modes do not match [MRECV; MUTIL]
IPCF17:	601036	Argument block too small [MUTIL]
IPCF18:	601037	Invalid MUTIL JSYS function [MUTIL]
IPCF19:	601040	No PID for [SYSTEM]INFO [MSEND; MUTIL]
IPCF20:	601041	Invalid process handle [MUTIL]
IPCF21:	601042	Invalid job number [MUTIL]
IPCF22:	601043	Invalid software interrupt channel number [MUTIL]
IPCF23:	601044	[SYSTEM]INFO already exists [MUTIL]
IPCF24:	601045	Invalid message size [MRECV; MSEND; MUTIL]
IPCF25:	601046	PID does not belong to this job [MRECV; MSEND; MUTIL]
IPCF26:	601047	PID does not belong to this process [MRECV; MSEND; MUTIL]
IPCF27:	601050	PID is not defined [MRECV; MSEND; MUTIL]
IPCF28:	601051	PID not accessible by this process [MRECV; MSEND; MUTIL]
IPCF29:	601052	PID already being used by another process [MRECV; MSEND; MUTIL]
IPCF30:	601053	Job is not logged in [MUTIL]
IPCF31:	601102	Invalid page number [MRECV; MSEND]
IPCF32:	601103	Page is not private [MRECV; MSEND; MUTIL]

IPCF33:	601130	Invalid index into system PID table [MUTIL]
IPCF34:	601320	Cannot receive into an existing page [MRECV]
IPCF35:	602125	Invalid IPCF quota [MUTIL]
IPCFX1:	601016	Length of packet descriptor block cannot be less than 4 [MRECV; MSEND]
IPCFX2:	601017	No message for this PID [MRECV; MUTIL]
IPCFX3:	601020	Data too long for user's buffer [MRECV; MUTIL]
IPCFX4:	601021	Receiver's PID invalid [MRECV; MSEND; MUTIL]
IPCFX5:	601022	Receiver's PID disabled [MRECV; MSEND; MUTIL]
IPCFX6:	601023	Send quota exceeded [MSEND; MUTIL]
IPCFX7:	601024	Receiver quota exceeded [MSEND; MUTIL]
IPCFX8:	601025	IPCF free space exhausted [MSEND; MUTIL]
IPCFX9:	601026	Sender's PID invalid [MSEND; MUTIL]
KDPXO1:	602206	KMC11 not running [BOOT]
KFRKX1:	600365	Illegal to kill top level process [KFORK]
KFRKX2:	600366	Illegal to kill self [KFORK]
LCBDBP:	601475	Bad byte pointer passed to LCS
LCNOND:	601477	LCS No such node
LGINX1:	600010	Invalid account identifier [LOGIN]
LGINX2:	600011	Directory is "files-only" and cannot be logged into [ACCES; LOGIN]
LGINX3:	600012	Internal format of directory is incorrect [LOGIN]
LGINX4:	600013	Invalid password [LOGIN]
LGINX5:	600014	Job is already logged in [LOGIN]
LGINX6:	601337	No more job slots available for logging in [LOGIN]
LNGFX1:	601317	Page table does not exist and file not open for write [PMAP; UFPGS]
LNSTX1:	601002	No such logical name [LNMST]
LNSTX2:	601136	Invalid function [LNMST]
LOCKX1:	601771	Illegal to lock other than a private page [PLOCK]
LOCKX2:	601772	Requested page unavailable [PLOCK]
LOUTX1:	600035	Illegal to specify job number when logging out own job [LGOUT]
LOUTX2:	600036	Invalid job number [LGOUT; MSTR]
LOUTX3:	601227	WHEEL or OPERATOR capability required [LGOUT]
LOUTX4:	601230	LOG capability required [LGOUT]
LOUTX5:	601753	Illegal to log out job 0 [LGOUT]
LPINX1:	601333	Invalid unit number [LPINI]
LPINX2:	601334	WHEEL or OPERATOR capability required [LPINI]
LPINX3:	601335	Illegal to load RAM or VFU while device is OPEN [LPINI]
LSTRX1:	601405	Process has not encountered any errors [GETER]
LTLBLX:	602347	Too many user labels
METRX1:	602352	METER% not implemented for this processor [METER%]

MONX01:	601727	Insufficient system resources [GETOK%; MSTR]
MONX02:	601730	Insufficient system resources (JSB full) [PVDOP%; VACCT]
MONX03:	601731	Monitor internal error [CRDIR]
MONX04:	601732	Insufficient system resources (swapping space full)
MONX05:	602032	Insufficient system resources (no resident free space) [GETOK%; MSTR]
MONX06:	602433	Insufficient system resources (no swappable free space) [NODE]
MSTRX1:	601345	Invalid function [MSTR]
MSTRX2:	601346	WHEEL or OPERATOR capability required [MSTR]
MSTRX3:	601347	Argument block too small [MSTR]
MSTRX4:	601350	Insufficient system resources [MSTR]
MSTRX5:	601351	Drive is not on-line [MSTR]
MSTRX6:	601352	Home blocks are bad [MSTR]
MSTRX7:	601353	Invalid structure name [MSTR]
MSTRX8:	601354	Could not get OFN for ROOT-DIRECTORY [MSTR]
MSTRX9:	601355	Could not MAP ROOT-DIRECTORY [MSTR]
MSTX10:	601356	ROOT-DIRECTORY bad [MSTR]
MSTX11:	601357	Could not initialize Index Table [MSTR]
MSTX12:	601360	Could not OPEN Bit Table File [MSTR]
MSTX13:	601361	Backup copy of ROOT-DIRECTORY is bad [MSTR]
MSTX14:	601362	Invalid channel number [MSTR]
MSTX15:	601363	Invalid unit number [MSTR]
MSTX16:	601364	Invalid controller number [MSTR]
MSTX17:	601421	All units in a structure must be of the same type [MSTR]
MSTX18:	601422	No more units in system [MSTR]
MSTX19:	601423	Unit is already part of a mounted structure [MSTR]
MSTX20:	601424	Data error reading HOME blocks [MSTR]
MSTX21:	601425	Structure is not mounted [MSTR]
MSTX22:	601426	Illegal to change specified bits [MSTR]
MSTX23:	601430	Could not write HOME blocks [MSTR]
MSTX24:	601750	Illegal to dismount the System Structure [MSTR]
MSTX25:	601751	Invalid number of swapping pages [MSTR]
MSTX26:	601752	Invalid number of Front-End-Filesystem pages [MSTR]
MSTX27:	601757	Specified unit is not a disk [MSTR]
MSTX28:	601760	Could not initialize bit table for structure [MSTR]
MSTX29:	601761	Could not reconstruct ROOT-DIRECTORY [MSTR]
MSTX30:	601770	Incorrect Bit Table counts on structure [MSTR]
MSTX31:	602000	Structure already mounted [MSTR]
MSTX32:	602001	Structure was not mounted [GTDIR; MSTR]
MSTX33:	602002	Structure is unavailable for mounting [MSTR]
MSTX34:	602063	Unit is write-locked [MSTR]

MSTX35:	602201	Too many units in structure [MSTR]
MSTX36:	602223	Illegal while JFNs assigned [MSTR]
MSTX37:	602224	Illegal while connected to structure [MSTR]
MSTX40:	602225	Invalid PSI channel number given [MSTR]
MSTX41:	601461	Channel does not exist [MSTR]
MSTX42:	601462	Controller does not exist [MSTR]
MTOX10:	601323	VFU or RAM file cannot be OPENed [MTOPR]
MTOX11:	601324	Data too large for buffers [MTOPR]
MTOX12:	601325	Input error or not all data read [MTOPR]
MTOX13:	601326	Argument block too small [MTOPR]
MTOX14:	601327	Invalid software interrupt channel number [MTOPR]
MTOX15:	601331	Device does not have Direct Access [programmable] VFU [MTOPR]
MTOX16:	601332	VFU or Translation RAM file must be on disk [MTOPR]
MTOX17:	601336	Device is not on line [MTOPR]
MTOX18:	601407	Invalid software interrupt channel number [MTOPR]
MTOX19:	601755	Invalid terminal line width [MTOPR]
MTOX1:	601210	Invalid function [MTOPR]
MTOX20:	601756	Invalid terminal line length [MTOPR]
MTOX2:	601220	Record size was not set before I/O was done [MTOPR]
MTOX3:	601221	Function not legal in dump mode [MTOPR]
MTOX4:	601222	Invalid record size [MTOPR]
MTOX5:	601213	Invalid hardware data mode for magtape [MTOPR]
MTOX6:	601223	Invalid magtape density [MTOPR]
MTOX7:	601226	WHEEL or OPERATOR capability required [MTOPR]
MTOX8:	601274	Argument block too long [MTOPR]
MTOX9:	601322	Output still pending [MTOPR]
NODX02:	602207	Line not turned off [NODE]
NODX03:	602210	Another line already looped [NODE]
NOUTX1:	600407	Radix is not in range 2 to 36 [NOUT]
NOUTX2:	600410	Column overflow [NOUT]
NPX2CL:	602413	Two colons required on node name [COMND]
NPXAMB:	602044	Ambiguous [COMND]
NPXCMA:	602057	Comma not given [COMND]
NPXICN:	602052	Invalid character in number [COMND]
NPXIDT:	602053	Invalid device terminator [COMND]
NPXINW:	602050	Invalid guide word [COMND]
NPXNC:	602051	Not confirmed [COMND]
NPXNMD:	602056	Does not match directory or user name [COMND]
NPXNMT:	602055	Does not match token [COMND]
NPXNOM:	602046	Does not match switch or keyword [COMND]
NPXNQS:	602054	Not a quoted string - quote missing at beginning or end [COMND]
NPXNSW:	602045	Not a switch - does not begin with slash [COMND]
NPXNUL:	602047	Null switch or keyword given [COMND]
NSPX00:	602353	Connection not accepted
NSPX01:	602354	Resource allocation failure

NSPX02:	602355	Destination node does not exist
NSPX03:	602356	Node shutting down
NSPX04:	602357	Destination process does not exist
NSPX05:	602360	Invalid process name
NSPX06:	602361	Destination process queue overflow
NSPX07:	602362	Unspecified error
NSPX08:	602363	Connection aborted by third party
NSPX09:	602364	Link aborted by process
NSPX10:	602365	NSP Failure - Flow control violation
NSPX11:	602366	Too many connections to node
NSPX12:	602367	Too many connections to destination process
NSPX13:	602370	Access denied due to unacceptable user name or password
NSPX14:	602371	NSP failure - invalid SERVICES field
NSPX15:	602372	Invalid account
NSPX16:	602373	NSP failure - invalid SEGSIZ field
NSPX17:	602374	Process aborted, timed out, or cancelled request
NSPX18:	602375	No path to destination node
NSPX19:	602376	NSP failure - flow control failure
NSPX20:	602377	NSP failure - invalid DSTADDR
NSPX21:	602400	Disconnect confirmation
NSPX22:	602401	NSP failure - image data field too long
NSPX23:	602411	Invalid NSP reason code
NSPX24:	602456	Node name not assigned to a network node
NSPX25:	602457	Illegal DECnet node number [NODE]
NSPX26:	602460	Table of toplogy watchers is full [NODE]
NTMX1:	602451	Network Management unable to complete request [NTMAN%]
NTWZX1:	600737	NET WIZARD capability required [ASNSQ]
ODTNX1:	600462	Time zone must be USA or Greenwich [ODTNC]
OPNX1:	600120	File is already open [GNJFN; MTU%; OPENF; RLJFN; RNAMF]
OPNX2:	600121	File does not exist [GET; OPENF]
OPNX3:	600122	Read access required [OPENF]
OPNX4:	600123	Write access required [OPENF]
OPNX5:	600124	Execute access required [OPENF]
OPNX6:	600125	Append access required [OPENF]
OPNX7:	600126	Device already assigned to another job [OPENF]
OPNX8:	600127	Device is not on line [MTU%; OPENF]
OPNX9:	600130	Invalid simultaneous access [OPENF; VACCT]
OPNX10:	600131	Entire file structure full [OPENF]
OPNX12:	600133	List access required [OPENF]
OPNX13:	600134	Invalid access requested [OPENF]
OPNX14:	600135	Invalid mode requested [OPENF]
OPNX15:	600136	Read/write access required [OPENF]
OPNX16:	600137	File has bad index block [OPENF; VACCT]
OPNX17:	600140	No room in job for long file page table [OPENF]
OPNX18:	600141	Unit Record Devices are not available [OPENF]
OPNX19:	600142	IMP is not up
OPNX20:	600143	Host is not up
OPNX21:	600144	Connection refused

OPNX22:	600145	Connection byte size does not match
OPNX23:	601132	Disk quota exceeded [OPENF]
OPNX25:	601224	Device is write locked [OPENF; SFTAD]
OPNX26:	601410	Illegal to open a string pointer [OPENF]
OPNX30:	602326	File has archive status, modification is prohibited [ARCF]
OPNX31:	602327	File is off-line [ARCF; OPENF]
PDVXO1:	601554	Address in .POADE must be as large as address in .POADR [PVDOP%]
PDVXO2:	601555	Addresses in .PODAT block must be in strict ascending order [PVDOP%]
PDVXO3:	601556	Address in .POADR must be a program data vector address [PVDOP%]
PEEKX2:	600617	Read access failure on monitor page [PEEK]
PMAPX1:	600240	Invalid access requested [PMAP]
PMAPX2:	600241	Invalid use of PMAP [PMAP]
PMAPX3:	601104	Illegal to move shared page into file [PMAP]
PMAPX4:	601105	Illegal to move file page into process [PMAP]
PMAPX5:	601106	Illegal to move special page into file [PMAP]
PMAPX6:	601107	Disk quota exceeded [PMAP]
PMAPX7:	601415	Illegal to map file on dismounted structure [PMAP]
PMAPX8:	602464	Indirect page map loop detected [PMAP]
PMCLX1:	602005	Invalid page state or state transition [PMCTL]
PMCLX2:	602006	Requested physical page is unavailable [PMCTL]
PMCLX3:	602007	Requested physical page contains errors [PMCTL]
PMCLX4:	602165	No more error information [PMCTL]
PPNX1:	601444	Invalid PPN [PPNST]
PPNX2:	601445	Structure is not mounted [PPNST]
PRAX1:	601263	Invalid PRARG function code [PRARG]
PRAX2:	601264	No room in monitor data base for argument block [PRARG]
PRAX3:	601270	PRARG argument block too large [PRARG]
RCDIX1:	601376	Insufficient system resources [RCDIR]
RCDIX2:	601377	Invalid directory specification [ACCES; RCDIR]
RCDIX3:	601400	Invalid structure name [RCDIR]
RCDIX4:	601401	Monitor internal error [RCDIR; RCUSR]
RCUSX1:	601402	Insufficient system resources [RCUSR]
RDTX1:	601010	Invalid string pointer [RDTTY; TEXTI; WILD%]
RIRX1:	602426	RIR JSYS incompatible with previous XSIR% [RIR]
RJFNX1:	600165	File is not closed [RLJFN]
RJFNX2:	600166	JFN is being used to accumulate filename [RLJFN]
RJFNX3:	600167	JFN is not accessible by this process [RLJFN]
RNAMX1:	600450	Files are not on same device [RNAMF]
RNAMX2:	600451	Destination file expunged [RNAMF]

RNAMX3:	600452	Write or owner access to destination file required [RNAMF]
RNAMX4:	600453	Quota exceeded in destination of rename [RNAMF]
RNAMX5:	600750	Destination file is not closed [RNAMF]
RNAMX6:	600751	Destination file has bad page table [RNAMF]
RNAMX7:	600752	Source file expunged [RNAMF]
RNAMX8:	600753	Write or owner access to source file required [RNAMF]
RNAMX9:	600754	Source file is nonexistent [RNAMF]
RNMX10:	600755	Source file is not closed [RNAMF]
RNMX11:	600756	Source file has bad page table [RNAMF]
RNMX12:	600757	Illegal to rename to self [RNAMF]
RNMX13:	601454	Insufficient system resources [RNAMF]
RSCNX1:	600361	Overflowed rescan buffer, input string truncated [RSCAN]
RSCNX2:	600362	Invalid function code [RSCAN]
RUNTX1:	600273	Invalid process handle -3 or -4 [RUNTM]
SACTX1:	600530	File is not on multiple-directory device [SACTF]
SACTX2:	600531	Insufficient system resources (Job Storage Block full) [SACTF]
SACTX3:	600532	Directory requires numeric account [SACTF]
SACTX4:	600533	Write or owner access required [SACTF]
SAVX1:	601330	Illegal to save files on this device [SAVE]
SCTX1:	601550	Invalid function code [SCTTY]
SCTX2:	601551	Terminal already in use as controlling terminal [SCTTY]
SCTX3:	601552	Illegal to redefine the job's controlling terminal [SCTTY]
SCTX4:	601553	SC%SCT capability required [SCTTY]
SEVEX1:	600610	Entry vector length is not less than 1000 [SEVEC; XSVEC%]
SFBSX1:	600210	Illegal to change byte size for this opening of file [SFBSZ]
SFBSX2:	600211	Invalid byte size [OPENF; SFBSZ]
SFPTX1:	600175	File is not open [SFPTR]
SFPTX2:	600176	Illegal to reset pointer for this file [BKJFN; SFPTR]
SFPTX3:	600177	Invalid byte number [BKJFN; SFPTR]
SFRVX1:	600377	Invalid position in entry vector [SFRKV]
SFUSX1:	601373	Invalid function [SFUST]
SFUSX2:	601374	Insufficient system resources [SFUST]
SFUSX4:	601700	File expunged [SFUST]
SFUSX5:	601701	Write or owner access required [SFUST]
SFUSX6:	601702	No such user name [SFUST]
SIRX1:	600570	Table address is not greater than 20 [SIR; XSIR%]
SIRX2:	602425	SIR JSYS invoked from non-zero section [SIR]
SJBX1:	601244	Invalid function [SETJB]
SJBX2:	601245	Invalid magtape density [SETJB]
SJBX3:	601246	Invalid magtape data mode [SETJB]
SJBX4:	601251	Invalid job number [SETJB]

SJBX5:	601252	Job is not logged in [SETJB]
SJBX6:	601253	WHEEL or OPERATOR capability required [SETJB]
SJBX7:	602077	Remark exceeds 39 characters [SETJB]
SJBX8:	601455	Illegal to perform this function [SETJB]
SJPRX1:	601276	Job is not logged in [SJPRI]
SKDX1:	602247	Cannot change class [SKED%]
SMAPX1:	602431	Attempt to delete a section still shared [SKPIR]
SMAPX2:	602465	Indirect section map loop detected [SKPIR]
SMONX1:	600516	WHEEL or OPERATOR capability required [SMON]
SMONX2:	601250	Invalid SMON function [SMON]
SNDIX1:	600732	Invalid message size [SNDIM]
SNDIX2:	600733	Insufficient system resources (No buffers available) [SNDIM]
SNDIX3:	600734	Illegal to specify NCP links 0 - 72 [SNDIM]
SNDIX4:	600735	Invalid header value for this queue [SNDIM]
SNDIX5:	600736	IMP down [SNDIM]
SNOP10:	601121	Breakpoints already inserted [SNOOP]
SNOP11:	601122	Breakpoints not inserted [SNOOP]
SNOP12:	601123	Invalid format for program name symbol [SNOOP]
SNOP13:	601124	No such program name symbol [SNOOP]
SNOP14:	601125	No such symbol [SNOOP]
SNOP15:	601126	Not enough free pages for snooping [SNOOP]
SNOP16:	601127	Multiply-defined symbol [SNOOP]
SNOP17:	601131	Breakpoint already defined [SNOOP]
SNOP18:	601163	Data page is not private or copy-or-write [SNOOP]
SNOPX1:	601110	WHEEL or OPERATOR capability required [SNOOP]
SNOPX2:	601111	Invalid function [SNOOP]
SNOPX3:	601112	.SNPLC function must be first [SNOOP]
SNOPX4:	601113	Only one .SNPLC function allowed [SNOOP]
SNOPX5:	601114	Invalid page number [SNOOP]
SNOPX6:	601115	Invalid number of pages to lock [SNOOP]
SNOPX7:	601116	Illegal to define breakpoints after inserting them [SNOOP]
SNOPX8:	601117	Breakpoint is not set on instruction [SNOOP]
SNOPX9:	601120	No more breakpoints allowed [SNOOP]
SPACX1:	600245	Invalid access requested [SPACS]
SPLFX1:	600260	Process is not inferior or equal to self [SPLFK]
SPLFX2:	600261	Process is not inferior to self [SPLFK]
SPLFX3:	600262	New superior process is inferior to intended inferior [SPLFK]
SPLX1:	601144	Invalid function [SPOOL]
SPLX2:	601145	Argument block too small [SPOOL]
SPLX3:	601146	Invalid device designator [SPOOL]
SPLX4:	601147	WHEEL or OPERATOR capability required [SPOOL]
SPLX5:	601150	Illegal to specify 0 as generation number

		for first file [SPOOL]
SPLX6:	601450	No directory to write spooled files into [SPOOL]
SQX1:	600742	Special network queue handle out of range [RCVIM; SNDIM]
SQX2:	600743	Special network queue not assigned [RCVIM; SNDIM]
SSAVX1:	600600	Illegal to save files on this device [GET; SSAVE]
SSAVX2:	600601	Page count (left half of table entry) must be negative [SSAVE]
SSAVX3:	601232	Insufficient system resources (Job Storage Block full) [SSAVE]
SSAVX4:	601233	Directory area of EXE file is more than one page [SSAVE]
SSAVX5:	601500	Number of PDVs grew during save [SSAVE]
STADX1:	600275	WHEEL or OPERATOR capability required [STAD]
STADX2:	600276	Invalid date or time [SFTAD; STAD]
STDIX1:	602003	The STDIR JSYS has been replaced by RCDIR and RCUSR
STDVX1:	600332	No such device [MSTR; PPNST; STDEV]
STRXO1:	601436	Structure is not mounted [ACCES; DIRST; MSTR; PPNST; RCDIR; VACCT]
STRXO2:	601437	Insufficient system resources [ACCES; MSTR; STPPN]
STRXO3:	601442	No such directory name [ACCES; STPPN]
STRXO4:	601443	Ambiguous directory specification [ACCES; STPPN]
STRXO6:	601747	No such user number [PPNST]
STRXO7:	602142	Invalid user number [RCUSR]
STRXO8:	602143	Invalid user name [RCUSR]
STRXO9:	602222	Prior structure mount required [ACCES; GNJFN; GTJFN]
STYPX1:	601414	Invalid terminal type [STTYP]
SWJFX1:	601406	Illegal to swap same JFN [SWJFN]
SWJFX2:	602242	Illegal to swap ATS JFN
SYEX1:	601206	Unreasonable SPEAR block size [SYERR]
SYEX2:	601207	No buffer space available for SPEAR [SYERR]
TADDX1:	601235	Table is full [TBADD]
TADDX2:	601236	Entry is already in table [TBADD]
TDELX1:	601234	Table is empty [TBDEL]
TDELX2:	601403	Invalid table entry location [TBDEL]
TERMX1:	600350	Invalid terminal code [ATI; DTI]
TFRKX1:	600375	Invalid function code [TFORK]
TFRKX2:	600376	Unassigned process handle or not immediate inferior [TFORK]
TFRKX3:	600411	Process not frozen [TFORK]
TILFX1:	600465	Invalid time format [IDTIM; IDTNC]
TIMEX1:	600460	Time cannot be greater than 24 hours [HSYS; IDCNV; ODCNV; ODTIM]
TIMEX2:	601302	Downtime cannot be more than 7 days in the future [HSYS]
TIMX1O:	601541	No system date and time [TIMER]
TIMX1:	601157	Invalid function [TIMER]

TIMX2:	601160	Invalid process handle [TIMER]
TIMX3:	601161	Time limit already set [TIMER]
TIMX4:	601162	Illegal to clear time limit [TIMER]
TIMX5:	601404	Invalid software interrupt channel number [TIMER]
TIMX6:	601535	Time has already passed [TIMER]
TIMX7:	601536	No space available for a clock [TIMER]
TIMX8:	601537	User clock allocation exceeded [TIMER]
TIMX9:	601540	No such clock entry found [TIMER]
TLNKX1:	600351	Illegal to set remote to object before object to remote [TLINK]
TLNKX2:	600356	Link was not received within 15 seconds [TLINK]
TLNKX3:	600357	Links full [TLINK]
TLUKX1:	601237	Internal format of table is incorrect [TBLUK]
TMONX1:	601247	Invalid TMON function [TMON]
TTMSX1:	602432	Could not send message within timeout interval [TTMSX1]
TTYXO1:	602030	Line is not active [BKJFN; CFIBF; CFOBF; DIBE; DOBE; GTTYP; MTOPR; OPENF; RFCOC; RFMOD; RFPOS; SFCOC; SFMOD; SFPOS; SIBE; SOBE; STI; STO; STPAR; STTYP; TLINK]
TTYXO2:	602455	Illegal character specified
TTYX1:	600360	Device is not a terminal [STI; STO]
UFPGX1:	601316	File is not opened for write [UFPGS]
USGXO1:	602113	Invalid USAGE entry type code [USAGE]
USGXO2:	602116	Item not found in argument list [USAGE]
USGXO3:	602124	Default item not allowed [USAGE]
UTSTX1:	602013	Invalid function code [UTEST]
UTSTX2:	602014	Area of code too large to test [UTEST]
UTSTX3:	602015	UTEST facility in use by another process [UTEST]
VACCXO:	602111	Invalid account [CACCT; SACTF; VACCT]
VACCX1:	602112	Account string exceeds 39 characters [CACCT; SACTF; VACCT; COMND]
VACCX2:	602126	Account has expired [SACTF; VACCT]
WHELX1:	600614	WHEEL or OPERATOR capability required [DELF; DSKAS; DSKOP; MDDT%; SJPRI; SPRIW; STI]
WILDX1:	601460	Second JFN cannot be wild [WILD%]
XSEVX1:	602472	Illegal entry vector type [XSSEV%; XGSEV%]
XSEVX2:	602473	Invalid entry vector length [XSSEV%]
XSEVX3:	602474	Cannot get extended values with this monitor call [GCVEC; GDVEC]
XSIRX1:	602424	Channel table crosses section boundary [XSIR%]
XSIRX2:	602427	Level table crosses section boundary [XSIR%]
ZONEX1:	600461	Time zone out of range [IDCNV; ODCNV; ODTNC]

POINTER FORMATS

One Word Global Byte Pointer

P,S	SEC	ADDR
0 5	6 17	18 35

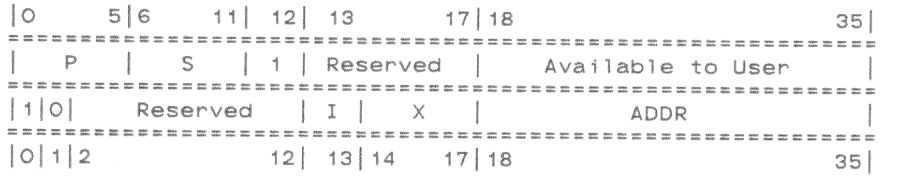
Legal sizes and positions are as follows:

Size	Positions (Octal)	P & S Value (Octal)
6	44	45
	36	46
	30	47
	22	50
	14	51
	6	52
	0	53
7	44	61
	35	62
	26	63
	17	64
	10	65
	1	66
8	44	54
	34	55
	24	56
	14	57
	4	60
9	44	67
	33	70
	22	71
	11	72
	0	73
18	44	74
	22	75
	0	76

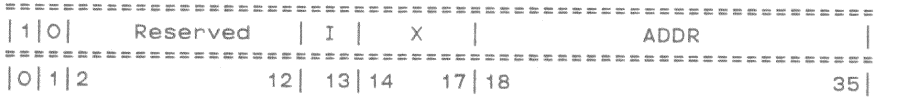
Global Format Indirect Word

0	I	X	SEC	ADDR
0	1	2 5	6 17	18 35

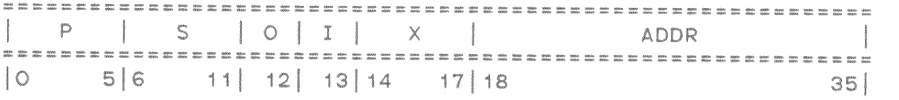
Two-word Local Byte Pointer



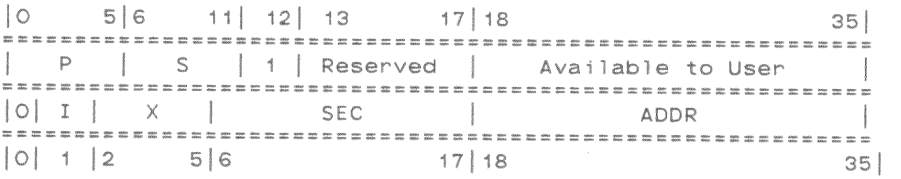
Local Format Indirect Word



One-word Local Byte Pointer



Two-word Global Byte Pointer



PDP-10 INSTRUCTION SET

Arithmetic Testing Instructions

AOBJP	252	(AC) + 1,1 --> (AC); If AC >= 0: E --> (PC)
AOBJN	253	(AC) + 1,1 --> (AC); If AC < 0: E --> (PC)
CAI	300	No-op
CAIL	301	If (AC) < E: skip
CAIE	302	If (AC) = E: skip
CAILE	303	If (AC) <= E: skip
CAIA	304	Skip
CAIGE	305	If (AC) >= E: skip
CAIN	306	If (AC) ≠ E: skip
CAIG	307	If (AC) > E: skip
CAM	310	No-op
CAML	311	If (AC) < (E): skip
CAME	312	If (AC) = (E): skip
CAMLE	313	If (AC) <= (E): skip
CAMA	314	Skip
CAMGE	315	If (AC) >= (E): skip
CAMN	316	If (AC) ≠ (E): skip
CAMG	317	If (AC) >= (E): skip
JUMP	320	No-op
JUMPL	321	If (AC) < 0: E --> (PC)
JUMPE	322	If (AC) = 0: E --> (PC)
JUMPLE	323	If (AC) <= 0: E --> (PC)
JUMPA	324	E --> (PC)
JUMPGE	325	If (AC) >= 0: E --> (PC)
JUMPN	326	If (AC) ≠ 0: E --> (PC)
JUMPG	327	If (AC) > 0: E --> (PC)
SKIP	330	If AC ≠ 0: (E) --> (AC)
SKIPL	331	If (E) < 0: skip; If AC ≠ 0: (E) --> (AC)
SKIPE	332	If (E) = 0: skip; If AC ≠ 0: (E) --> (AC)
SKIPLE	333	If (E) <= 0: skip; If AC ≠ 0: (E) --> (AC)
SKIPA	334	Skip; If AC ≠ 0: (E) --> (AC)
SKIPGE	335	If (E) >= 0: skip; If AC ≠ 0: (E) --> (AC)
SKIPN	336	If (E) ≠ 0: skip; If AC ≠ 0: (E) --> (AC)
SKIPG	337	If (E) > 0: skip; If AC ≠ 0: (E) --> (AC)
AOJ	340	(AC) + 1 --> (AC)
AOJL	341	(AC) + 1 --> (AC); If (AC) < 0: E --> (PC)
AOJE	342	(AC) + 1 --> (AC); If (AC) = 0: E --> (PC)
AOJLE	343	(AC) + 1 --> (AC); If (AC) <= 0: E --> (PC)
AOJA	344	(AC) + 1 --> (AC); E --> (PC)
AOJGE	345	(AC) + 1 --> (AC); If (AC) >= 0: E --> (PC)
AOJN	346	(AC) + 1 --> (AC); If (AC) ≠ 0: E --> (PC)
AOJG	347	(AC) + 1 --> (AC); If (AC) > 0: E --> (PC)

AOS	350	(E) + 1 --> (E); If (AC) ≠ 0: (E) --> (AC)
AOSL	351	(E) + 1 --> (E); If (AC) ≠ 0: (E) --> (AC); If (E) < 0: skip
AOSE	352	(E) + 1 --> (E); If (AC) ≠ 0: (E) --> (AC); If (E) = 0: skip
AOSLE	353	(E) + 1 --> (E); If (AC) ≠ 0: (E) --> (AC); If (E) <= 0: skip
AOSA	354	(E) + 1 --> (E); If (AC) ≠ 0: (E) --> (AC); skip
AOSGE	355	(E) + 1 --> (E); If (AC) ≠ 0: (E) --> (AC); If (E) >= 0: skip
AOSN	356	(E) + 1 --> (E); If (AC) ≠ 0: (E) --> (AC); If (E) ≠ 0: skip
AOSG	357	(E) + 1 --> (E); If (AC) ≠ 0: (E) --> (AC); If (E) > 0: skip
SOJ	360	(AC) - 1 --> (AC)
SOJL	361	(AC) - 1 --> (AC); If (AC) < 0: E --> (PC)
SOJE	362	(AC) - 1 --> (AC); If (AC) = 0: E --> (PC)
SOJLE	363	(AC) - 1 --> (AC); If (AC) <= 0: E --> (PC)
SOJA	364	(AC) - 1 --> (AC); E --> (PC)
SOJGE	365	(AC) - 1 --> (AC); If (AC) >= 0: E --> (PC)
SOJN	366	(AC) - 1 --> (AC); If (AC) ≠ 0: E --> (PC)
SOJG	367	(AC) - 1 --> (AC); If (AC) > 0: E --> (PC)
SOS	370	(E) - 1 --> (E); If AC ≠ 0: (E) --> (AC)
SOSL	371	(E) - 1 --> (E); If AC ≠ 0: (E) --> (AC); If (E) < 0: skip
SOSE	372	(E) - 1 --> (E); If AC ≠ 0: (E) --> (AC); If (E) = 0: skip
SOSLE	373	(E) - 1 --> (E); If AC ≠ 0: (E) --> (AC); If (E) <= 0: skip
SOSA	374	(E) - 1 --> (E); If AC ≠ 0: (E) --> (AC); skip
SOSGE	375	(E) - 1 --> (E); If AC ≠ 0: (E) --> (AC); If (E) >= 0: skip
SOSN	376	(E) - 1 --> (E); If AC ≠ 0: (E) --> (AC); If (E) ≠ 0: skip
SOSG	377	(E) - 1 --> (E); If AC ≠ 0: (E) --> (AC); If (E) > 0: skip

Boolean Instructions

SETZ	400	0 --> (AC)
SETZI	401	0 --> (AC)
SETZM	402	0 --> (E)
SETZB	403	0 --> (AC)(E)
SETM	414	(E) --> (AC)
SETMI	415	0,E --> (AC)
SETMM	416	(E) --> (E) [no-op]
SETMB	417	(E) --> (AC)(E)
SETA	424	(AC) --> (AC) [no-op]
SETAI	425	(AC) --> (AC) [no-op]
SETAM	426	(AC) --> (E)

```
SETAB    427    (AC) --> (E)
SETO     474    777777777777 --> (AC)
SETOI    475    777777777777 --> (AC)
SETOM    476    777777777777 --> (E)
SETOB    477    777777777777 --> (AC)(E)
SETCA    450    ¬(AC) --> (AC)
SETCAI   451    ¬(AC) --> (AC)
SETCAM   452    ¬(AC) --> (E)
SETCAB   453    ¬(AC) --> (AC)(E)
SETCM    460    ¬(E) --> (AC)
SETCMI   461    ¬[0,E] --> (AC)
SETCMM   462    ¬(E) --> (E)
SETCMB   463    ¬(E) --> (AC)(E)

AND      404    (AC) & (E) --> (AC)
ANDI     405    (AC) & 0,E --> (AC)
ANDM     406    (AC) & (E) --> (E)
ANDB     407    (AC) & (E) --> (AC)(E)
ANDCA    410    ¬(AC) & (E) --> (AC)
ANDCAI   411    ¬(AC) & 0,E --> (AC)
ANDCAM   412    ¬(AC) & (E) --> (E)
ANDCAB   413    ¬(AC) & (E) --> (AC)(E)
ANDCM    420    (AC) & ¬(E) --> (AC)
ANDCMI   421    (AC) & ¬[0,E] --> (AC)
ANDCMM   422    (AC) & ¬(E) --> (E)
ANDCMB   420    (AC) & ¬(E) --> (AC)(E)
ANDCB    440    ¬(AC) & ¬(E) --> (AC)
ANDCBI   441    ¬(AC) & ¬[0,E] --> (AC)
ANDCBM   442    ¬(AC) & ¬(E) --> (E)
ANDCBB   443    ¬(AC) & ¬(E) --> (AC)(E)

IOR      434    (AC) ! (E) --> (AC)
IORI     435    (AC) ! 0,E --> (AC)
IORM     436    (AC) ! (E) --> (E)
IORB     437    (AC) ! (E) --> (AC)(E)

ORCA     454    ¬(AC) ! (E) --> (AC)
ORCAI    455    ¬(AC) ! 0,E --> (AC)
ORCAM    456    ¬(AC) ! (E) --> (E)
ORCAB    457    ¬(AC) ! (E) --> (AC)(E)
ORCM     464    (AC) ! ¬(E) --> (AC)
ORCMI    465    (AC) ! ¬[0,E] --> (AC)
ORCMM    466    (AC) ! ¬(E) --> (E)
ORCMB    464    (AC) ! ¬(E) --> (AC)(E)
ORCB     470    ¬(AC) ! ¬(E) --> (AC)
ORCBI    471    ¬(AC) ! ¬[0,E] --> (AC)
ORCBM    472    ¬(AC) ! ¬(E) --> (E)
ORCBB    473    ¬(AC) ! ¬(E) --> (AC)(E)

XOR      430    (AC) ^! (E) --> (AC)
XORI     431    (AC) ^! 0,E --> (AC)
XORM     432    (AC) ^! (E) --> (E)
XORB     433    (AC) ^! (E) --> (AC)(E)

EQV      444    ¬[(AC) ^! (E)] --> (AC)
EQVI     445    ¬[(AC) ^! 0,E] --> (AC)
```

```
EQVM    446     ¬[(AC) ^! (E)] --> (E)
EQVB    447     ¬[(AC) ^! (E)] --> (AC)(E)
```

Byte Instructions

```
IBP     133     Linear operations on pointer in E or E,E+1
        AC=0    if P-S >= 0: P-S --> P
                if P-S < 0: Y+1 --> Y; 36-S --> P

ADJBP   133     Array operations on pointer in E or E,E+1
        AC ≠ 0  Let A = REMAINDER ((36-P)/S)
                If S > 36-A: 1 --> NO DIVIDE
                If S = 0: (E) --> (AC) or
                  (E,E+1) --> (AC,AC+1)
                If 0 < S < 36-A: make copy C of (E) or
                  (E,E+1)
                Compute: (AC)+((36-P)/S) = Q * BYTES/WORD + R
                1 <= R <= BYTES/WORD = ((36-P)/S) + (P/S)
                Y{C} + Q --> Y{C}
                36 - (R * S) - A --> P{C}
                C --> (AC) or (AC,AC+1)

LDB     135     BYTE IN ((E)) --> (AC)
DPB     137     BYTE IN (AC) --> BYTE IN ((E))
ILDB    134     IBP and LDB
IDPB    136     IBP and DPB
```

Fixed-point Arithmetic Instructions

```
ADD     270     (AC) + (E) --> (AC)
ADDI    271     (AC) + 0,E --> (AC)
ADDM    272     (AC) + (E) --> (E)
ADDB    273     (AC) + (E) --> (AC)(E)

SUB     274     (AC) - (E) --> (AC)
SUBI    275     (AC) - 0,E --> (AC)
SUBM    276     (AC) - (E) --> (E)
SUBB    277     (AC) - (E) --> (AC)(E)

IMUL    220     (AC) * (E) --> (AC) [1]
IMULI   221     (AC) * 0,E --> (AC) [1]
IMULM   222     (AC) * (E) --> (E) [1]
IMULB   223     (AC) * (E) --> (AC)(E) [1]

MUL     224     (AC) * (E) --> (AC,AC+1)
MULI    225     (AC) * 0,E --> (AC,AC+1)
MULM    226     (AC) * (E) --> (E) [2]
MULB    227     (AC) * (E) --> (AC,AC+1)(E)

IDIV    230     (AC) / (E) --> (AC); REMAINDER --> (AC+1)
[1]  High order word of product is discarded.
[2]  LOW order word of product is discarded.
```

```
IDIVI   231     (AC) / O,E --> (AC); REMAINDER --> (AC+1)
IDIVM   232     (AC) / (E) --> (E); REMAINDER --> (AC+1)
IDIVB   233     (AC) / (E) --> (AC)(E); REMAINDER --> (AC+1)

DIV     234     (AC,AC+1) / (E) --> (AC);
                  REMAINDER --> (AC+1)
DIVI    235     (AC,AC+1) / O,E --> (AC);
                  REMAINDER --> (AC+1)
DIVM    236     (AC,AC+1) / (E) --> (E); REMAINDER --> (AC+1)
DIVB    237     (AC,AC+1) / (E) --> (AC)(E);
                  REMAINDER --> (AC+1)

DADD    114     (AC,AC+1) + (E,E+1) --> (AC,AC+1)
DSUB    115     (AC,AC+1) - (E,E+1) --> (AC,AC+1)
DMUL    116     (AC,AC+1) * (E,E+1) --> (AC,AC+1,AC+2,AC+3)
DDIV    117     (AC,AC+1,AC+2,AC+3) / (E,E+1) --> (AC,AC+1)
```

Floating-point Arithmetic Instructions

```
FAD     140     (AC) + (E) --> (AC)
FADL    141     (AC) + (E) --> (AC,AC+1)
FADM    142     (AC) + (E) --> (E)
FADB    143     (AC) + (E) --> (AC)(E)
FADR    144     (AC) + (E) --> (AC)
FADRI   145     (AC) + E,O --> (AC)
FADRM   146     (AC) + (E) --> (E)
FADRB   147     (AC) + (E) --> (AC)(E)

FSB     150     (AC) - (E) --> (AC)
FSBL    151     (AC) - (E) --> (AC,AC+1)
FSBM    152     (AC) - (E) --> (E)
FSBB    153     (AC) - (E) --> (AC)(E)
FSBR    154     (AC) - (E) --> (AC)
FSBRI   155     (AC) - E,O --> (AC)
FSBRM   156     (AC) - (E) --> (E)
FSBRB   157     (AC) - (E) --> (AC)(E)

FMP     160     (AC) * (E) --> (AC)
FMPL    161     (AC) * (E) --> (AC,AC+1)
FMPM    162     (AC) * (E) --> (E)
FMPB    163     (AC) * (E) --> (AC)(E)
FMPR    164     (AC) * (E) --> (AC)
FMPRI   165     (AC) * E,O --> (AC)
FMPRM   166     (AC) * (E) --> (E)
FMPRB   167     (AC) * (E) --> (AC)(E)

FDV     170     (AC) / (E) --> (AC)
FDVL    171     (AC) / (E) --> (AC)  Remainder -->(AC+1)
FDVM    172     (AC) / (E) --> (E)
FDVB    173     (AC) / (E) --> (AC)(E)
FDVR    174     (AC) / (E) --> (AC)
FDVRI   175     (AC) / E,O --> (AC)
FDVRM   176     (AC) / (E) --> (E)
```

FDVRB	177	(AC) / (E) --> (AC)(E)
UFA	130	(AC) + (E) --> (AC+1) without normalization
DFN	131	-(AC,E) --> (AC,E)
FSC	132	(AC) * 2** E --> (AC)
GFSC	031	(AC,AC+1) * 2**E --> (AC,AC+1)
FLTR	127	(E) floated, rounded --> (AC)
GFLTR	030	(E) floated, rounded --> (AC,AC+1)
DGFLTR	027	(E,E+1) floated, rounded --> (AC,AC+1)
FIX	122	(E) fixed --> (AC)
FIXR	126	(E) fixed, rounded --> (AC)
GFIX	024	(E,E+1) fixed --> (AC)
GFIXR	026	(E,E+1) fixed, rounded --> (AC)
GDFIX	023	(E,E+1) fixed --> (AC,AC+1)
GDFIXR	025	(E,E+1) fixed, rounded --> (AC,AC+1)
GSNGL	021	(E,E+1) converted --> (AC)
GDBLE	022	(E) converted --> (AC,AC+1)
DFAD	110	(AC,AC+1) + (E,E+1) --> (AC,AC+1)
DFSB	111	(AC,AC+1) - (E,E+1) --> (AC,AC+1)
DFMP	112	(AC,AC+1) * (E,E+1) --> (AC,AC+1)
DFDV	113	(AC,AC+1) / (E,E+1) --> (AC,AC+1)
GFAD	102	(AC,AC+1) + (E,E+1) --> (AC,AC+1)
GFSB	103	(AC,AC+1) - (E,E+1) --> (AC,AC+1)
GFMP	106	(AC,AC+1) * (E,E+1) --> (AC,AC+1)
GFDV	107	(AC,AC+1) / (E,E+1) --> (AC,AC+1)

Fullword Instructions

EXCH	250	(AC) <--> (E)
MOVE	200	(E) --> (AC)
MOVEI	201	0,E --> (AC)
MOVEM	202	(AC) --> (E)
MOVES	203	If AC ≠ 0: (E) --> (AC)
MOVS	204	(E)S-->(AC)
MOVSI	205	E,0 --> (AC)
MOVSM	206	(AC)S --> (E)
MOVSS	207	(E)S --> (E) If AC ≠ 0: (E) --> (AC)
MOVN	210	-(E) --> (AC)
MOVNI	211	-[0,E] --> (AC)
MOVNM	212	-(AC) --> (E)
MOVNS	213	-(E) --> (E) If AC ≠ 0: (E) --> (AC)
MOVM	214	|(E)| --> (AC)
MOVMI	215	0,E --> (AC)
MOVMM	216	|(AC)| --> (E)
MOVMS	217	|(E)| --> (E) If AC ≠ 0: (E) --> (AC)
XMOVEI	415	E --> (AC) Non-local AC reference 1,E --> (AC) Local AC reference

```
DMOVE    120    (E,E+1) --> (AC,AC+1)
DMOVN    121    -(E,E+1) --> (AC,AC+1)
DMOVEM   124    (AC,AC+1) --> (E,E+1)
DMOVNM   125    -(AC,AC+1) --> (E,E+1)

BLT      251    Move E(R) - (AC)R + 1 words starting with:
                  ((AC)L) --> ((AC)R)

XBLT     020    Move |(AC)| words
                If (AC) > 0:
                  start with ((AC+1) --> ((AC+2)) and go up
                If (AC) < 0:
                  start with ((AC+1)-1) --> ((AC+2)-1)
                    and go down
```

Halfword Instructions (Source Left)

```
HLL      500    (E)L --> (AC)L
HLLI     501    0 --> (AC)L
HLLM     502    (AC)L --> (E)L
HLLS     503    If AC ≠ 0: (E) --> (AC)
HLLZ     510    (E)L,0 --> (AC)
HLLZI    511    0 --> (AC)
HLLZM    512    (AC)L,0 --> (E)
HLLZS    513    0 --> (E)R
HLLO     520    (E)L,777777 --> (AC)
HLLOI    521    0,777777 --> (AC)
HLLOM    522    (AC)L,777777 --> (E)
HLLOS    523    777777 --> (E)R; If AC ≠ 0: (E) --> (AC)
HLLE     530    (E)L,[(E)0 * 7777777] --> (AC)
HLLEI    531    0 --> (AC)
HLLEM    532    (AC)L,[(AC)0 * 777777] --> (E)
HLLES    533    (E)0 * 777777 --> (E)R;
                  If AC ≠ 0: (E) --> (AC)
HLR      544    (E)L --> (AC)R
HLRI     545    0 --> (AC)R
HLRM     546    (AC)L --> (E)R
HLRS     547    (E)L --> (E)R; If AC ≠ 0: (E) --> (AC)
HLRZ     554    0,(E)L --> (AC)
HLRZI    555    0 --> (AC)
HLRZM    556    0,(AC)L --> (E)
HLRZS    557    0,(E)L --> (E); If AC ≠ 0: (E) --> (AC)
HLRO     564    777777,(E)L --> (AC)
HLROI    565    777777,0 --> (AC)
HLROM    566    777777,(AC)L --> (E)
HLROS    567    777777,(E)L --> (E); If AC ≠ 0: (E) --> (AC)
HLRE     574    [(E)0 * 777777],(E)L --> (AC)
HLREI    575    0 --> (AC)
HLREM    576    [(AC)0 * 777777],(AC)L --> (E)
HLRES    577    [(E)0 * 777777],(E)L --> (E);
                  If AC ≠ 0: (E) --> (AC)

XHLLI    501    If zero section: E(L) --> (AC)L
                If non-zero section:
```

1. 0 --> (AC bits 0-5)
2. section # --> (AC bits 6-17); if E is a local AC address, section # = 1

Halfword Instructions (Source Right)

HRR	540	(E)R --> (AC)R
HRRI	541	E --> (AC)R
HRRM	542	(AC)R --> (E)R
HRRS	543	If AC ≠ 0: (E) --> (AC)
HRRZ	550	0,(E)R --> (AC)
HRRZI	551	0,E --> (AC)
HRRZM	552	0,(AC)R --> (E)
HRRZS	553	0 --> (E)L
HRRO	560	777777,(E)R --> (AC)
HRROI	561	777777,E --> (AC)
HRROM	562	777777,(AC)R --> (E)
HRROS	563	777777 --> (E)L
HRRE	570	[(E)18 * 777777],(E)R --> (AC)
HRREI	571	[E18 * 777777],E --> (AC)
HRREM	572	[(AC)18 * 777777],(AC)R --> (E)
HRRES	573	(E)18 * 777777 --> (E)L; If AC ≠ 0: (E) --> (AC)
HRL	504	(E)R --> (AC)L
HRLI	505	E --> (AC)L
HRLM	506	(AC)R --> (E)L
HRLS	507	(E)R --> (E)L; If AC ≠ 0: (E) --> (AC)
HRLZ	514	(E)R,0 --> (AC)
HRLI	515	E,0 --> (AC)
HRLZM	516	(AC)R,0 --> (E)
HRLZS	517	(E)R,0 --> (E); If AC ≠ 0: (E) --> (AC)
HRLO	524	(E)R,777777 --> (AC)
HRLOI	525	E,777777 --> (AC)
HRLOM	526	(AC)R,777777 --> (E)
HRLOS	527	(E)R,777777 --> (E); If AC ≠ 0: (E) --> (AC)
HRLE	534	(E)R,[(E)18 *777777] --> (AC)
HRLEI	535	E,[E18 * 777777] --> (AC)
HRLEM	536	(AC)R,[(AC)18 * 777777] --> (E)
HRLES	537	(E)R,[(E)18 * 777777] --> (E); If AC ≠ 0: (E) --> (AC)

I/O Instructions

DATAO	70014	(E) --> DATA
DATAI	70004	DATA --> (E)
CONO	70020	E --> COMMAND
CONI	70024	STATUS --> (E)
CONSZ	70030	If STATUS(R) & E = 0: skip
CONSO	70034	If STATUS(R) & E ≠ 0: skip
BLKI	70000	(E) + 1,1 --> (E); DATA --> ((E)R); If (E)L ≠ 0: skip

BLKO	70010	(E) + 1,1 --> (E); ((E)R) --> DATA; If (E)L ≠ 0: skip

Logical Testing Instructions

TLN	601	No-op
TLNE	603	If (AC)L & E = 0: skip
TLNA	605	Skip
TLNN	607	If (AC)L & E ≠ 0: skip
TLZ	621	(AC)L & ¬E --> (AC)L
TLZE	623	If (AC)L & E = 0: skip; (AC)L & ¬E --> (AC)L
TLZA	625	(AC)L & ¬E --> (AC)L; skip
TLZN	627	If (AC)L & E ≠ 0: skip; (AC)L & ¬E --> (AC)L
TLC	641	(AC)L ^! E --> (AC)L
TLCE	643	If (AC)L & E = 0: skip; (AC)L ^! E --> (AC)L
TLCA	645	(AC)L ^! E --> (AC)L; skip
TLCN	647	If (AC)L & E ≠ 0: skip; (AC)L ^! E --> (AC)L
TLO	661	(AC)L ! E --> (AC)L
TLOE	663	If (AC)L & E = 0: skip; (AC)L ! E --> (AC)L
TLOA	665	(AC)L ! E --> (AC)L; skip
TLON	667	If (AC)L & E ≠ 0: skip; (AC)L ! E --> (AC)L
TRN	600	No-op
TRNE	602	If (AC)R & E = 0: skip
TRNA	604	Skip
TRNN	606	If (AC)R & E ≠ 0: skip
TRZ	620	(AC)R & ¬E --> (AC)R
TRZE	622	If (AC)R & E = 0: skip; (AC)R & ¬E --> (AC)R
TRZA	624	(AC)R & ¬E --> (AC)R; skip
TRZN	626	If (AC)R & E ≠ 0: skip; (AC)R & ¬E --> (AC)R
TRC	640	(AC)R ^! E --> (AC)R
TRCE	642	If (AC)R & E = 0: skip; (AC)R ^! E --> (AC)R
TRCA	644	(AC)R ^! E --> (AC)R; skip
TRCN	646	If (AC)R & E ≠ 0: skip; (AC)R ^! E --> (AC)R
TRO	660	(AC)R ! E --> (AC)R
TROE	662	If (AC)R & E = 0: skip; (AC)R ! E --> (AC)R
TROA	664	(AC)R ! E --> (AC)R; skip
TRON	666	If (AC)R & E ≠ 0: skip; (AC)R ! E --> (AC)R
TDN	610	No-op
TDNE	612	If (AC) & (E) = 0: skip
TDNA	614	Skip
TDNN	616	If (AC) & (E) ≠ 0: skip
TDZ	630	(AC) & ¬(E) --> (AC)
TDZE	632	If (AC) & (E) = 0: skip; (AC) & ¬(E) --> (AC)
TDZA	634	(AC) & ¬(E) --> (AC); skip
TDZN	636	If (AC) & (E) ≠ 0: skip; (AC) & ¬(E) --> (AC)
TDC	650	(AC) ^! (E) --> (AC)
TDCE	652	If (AC) & (E) = 0: skip; (AC) ^! (E) --> (AC)
TDCA	654	(AC) ^! (E) --> (AC); skip
TDCN	656	If (AC) & (E) ≠ 0: skip; (AC) ^! (E) --> (AC)
TDO	670	(AC) ! (E) --> (AC)
TDOE	672	If (AC) & (E) = 0: skip; (AC) ! (E) --> (AC)

TDOA	674	(AC) ! (E) --> (AC); skip
TDON	676	If (AC) & (E) ≠ 0: skip; (AC) ! (E) --> (AC)
TSN	611	No-op
TSNE	613	If (AC) & (E)S = 0: skip
TSNA	615	Skip
TSNN	617	If (AC) & (E)S ≠ 0: skip
TSZ	631	(AC) & ¬(E)S --> (AC)
TSZE	633	If (AC) & (E)S = 0: skip; (AC) & ¬(E)S --> (AC)
TSZA	635	(AC) & ¬(E)S --> (AC); skip
TSZN	637	If (AC) & (E)S ≠ 0: skip; (AC) & ¬(E)S --> (AC)
TSC	651	(AC) ^! (E)S --> (AC)
TSCE	653	If (AC) & (E)S = 0: skip; (AC) ^! (E)S --> (AC)
TSCA	655	(AC) ^! (E)S --> (AC); skip
TSCN	657	If (AC) & (E)S ≠ 0: skip; (AC) ^! (E)S --> (AC)
TSO	671	(AC) ! (E)S --> (AC)
TSOE	673	If (AC) & (E)S = 0: skip; (AC) ! (E)S --> (AC)
TSOA	675	(AC) ! (E)S --> (AC); skip
TSON	677	If (AC) & (E)S ≠ 0: skip; (AC) ! (E)S --> (AC)

Program-control Instructions

XCT	256	Execute (E)
JFFO	243	If (AC) = 0: 0 --> (AC+1) If (AC) ≠ 0: E --> (PC)
JFCL	255	If AC & FLAGS ≠ 0: E --> (PC); ¬AC & FLAGS --> FLAGS
JRST	25400	E --> (PC)
PORTAL	25404	0 --> PUBLIC; E --> (PC)
JRSTF	25410	(X)L or (Y)L --> FLAGS; E --> (PC)
HALT	25420	E --> (PC); stop
XJRSTF	25424	(E)L --> FLAGS; (E+1) --> (PC)
XJEN	25430	Dismiss PI; (E)L --> FLAGS; (E+1) --> (PC)
XPCW	25434	FLAGS,0 --> (E); PC+1 --> (E+1); (E+2)L --> FLAGS; (E+3) --> (PC)
JEN	25450	Dismiss PI; (X)L or (Y)L --> FLAGS; E --> (PC)
SFM	25460	FLAGS,0 --> (E)
JSR	264	If PC(L) = 0: FLAGS,PC(R)+1 --> (E); E+1 --> (PC) If PC(L) ≠ 0: PC+1 --> (E); E+1 --> (PC)
JSP	265	If PC(L) = 0: FLAGS,PC(R)+1 --> (AC); E --> (PC)

```
                If PC(L) ≠ 0: PC+1 --> (AC); E --> (PC)

JSA     266     (AC) --> (E); E(R),PC(R)+1 --> (AC);
                  E+1 --> (PC)
JRA     267     ((AC)L) --> (AC); E --> (PC)
MAP     257     PHYSICAL MAP DATA --> (AC)
```

Shift And Rotate Instructions

```
ASH     240     (AC) * 2**E --> (AC)
ROT     241     Rotate (AC) E places
LSH     242     Shift (AC) E places
ASHC    244     (AC,AC+1) * 2**E --> (AC,AC+1)
ROTC    245     Rotate (AC,AC+1) E places
LSHC    246     Shift (AC,AC+1) E places
```

Stack Instructions

```
PUSH    261     If PC(L) = 0 or (AC)0,6-17 <= 0:
                  (AC) + 1,1 --> (AC); (E) --> ((AC)R)
                If PC(L) ≠ 0 and (AC)0,6-17 > 0:
                  (AC) + 1 --> (AC); (E) --> ((AC))

POP     262     If PC(L) = 0 or (AC)0,6-17 <= 0:
                  ((AC)R) --> (E); (AC) - 1,1 --> (AC)
                If PC(L) ≠ 0 and (AC)0,6-17 > 0:
                  ((AC)) --> (E); (AC) - 1 --> (AC)

PUSHJ   260     If PC(L) = 0: (AC) + 1,1 --> (AC);
                  FLAGS,PC+1 --> ((AC)R)
                If PC(L) ≠ 0 and (AC)0,6-17 <= 0:
                  (AC) + 1,1 --> (AC); PC+1 --> ((AC)R)
                If PC(L) ≠ 0 and (AC)0,6-17 > 0:
                  (AC) + 1 --> (AC); PC+1 --> ((AC))
                E --> PC

POPJ    263     If PC(L) = 0: ((AC)R)R --> (PC);
                  (AC) - 1,1 --> (AC)
                If PC(L) ≠ 0 and (AC)0,6-17 <= 0:
                  ((AC)R) --> (PC); (AC) - 1,1 --> (AC)
                If PC(L) ≠ 0 and (AC)0,6-17 > 0:
                  ((AC)) --> (PC); (AC) - 1 --> (AC)

ADJSP   105     If PC(L) = 0 or (AC)0,6-17 <= 0:
                  (AC) + [+-]E(R),E(R) --> (AC)
                If PC(L) ≠ 0 and (AC)0,6-17 > 0:
                  (AC) + [+-]E(R) --> (AC)
```

MACRO-20 PSEUDO-OPS

Pseudo-op/Arguments	Function
ARRAY addr[expr]	Reserves a block of storage with length expr at address addr
ASCII dtextd	Enters ASCII text; d is any delimiter not in text
ASCIZ dtextd	Enters ASCII text with guaranteed trailing null; d is any delimiter not in text
.ASSIGN sym1,sym2,increment	Assigns value of sym2 to sym1 and adds increment to sym2
ASUPPRESS	Causes all local or INTERNAL symbols not referenced after ASUPPRESS to be deleted from symbol table
BLOCK expr	Reserves a block of length expression
BYTE (n)expr	Stores value of expresson in n-bit bytes
COMMENT dtextd	Treats text as comment; d is any delimiter not in text
.COMMON symbol[expr]	Defines FORTRAN or FORTRAN-compatible COMMON block
.CREF	Resumes output of suspended cross-referencing
DEC expr,...,expr	Defines local radix as decimal
DEFINE macro(dummyarg)<macrotext>	Defines macro macro
DEPHASE	Suspends effect of PHASE pseudo-op
.DIRECTIVE directive,...,directive	Sets switches to enable/disable MACRO features .ITABM Include spaces and tabs in passed arguments .XTABM Strip leading/trailing spaces and tabs from passed arguments MACMPD Match paired delimiters in MACRO call LITLST List binary code for in-line literals

FLBLST	List binary code for 1st line of multi-line text
.OKOVL	Allow arithmetic overflow
.EROVL	Disallow arithmetic overflow
MACPRF	Prefer MACRO symbol definition over other definitions
SFCOND	Suppress source listing for failing conditional assembly
.NOBIN	Suppress binary generation
KA10	Enter KA10 as CPU type in binary file header block
KI10	Enter KI10 as CPU type in binary file header block
KL10	Enter KL10 as CPU type in binary file header block

END expr	Ends a MACRO program that starts at expresson
.ENDPS	Suspends use of relocation counter associated with current PSECT
ENTRY symbol,...,symbol	Defines each symbol as INTERNAL
EXP expr,...,expr	Enters value of each expr in a fullword in current radix
EXTERN symbol,...,symbol	Identifies each symbol as EXTERNAL to current program
.HWFRMT	Causes binary code to be listed in halfword format
.IF expr,qualifier,<code>	Specifies criteria for conditional assembly (IF expr)
.IFN expr,qualifier,<code>	Specifies criteria for conditional assembly (if NOT expr)
IFx expr,<code>	Specifies criteria and code for conditional assembly
E	Assemble if expr=0
N	Assemble if expr≠0
G	Assemble if expr>0

GE	Assemble if expr>=0
L	Assemble if expr<0
LE	Assemble if expr<=0
IF1 <code>	Assemble on Pass 1
IF2 <code>	Assemble on Pass 2
IFDEF symbol<code>	Assemble if symbol defined
IFNDEF symbol<code>	Assemble if symbol not defined
IFIDN <string1><string2>,<code>	Assemble if string1 matches string2
IFDIF <string1><string2><<code>	Assemble if string1 does not match string2
IFB <string>,<code>	Assemble if only blanks and tabs in string
IFNB <string>,<code>	Assemble if not only blanks and tabs in string
INTEGER symbol,...,symbol	Reserves storage locations at end of program at one-per-symbol
INTERN symbol,...,symbol	Declares each symbol as INTERNAL to current program
IOWD expr1,expr2	Generates I/O transfer word such that word = <2's complement (expr1)>,,<expr2-1>
IRP dummyarg,<code>	Generates expansion of code for each subargument of dummyarg
IRPC dummyarg,<code>	Generates expansion of code for each character of dummyarg
LALL	Lists all expansions (including text and macros> in program
.LINK chainnum,addr,chainaddr	Generates static chains at load time for links with same chainnum at addr and optionally places chain at chainaddr
LIST	Resumes listing following XLIST
LIT	Assembles literals beginning at current address
.LNKEND chainnum,addr	Ends a static chain for links with same chainnum at addr
LOC expr	Sets location counter to expr
.MFRMT	Causes multi-format listing of binary code
MLOFF	Terminates literals at EOL even if no closing bracket (])

MLON	Suspends MLOFF
.NODDT symbol,...,symbol	Suppresses DDT recognition of symbol
NOSYM	Suppress listing of symbol table in listing file
OCT expr,...,expr	Defines radix of expr as octal
OPDEF symbol[expr]	Defines symbol as equivalent to expr
.ORG addr	Sets location counter to addr
PAGE	Causes assembler to skip to top of next lsting page
PASS2	Switches assembler to Pass 2 processing of remaining code
PHASE addr	Assembles part of program so that it can be moved to other location for execution
POINT bytesize,addr,bitplace	Generates a byte pointer for machine byte instructions
PRGEND	Replaces END for all but last program in multi-program assembly
PRINTX text	Causes text to be output during assembly to TTY and/or listing device
.PSECT name/attribute,origin	Specifies relocation counter for code following
PURGE symbol,...,symbol	Deletes symbol from symbol table
RADIX n	Sets radix to value of n
RADIX50 code,symbol	Packs symbol into B4-35 of storage word, with code in B0-3
RELOC expr	Sets location counter to value of expr and assigns relocatable addresses to code following
REMARK text	Marks text as comment
REPEAT n,<code>	Generates code n times
.REQUEST filespec	Causes file filespec to be loaded to satisfy a global request
.REQUIRE filespec	Causes file filespec to be loaded automatically
SALL	Causes suppression of all macro and REPEAT expansions
SEARCH tabnam(filespec)	Defines list of symbol tables to be searched

SIXBIT dtextd	Enters string text in 6-bit format; d is any delimiter not in text
SQUOZE code,symbol	Same as RADIX50
STOPI	Ends IRP or IRPC before all subarguments or characters are used
SUBTTL subtitle	Defines subtitle (80. chars max) to be printed at top of each listing page
SUPPRESS symbol,...,symbol	Turns on suppress bit for symbol in symbol table; turned off when symbol referenced
SYN sym1,sym2	Defines sym2 as synonymous with sym1
TAPE	Causes assembler to begin assembling next source file in MACRO command string
.TEXT dtextd	Generates ASCIZ REL block for LINK and inserts text directly into .REL file; d is any delimiter not in text
TITLE title	Names program title and causes title to be printed on each page of listing
TWOSEG addr	Directs MACRO to assemble two-segment program with HISEG beginning at addr
UNIVERSAL tabnam	Declares symbol table of current program as available to other programs and stores tabnam in MACRO's internal UNIVERSAL table
VAR	Causes variable symbols defined with symbol#, ARRAY, or INTEGER to be assembled as BLOCK statements
XALL	Resumes standard listing after LALL or SALL
.XCREF symbol,...,symbol	Suspends output of cross-referencing for symbol
XLIST	Suspends output to program listing file for Pass 2 until next LIST
XPUNGE	Deletes all local symbols during Pass 2
XWD word1,word2	Enters low-order 18 bits of each word into a

	single storage word; high-order bits are ignored
Z ac,addr	Generates instruction word with 0 in opcode field (B0-8), ac in accumulator field (B9-12), and addr in address field (B18-35)

TOPS–20
Monitor Calls
Quick Reference Guide
AV–P173A–TM

READER'S COMMENTS

NOTE: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. If you require a written reply and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement.

Did you find errors in this manual? If so, specify the error and the page number.

Please indicate the type of reader that you most nearly represent.

- ☐ Assembly language programmer
- ☐ Higher-level language programmer
- ☐ Occasional programmer (experienced)
- ☐ User with little programming experience
- ☐ Student programmer
- ☐ Other (please specify) _____

Name _____ Date _____

Organization _____ Telephone _____

Street _____

City _____ State _____ Zip Code _____
or Country

Do Not Tear – Fold Here and Tape

digital

No Postage
Necessary
if Mailed in the
United States

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

SOFTWARE PUBLICATIONS
200 FOREST STREET MRO1-2/L12
MARLBOROUGH, MA 01752

Do Not Tear – Fold Here and Tape

Cut Along Dotted Line