



ZEUS



© Copyright
1983

**Cosmopolitan
Electronics
Corporation**

5700 Plymouth Road
Ann Arbor, MI 48105

INTRODUCTION

Cosmopolitan Electronics Corporation distributes all software on an "as-is" basis without warranty. Cosmopolitan Electronics Corporation shall not be liable or responsible to the purchaser with respect to liability, loss, or damage caused or alleged to be caused directly or indirectly by the use of this software, which includes but is not limited to any interruption of service, loss of business, anticipatory profits, or consequential damage resulting from the operation or use of this software.

ZEUS is copyrighted with all rights reserved. Copying, duplicating, selling, or any unauthorized distributing of this product is expressly forbidden. In accepting this product, the purchaser recognizes and accepts this agreement.

Throughout this manual references are made to trademarked products. The (tm) symbol is used once here to serve throughout the manual.

DOSPLUS (tm)	Micro Systems Software.
LDOS (tm)	Logical Systems, Inc.
NEWDOS/80 (tm)	Apparat, Inc.
TRSDOS (tm)	Radio Shack, a division of Tandy Corp.
Z80 (tm)	Zilog, Inc.

ZEUS

Copyright (c) 1983 by Cosmopolitan Electronics Corporation.

ZEUS

Copyright (c) 1983 by Cosmopolitan Electronics Corporation.

This Manual

Copyright (c) 1983 by Cosmopolitan Electronics Corporation.

ACKNOWLEDGEMENT

We would like to thank Rowland Archer, Johnn Audritsh, Mark Feldman, Andrew Gariepy, Phil Manfield, Mark Robinson, and Dave Welsh for taking the time to beta test and provide constructive input for ZEUS.

INTRODUCTION

ZEUS COMMANDS

A Assemble.
B Calculator.
C Global change.
D Delete line(s).
E Edit line.
F Find label.
G LPrint format.
H LPrint line (s).
I Insert.
J LPrint raw data.
K Kill file.
L Load file.
M Move/Duplicate line(s).
N Reset buffer.
O Object/Operand reference.
P Print line (s).
Q Exit.
R Reference.
S Save file.
T Print label table.
U Usage.
X Recovery on reentry.

Print, as used in the following text, means output to the device in the video DCB (401DH).

Lprint, as used in the following text, means output to the device in the printer DCB (4025H).

INTRODUCTION

The "ZEUS" files on the *ZEUS* diskette are:

ZEUS/CMD - Editor/Assembler.

ZCON/CMD - DOS configuration program and default forms.

Z80E/ASM - Source code of 794 Z80 instructions in numeric order.

ZEUS is executed by entering:

```
ZEUS
```

At the DOS command level.

ZEUS/CMD will work "as is" on DBLDOS, DOSPLUS, MULTIDOS, NEWDOS, ULTRADOS, Z'DOS, and Model III TRSDOS version 1.3. However, ZEUS/CMD must be modified to work with Model I TRSDOS, Model III TRSDOS 1.0, 1.1, 1.2, VTOS and LDOS.

The ZEUS diskette is a read only, 35 cylinder, single sided diskette, with the directory on cylinder/lump 17. (The DOS is described in Section 4).

A sophisticated DOS (DOSPLUS, LDOS, MULTIDOS, NEWDOS/80, & Z'DOS) can be configured to read the ZEUS diskette directly. However, if Model III TRSDOS is used, then "CONVERT" the files from the ZEUS diskette to the Model III TRSDOS diskette.

If a sophisticated DOS will be used in drive zero, copy the three files to this DOS.

Model I users of TRSDOS, NEWDOS/2.1, VTOS, ULTRADOS, and DBLDOS will be confronted with the "address mark" syndrom. To avoid grinding away at the ZEUS diskette ten times, boot the ZEUS diskette in drive 0 and place the ancient DOS into drive one. Use ZEUS copy function, "\$", to copy the files using the following syntax:

With two drives

```
$ ZEUS/CMD 1
```

or with one drive

```
$ $ZEUS/CMD {swap as directed. NOT every prompt requires a swap!}
```

Use the appropriate procedure to copy the other two files to the ancient DOS. After all three files have been copied, the user must reset the directory address marks on the ancient DOS by keying in:

```
SETDIR :d (0) {d = drive ancient DOS will be in}
```

ZEUS is now on the DOS to be used in drive zero.

INTRODUCTION

ZEUS has its own forms to explicitly control the printout of source text. ZEUS forms must be active to utilize all of the assemblers printout functions.

To load the forms code, enter

```
ZEUS F
```

At the DOS comand level. Or configure ZEUS/CMD to automatically load the forms code.

The configuration program, ZCON/CMD, will modify ZEUS/CMD to execute properly with all of the popular operating systems (DOS), alter the default ZEUS forms parameters, and configure ZEUS/CMD to automatically load the ZEUS forms code upon entry.

ZCON/CMD is accessed by entering:

```
ZCON
```

In response to the DOS command prompt.

ZCON/CMD is menu driven, and self explanatory.

The default condition of ZEUS is:

DOS selection = 2.

FORMS parameters

printed lines per page = 60.

number of charaacters per line = 80

total number of printed lines per page = 66

auto forms = N

INTRODUCTION

FILENAMES

ZEUS will automatically append "/ASM" to all user prompted source filenames, and "/CMD" to all object code filenames which do not include an extension. Since, the GET instruction is not user prompted, an extension, if part of the GET file, must be in the filespec of the GET instruction. To access a user prompted file without an extension, append a "/" immediately behind the filename.

EXAMPLES:

User input	Filespec given to DOS.
DOGGY	DOGGY/ASM
KOOL.MAN	KOOL/ASM.MAN
COLUMNS:2	COLUMNS/ASM:2
TABLES.MONKEY:0	TABLES/ASM.MONKEY:0
MOVIE/INC	MOVIE/INC
VIDEO/SOR.CAR	VIDEO/SOR.CAR
KILLER/GET:2	KILLER/GET:2
OVERMAKE/SRC.FUNNY:1	OVERMAKE/SRC.FUNNY:1
SEXY/	SEXY
MUCHO/.GIRL	MUCHO.GIRL
WOMAN/:3	WOMAN:3
COLDHEAT/.SUN:3	COLDHEAT.SUN:3

ZEUS I/O

Throughout this manual, the words `print` and `Lprint` are used to designate ZEUS output.

`Print` is used whenever output is directed to the device in the video DCB at 401DH.

`Lprint` is used whenever output is directed to the device in the printer DCB at 4025H.

ZEUS uses the *KI device at 401EH for all keyboard input, but scans the 3800H page of RAM to implement PAUSE, control keys, and to distinguish a shift "ENTER" from a non-shift "ENTER".

Disk I/O is performed by using the ROM calls at 0013H and 001BH for byte I/O, and the DOS vectors at 4436H and 4439H for sector I/O. ZEUS uses 256 as the logical record length for all files.

INTRODUCTION

GLOSSARY

ASCII format - Alphanumeric source code stored without line numbers.

Current line - The last printed text buffer line.

EDTASN format - Source code stored with a header byte, X'D3', followed by a six character name, alphanumeric text with each line preceded by a five character line number with the high bit set, and a terminator byte, X'1A'.

LABEL - A string used to symbolically represent a value.

Mnemonic - An assembly language instruction.

Object code - Binary code directly executable by the microprocessor.

Operator - A single symbol describing an operation to be performed.

Pseudo-op - Special orders given by the programmer to the assembler, which do not generate object code.

Relative address - The value for the beginning of the current instruction.

Source code - The symbolic representation of LABELS and mnemonics, which the assembler translates into specific object code.

Text buffer - The RAM resident storage area for source code.

REFERENCE MANUAL NOTATION

< > Carats enclose a key to be pressed. The "<" and ">" are not keyed in.
i.e. <ENTER> = Press the "ENTER" key.
i.e. <BREAK> = Press the "BREAK" key.
i.e. <N> = Press the "N" key.

{ } Braces enclose manual comments. The braces "{and the enclosed comment}" are not keyed in.
i.e. {will be shown...}

[] Brackets enclose optional parameters. The brackets "[" and "]" are not keyed in.
i.e. [H]

... Ellipses represent the repetition of an optional parameter,
i.e. [,EXP][,EXP][,...]

RELNUM RELNUM is a parameter which may be 1) a number which represents a relative line in the text buffer, or 2) a LABEL which indirectly represents a relative line in the text buffer where LABEL is in the label field.

The pound sign is a RELNUM parameter for line #1.

* The asterisk is a RELNUM parameter for the highest line.

• The period is a RELNUM parameter for the current line.

< Less than modifier for RELNUM. (One number less)
i.e. <354. {parameter = 353}

ZEUS SYNTAX

ZEUS FUNCTIONAL COMMANDS

SINGLE KEYSTROKE COMMANDS

Single keystroke commands, must be the first keystroke after the prompt, ">" is printed.

<UP-ARROW>	Print next lower line, if exists, otherwise print line #1 (first line).
<DOWN-ARROW>	Print next higher line, if exists, otherwise null.
<ENTER>	Print page from current line.
<SHIFT><ENTER>	Print page to current line, first press. Subsequent presses, print page to current line less 14.
<SHIFT><UP-ARROW>	Print line #1 (first line).
<SHIFT><RIGHT-ARROW>	Print the highest line.
. {period}	Print current line.

PAUSE COMMAND

<SPACE>	The <SPACE> will pause printing. After a pause, single step with <SPACE>, continue with <ENTER>, or abort with <BREAK>•
---------	---

CURSOR MOVES

<RIGHT-ARROW>	Move the cursor to the next TAB position.
<SHIFT><SPACE>	Move the cursor to the next TAB position.
<LEFT-ARROW>	Backspace one position and remove character at the new cursor position.
<SHIFT><LEFT-ARROW>	Erase all characters previously keyed in, and position cursor to the first input position.

INTERPRET/ABORT

The "ENTER" key is used to signify completion of an user input command. All commands are terminated with <ENTER>.

The "BREAK" key will abort any user command, and return the user to the command prompt ">".

ZEUS SYNTAX

In order for the assembler to translate source code into object code, a specific format is required when entering text. Fields are used to distinguish the various zones in a source text line for the symbolic representation of a user's program, and user comments.

The LABEL field, optionally contains a label with .a value equivalent to the relative address of, or by, the instruction which follows.

The OPCODE field contains a Z80 opcode, or a ZEUS psuedo-op.

The OPERAND field contains operands. An operand is one or two groups of characters separated by a comma.

The INSTRUCTION field contains a opcode and operand consecution - mnemonic, which will be translated into object code by the assembler.

The COMMENT field, optional, contains user remarks.

SOURCE TEXT LINE:

```
<----- FIELDS----->
<----- SYMBOLIC CODE----->
<---- INSTRUCTION----->
```

```
LINE# [LABEL] OPCODE [OPERAND(S)]           [;COMMENT]
```

LINE NUMBERS - The line numbers printed, indicate the relative line in the text buffer.

LABELS - Labels are a string of characters with every character and character case significant. To distinguish a label from a numeric constant, the first character in a label must be a @, A thru Z, [, \,], ^, _, ', a thru z, {, |, }, and ~. The balance of the label can also have 0 thru 9, :, =, >, or ?.

COMMENT - A comment begins with a semicolon, and overrides any field.

During assembly printing, two additional zones are added to the left of the line numbers.

```
<----- FIELDS----->
<----- SYMBOLIC CODE----->
<---- INSTRUCTION----->
```

```
ADDR OBJCODE LINE# [LABEL] OPCODE [OPERAND(S)]           [;COMMENT]
```

ADDR - The addr zone contains a hexadecimal number which is either 1) the relative address of the current instruction, 2) the label value for the EQU and DEFL psuedo-ops, 3) the length of the space for the DEFS psuedo-op, or 4) the transfer address for the END psuedo-op.

OBJCODE - The objcode zone contains the hexadecimal object code translated from the mnemonics.

ZEUS SYNTAX

CONSTANT - A constant is a numeric representation of a value. To distinguish a numeric constant from a label, the first character must be numeric or the constant must be enclosed in single quotes. Decimal constants use the numbers 0 thru 9, and optionally are suffixed with the letter D. Hexadecimal constants use the numbers 0 thru 9, the letters A thru F, and are suffixed with the letter H. ASCII constants use the entire ASCII set and are enclosed in single quotes. The single character dollar sign, "\$", is the value of the relative address for the current instruction in which it appears.

EXPRESSION - An expression is a group of characters consisting of two or more constants and/or labels separated by an operator. Expressions are evaluated in a left to right order, maintaining an interim result after processing a constant and/or label, ignoring algebraic hierarchy, and not recognizing parentheses as a precedence modifier.

The nine ZEUS operators are:

1. + ADDITION. Adds the value following the operator to the interim result and establishes the sum as the new interim result.
2. - SUBTRACTION. Subtracts the value following the operator from the interim result and establishes the difference as the new interim result.
3. * MULTIPLICATION. Multiplies the value following the operator to the interim result and establishes the product as the new interim result.
4. / DIVISION. Divides the value following the operator into the interim result and establishes the integer portion of the quotient as the new interim result.
5. ! Logical OR. Logical OR between the value following the operator and the interim result establishing a new interim result.
6. # Logical XOR. Logical XOR between the value following the operator and the interim result establishing a new interim result.
7. & Logical AND. Logical AND between the value following the operator and the interim result establishing a new interim result.
8. % MODULO. Divides the value following the operator into the interim result and establishes the integer remainder as the new interim result.
9. < SHIFT. Shifts the interim result the number of bits of the value following the operator establishing a new interim result. If the value following the operator is positive, then the shift is left, otherwise the shift is right.

ZEUS SYNTAX

PSUEDO-OPS

1. COMM - Comment.
2. DEFB or DB-Define byte(s).
3. DEFL or DL-Define label.
4. DEFM or DM-Define string(s) and/or byte(s). (define message)
5. DEFS or DS-Define space.
6. DEFW or DW-Define word(s).
7. END - Terminate assembly.
8. ENIF - Delimit conditional assembly.
9. ERR - Abort assembly.
10. EQU - Set a label to a value. (equate)
11. GET - Include source from disk file.
12. IF - Begin conditional assembly.
13. LIST - Control printing.
14. ORG - Establish relative address. (origin)
15. PAGE - Lprint linefeeds to top of form.
16. SBTL - Establish a subtitle for Lprint.
17. TITL - Establish the title for Lprint.

Pseudo-ops are entered in the instruction field of a source text line.

ZEUS SYNTAX

ZEUS PSUEDO-OP DETAIL

COMM - Generate an object file comment block at the beginning of the file. COMM is primarily used to put a message on the first sector of a object file

NOTE: A COMM instruction is only recognized if no object code has been generated.

Syntax:

```
00017          COMM 'STRING
```

STRING is a string of characters 1 to 63 characters in length, and must be preceded by a single quote. To create a comment greater than 63 characters, add an additional line with a COMM instruction immediately behind the line with the first COMM instruction. The assembler will append these STRINGS into one comment.

EXAMPLES:

```
00003          COMM 'I      can't stop loving you.
```

Generates the comment "I can't stop loving you."

```
00008          COMM *****                               My      **      best      *
00009          COMM '* program! *****
```

Generates a comment block which, when viewed with a "ZAP" utility, would appear as:

```
.N*****
*      My      *
*      Best    *
*      Program! *
*****
```

The comment block on a disk sector is preceded with two bytes. The first byte will be a X'05', and the second byte will be the length of the comment.

The layout of this comment is as follows:

1. 14 *'S {adjust for the twobytes preceding comment.}
2. *, 6 Spaces, My, 6 spaces, *
3. *, 5 Spaces, Best, 5 spaces, *
4. *, 3 Spaces, Program!, 3 Spaces, *
5. 16 *'S

ZEUS SYNTAX

DEFB or DB - Generate a byte(s) at the relative address. DEFB is primarily used to create single byte lookup tables.

Syntax:

```
01333          DEFB BVAL[,BVAL][,BVAL][,...]
```

BVAL is a constant, expression, or label with a value of 0 to 255. For each byte generated, the relative address is incremented by one.

EXAMPLES:

```
00422          DEFB      45
00017          DEFB      46H,63H,73H,74H
00324          DEFB      78,CAT,'U'          {CAT          is definedelsewhere}
```

During assembly printing, only the first byte value is printed.

```
8923 4 E          00324          DEFB 78,CAT,'U'
```

DEFL or DL - Establishes a value for a label which only can be redefined with another DEFL psuedo-op. DEFL is primarily used to perform a complex assembly function. An example of a complex assembly function is shown at the end of ENIF.

Syntax:

```
00389 LABEL DEFL WVAL
```

WVAL is a constant, expression, or label. If WVAL is a label or an expression with a label, and the label has not been established, then LABEL will have value of zero on the first pass. If WVAL is LABEL then LABEL will be established as the value of the relative address.

EXAMPLES:

```
00120 TOPCO      DEFL      78BAH          {TOPCO = 78BAH}
00121 MAKE       DEFL      MAKE          {MAKE = relative address}
00122 KITTEN     DEFL      TOPCO-80H     {KITTEN = 783AH}
00123           DEFB      KITTEN&0FFH   {byte value * 3AH}
00124 KITTEN     DEFL      TOPCO+80H     {KITTEN = 793AH}
00125           DB        KITTEN<-8     {byte value = 79H}
```

ZEUS SYNTAX

DEFM or **DM** - Generate a string(s), and/or byte(s) at the relative address. DEFM is primarily used to insert messages into the object code.

Syntax:

```
00457          DEFM 'STRING *[,BVAL] [,'STRING'] [,...]
```

although DEFM contains DEFB in its repertoire, DEFM can generate object code from a string of ASCII characters enclosed in single quotes.

EXAMPLES:

```
00317          DEFM 'This is a message.'
00237          DEFM 'This is a message with a terminator.',13
01283          DEFM 28,31,'ERROR','!' +80H,10,13
```

During assembly printing, only the first byte value is printed.

```
AB77 1C          01283          DEFM 28,31,'ERROR','!' +80H,10,13
```

DEFS or DS - Increments the relative address by a user defined value. DEFS is primarily used to retain the contents of a block of memory. The special case, DEFS 0, is primarily used to easily mark the end of a table for reference purposes.

Syntax:

```
00017          DEFS WVAL
```

WVAL is a constant; expression with labels, if any, previously defined; or previously defined label with a value between 0 and 65535 inclusive. A DEFS psuedo-op will generate a new loading address for the object file, which when loaded, will skip over WVAL bytes of RAM. Therefore, DEFS preserves WVAL bytes of RAM starting at the relative address.

EXAMPLES:

```
00056          DEFS 256          {this will leave 256 bytes of RAM
                                untouched when the object program is
                                loaded into RAM.}

00330 PICBEG DB          'a','b','c','d','e','f'
00331 PICEND DEFS        0
... More source text
00677          LD          B,PICEND-PICBEG {loads B with the length of
                                the table, 6.}
```

ZEUS SYNTAX

DEFW or DW - Generate a word(s) at the relative address. DEFW is primarily used to create an address table.

Syntax:

```
00017          DEFW WVAL[,WVAL] [,WVAL] [,...]
```

WVAL is a constant, expression, or label with a value of 0 to 65535 inclusive. For each word generated, the relative address is incremented by two.

EXAMPLES:

```
00188          DEFW TEST          {TEST is defined elsewhere}
00376          DEFW    4488H,65123,3,12
```

During assembly printing, only the first word value is printed.

```
1228  8844          00376          DEFW 4488H,651233,3,12
```

END - Terminate assembly. END is used to direct the assembler to ignore any source text which follows. The END psuedo-op in a GET file will terminate assembly of the GET file and resume assembly from the source text which has the GET instruction.

Syntax:

```
00017          END          [TRANSFER]
```

TRANSFER is the address which the DOS program execution routine will transfer control (hopefully) over to the user program. This address has also been designated as the program execution point. TRANSFER can be a constant, expression (unusual), or a previously defined label.

EXAMPLES:

```
03122          END          {transfer address 0000H}
02883          END    402DH    {transfer address402DH}

00001          ORG    7000H
00002  START LD          A,(HL)    {START = 7000H}
... More source text
02893          RET
02894          END    START    {transfer address7000H}
```

ZEUS SYNTAX

ENIF and IF - Conditional assembly psuedo-ops.
Conditional assembly provides the programmer with an efficient tool to create programs for a variety of conditions. Conditional assembly source text is structured as follows:

```
00017          IF      EXP
      ... Conditional source text
00043          ENIF
```

If EXP is evaluated to a non zero value, then the conditional source text is assembled. If exp is evaluated to a zero value, then the conditional source text is not assembled,

or

```
00017          IF      NOT,EXP
      ... Conditional source text
00043          ENIF
```

If EXP is evaluated to a zero value, then the conditional source text is assembled. If exp is evaluated to a non zero value, then the conditional source text is not assembled.

During an assembly printing, the IF and ENIF text lines, regardless of the condition, are not printed. If the conditional source code is not assembled, then the lines which contain the conditional source text are also not printed.

Typically, the programmer will incorporate at the beginning of the source text, the status for conditional assembly

```
00002 MOD1 DEFL      1
00003 MOD3  DEFL      1-MOD1      {MOD3 = 0}
```

Then have subsequent code test this status

```
00789          IF      MOD1
00790          LD      A, (37ECH)
00791          ENIF
00792          IF      MOD3
00793          IN      A, (0F0H)
00794          ENIF
```

EXAMPLE:

```
00001          ORG      7000H
00002 DOG      DEFL      DOG-$
00003          IF      NOT,DOG
00004          GET      EQU/ASM
00005          ENIF
```

The GET instruction on line 00003 will only be executed on the first pass. The file EQU/ASM must only have EQU'sl.

ZEUS SYNTAX

ERR - Abort assembly and display message. The ERR psuedo-op is usually in a conditional block of source text, set to assemble if an undesirable situation exists. ERR may also be used to quickly get the relative address in a program.

Syntax:
00017 ERR 'STRING

If the line containing ERR is assembled, then the relative address is printed followed by STRING. STRING must be preceded by a single quote.

EXAMPLE:

```
00001      ORG      7000H
... More source text
01782      RET
01783      ERR      'is the first free byte.
01784      END      START
```

If line 01782 would assemble to relative address 8A21H, then an assembly would produce

8A22 is the first free byte.

during the first pass.

EQU - Explicitly establish the value for a label. The EQU psuedo-op is primarily used to symbolically make references to constants.

Syntax:
00017 LABEL EQU WVAL

WVAL is a constant; expression with labels, if any, previously defined; or previously defined label with a value between 0 and 65535 inclusive.

EXAMPLE:

```
00002 KEYBRD EQU      2BH
00003 VIDEO  EQU      33H
00004 PRINT  EQU      3BH
00005 LINEIN EQU      40H
00006 KBWAIT EQU      49H
... More source text
00345 VLINE  LD        A, (HL)
00346        CALL     VIDEO
00347        INC      HL
00348        CP       0DH
00349        JR       NZ, VLINE
00350        RET
```

ZEUS SYNTAX

GET - Include source text from disk file. The GET psuedo-op is used to assemble large amounts of source text, by reading the disk file as source text instead of the resident text buffer.

Syntax:

```
00017          GET FILESPEC
```

If FILESPEC is not found assembly is aborted.

An END instruction is not necessary in a GET file. If an END instruction is found in a GET file or the end of file is encountered, then assembly resumes in the source text which has the GET instruction. 592 Bytes of RAM are dynamically allocated for each concurrent opened GET file. The limit of nesting GET files is dependent on the amount of RAM space available.

During assembly printing, the text line with the GET instruction will not be printed.

LIST - Control assembly printing. The LIST psuedo-op is primarily used during program development to print only pertinent parts of the assembled code.

Syntax:

```
00017          LIST    OFF
```

LIST OFF will disable assembly printing, starting with this line,

or

Syntax:

```
00033          LIST    ON
```

LIST ON will enable assembly printing with the following line, provided the N option is not entered under the A command.

ZEUS SYNTAX

ORG - Establish a new value for relative address. The ORG pseudo-op is used to direct the assembler where the object code will be.

Syntax:

```
00001          ORG WVAL
```

WVAL is a constant; expression with labels, if any, previously defined; or previously defined label between the values of 0 to 65535 inclusive.

PAGE, SBTL, and TITL - Assembly Lprinting psuedo-ops. These psuedo-ops dress up the assembly Lprinting by placing a title and subtitle on each page. The PAGE psuedo-op generates linefeeds to the top of, the next page.

NOTE: These three psuedo-ops require ZEUS forms, and are not recognized in a GET file.

Syntax:

```
00023          TITL      'STRINGTITLE
00024          SBTL      'STRINGSUBTITLE
00876          PAGE
```

TITL is used to place a title, STRING TITLE, on the first line on each page of an assembly Lprinting. Only the first title is Lprinted, subsequent titles are ignored. The subtitle, SBTL STRING, if exists, is Lprinted on the line after title. New subtitles may be inserted in the source text, but will only appear at the next page break.

EXAMPLE:

```
00023          SBTL      'Screen formatting routines.
... More source text
00317          SBTL      'The DOG is MAN's best friend.
00318          PAGE
```

The original subtitle, "Sreen formatting routines.", will be printed immediately after each title, if exists. When the assembler processes lines 00317 and 00318, PAGE will generate linefeeds to the next top of form, then establish the subtitle, "The DOG is MAN's best friend.", as the new subtitle, printing this subtitle on the new page.

During assembly printing, text lines with one of these three psuedo-ops will not be printed.

ZEUS COMMAND DETAIL

ZEUS COMMANDS

A Assemble.
B Calculator.
C Global change.
D Delete line(s).
E Edit line.
F Find label.
G LPrint format.
H LPrint line(s).
I Insert.
J LPrint raw data.
K Kill file.
L Load file.
M Move/Duplicate line(s).
N Reset buffer.
O Object/Operand reference
P Print line(s).
Q Exit.
R Reference.
S Save file.
T Print label table.
U Usage.
X Recovery on reentry.

Print, as used in the following text, means output to the device in the video DCB (401DH).

Lprint, as used in the following text, means output to the device in the printer DCB (4025H).

ZEUS COMMAND DETAIL

Assemble source code.

Syntax A[E][H][N][O][Q][T[nn]]<ENTER>

- E - Wait on error. The E option will call assembly to be suspended whenever an error occurs. After the error message and the offending line is printed, assembly can be resumed by pressing either <SPACE> or <ENTER>.
- H - LPrint link to Print. The H option will link the assembly printing to the lineprinter.
- N - Printing off (does not suppress label table). The N option will inhibit assembly printing for non-erroneous lines, and overrides the LIST ON instruction.
- O - Generate object deck. The O option directs the assembler to generate loading instructions as well as object code to a user selected filespec. If the O option is entered, then the prompt

Filespec:

will be printed. The assembler will automatically append /CMD to the filename if none was entered, then search for the filespec.

If the filespec is found, then the query

Overwrite file?

will be printed. Press <Y> to overwrite file, <N> to return to the filespec prompt, or <BREAK> to abort assembly.

If the filespec is not found, then the query

Create file?

will be printed. Press <Y> to create a new file, <N> to return to the filespec prompt, or <BREAK> to abort assembly.

- Q - Quit after first pass. The Q option will terminate assembly if an error is detected during the first pass. After the assembler makes one pass (builds label table), the assembler will not make this first pass again unless Q is entered as an assembly option, a disk I/O error occurs, or one of the following commands is invoked.

C, D, E, I, L, M, N, or Q.

The psuedo ops IF, DEFS, DS, EQU, ORG, and END will generate first pass and second pass errors if the corresponding expressions cannot be evaluated (undefined label, or expression error). The totalerrors will include the first pass as well as the second pass errors, even if the same error is indicated on both passes. If the A command is executed again, and the first pass is not processed, then the total errors will be the results of the second pass only. A repeated label in the label field, redefinition error, will result in first pass errors only.

ZEUS COMMAND DETAIL

T - Print label table. The T option will sort the labels in ascending alphanumeric order, then print the labels and their corresponding hexadecimal values up to nn characters wide.

NOTE: Options may be 1) in any order, 2) repeated, 3) separated by one or more spaces, 4) separated by one or more commas, 5) separated by any combination of spaces and commas, or 6) may not be separated at all.

EXAMPLE:

```
ANO {is equivalent to.}
{1} AON           {2} A,O,N           {3} A,N,O           {4} A O N
{5} A ,O ,N      {6} A , ,O , , , ,N,O,O,N N N      etc.
```

EXAMPLE:

```
A,N,H<ENTER> {print and Lprint error(s) and error
line (s), if any.}
```

EXAMPLE:

```
AT80H<ENTER> {print and Lprint source code and label
table using 80 columns for table.}
```

EXAMPLE:

```
AQE<ENTER> {pause on error line(s), if any, and
terminate assembly after the first pass
if any errors occurred.}
```

EXAMPLE:

```
A N<ENTER> {print error(s) and error line(s), if
any, then print the total errors for the
processed source code.}
```

The assembler operates in three phases. During the first pass/phase the assembler creates the label table, and establishes the values for DEFS, DS, END, EQU, and ORG. If there are no first pass errors, or Q option not entered, the assembler sets a first phase complete status byte and continues with the second pass/phase. During the second pass/phase the assembler will output in accordance to the selected options, then print the total errors. After the second phase is complete, the assembler sets a second phase complete status byte. Additional A commands with the second phase status byte set, will proceed with phase three/second pass. Phase three is instant assembly on all processed source code in the text buffer.

This characteristic is unique with **ZEUS**, and encourages the user to assemble with "AN", to check for assembly errors, then proceed with "ANO" for the generation of an object deck, if there are no errors. The maximum speed is obtained if the user has maximized the amount of code in the resident text buffer.

ZEUS COMMAND DETAIL

B - Command mode calculator.

Syntax Bexpl[operator exp2][operator exp3][...]<ENTER>

Operator can be:

+ ADD.
- SUBTRACT (also unary) •
* MULTIPLY.
/ DIVIDE.
! OR.
XOR.
& AND.
% MODULO.
< SHIFT LEFT.
<- SHIFT RIGHT.

The B command will calculate expressions in the range of 0 to 65535 decimal and/or 0000H to 0FFFFH hexadecimal, returning an integer result in the range of 00000 to 65535 decimal and 0000H to FFFFH hexadecimal.

EXAMPLES: {The <ENTER> is left out for clarity.}

B 38H 00056 0038H	B 99 00099 0063H	B 8C9EH 35998 8C9EH
B 56+5 00061 003DH	B 56H+5 00091 005BH	B 0A0H+23H 00195 00C3H
B 37-3 00034 0022H	B 500-45H 00431 01AFH	B 0FFH-255 00000 0000H
B 67D*33 02211 08A3H	B 0AH*0AH 00100 0064H	B 233*56H 20038 4E46H
B 78/9 00008 0008H	B 0FE9H/23H 00116 0074H	B 3C00H/4 03840 0F00H
B 5!25 00029 001DH	B 37H123 00055 0037H	B 0F132H!89DDH 63999 F9FFH
B 0F23H#1A44H 05479 1567H	B 89#19 00074 004AH	B 99D#78H 00027 001BH
B 5&-5 00001 0001H	B 23&5/3 00001 0001H	B 89!8&444/2 00012 000CH
B 89%7+23*6 00168 00A8H	B 89H%7 00004 0004H	B 12345%2 00001 0001H
B 89<1 00178 00B2H	B 0FFFFH<8 65280 FF00H	B 0FFFFH<16 00000 0000H
B 89<-1 00044 002CH	B 0FFFFH<-8 00255 00FFH	B 566H/23<-3 00007 0007H

ZEUS COMMAND DETAIL

If the assembler has processed phase one, and the label table is intact, then the labels created by the assembler can be used as expressions.

EXAMPLES:

```
B WORD-BYTE                               {from the file Z80E/ASM).
01548 060CH
```

```
B OFFSET*3
00090 005AH
```

```
B DISPLACEMENT<8/BYTE+WORD
01797 0705H
```

Calculations are performed on a left to right basis. If an expression is missing or an operator is used improperly, then

Expression?

is printed, and the B command terminates. If a label is used as an expression, and this label is not found in the label table, then

Undefined labell

is printed, and the B command terminates.

EXAMPLES:

```
B 75-33*
Expression?
```

```
B UNCLE+3
Undefined label!
```

If the resulting calculation exceeds 65535 decimal, FFFFH hexadecimal, then the results will be a modulo 65536 answer.

EXAMPLES:

```
B300*512           {300*512=153600. 153600%65536=22528.}
22528 5800H
```

```
B65200+500
00164 00A4H
```

ZEUS COMMAND DETAIL

C - Global change.

Syntax C[OPT]STR1, [STR2] [,RELNUM1] [,RELNUM2]<ENTER>

The C command will change STR1 to STR2 from RELNUM1 to RELNUM2 inclusive. Single quoted text and comments are untouched, unless OPT is used as specified below. IF STR2 is null, the STR1 will be deleted.

OPT can either be a single quote, or a semicolon. If OPT is a single quote, then changes will only take place in single quoted text. If OPT is a semicolon, then changes will only take place in comment text.

If a change in a given line would result in the line exceeding 127 characters in length, then the message

Out of Memory!

will be printed and the C command aborted without changing the line.

The C command will print the line before the change occurs and again after the change has successfully taken place. The <SPACE> may be used to pause the C command, and the <BREAK> key to terminate the C command.

If RELNUM1 is not specified, then line one is used. If RELNUM2 is not specified and RELNUM1 is specified, then RELNUM2 will default to RELNUM1 (single line change). If neither is specified, then the changes will take place over the entire text.

EXAMPLES:

C'aborted,terminated	{change "aborted" to "terminated", in single quoted text, throughout the text buffer}
C DOG,,4,37	{delete "DOG", from line 4 to 37 inclusive}
C CAT,	{delete "CAT" throughout the text buffer}
C MAN,BOY,..*	{change "MAN*" to "BOY", from the current line to the end of the text buffer}
C ABC , DEF , LOG ,<PERK	{change "ABC" to "DEF", from the line which has the label "LOG" in the label field, to one line less than the line which has "PERK" in the label field}

NOTE: The C command processes each changed line through the syntax check routine. If a change is made which would result in an erroneous line, the E command will be invoked permitting corrections to the offending line. Although the C command continues after the line is deemed acceptable, the outcome may not be the desired results. The C command can be terminated immediately after the "force" edit, by pressing the <ENTER> and <BREAK> keys simultaneously. The C command can be directed to bypass the syntax checking by using an explanation mark for OPT. (User beware!)

ZEUS COMMAND DETAIL

D - Delete specific line(s) from the text buffer.

Syntax DRELNUM1, [RELNUM2]<ENTER>

The D command will remove lines from RELNUM1 to RELNUM2 inclusive. If RELNUM1 is not specified, then the current line is used. If RELNUM2 is not specified, then RELNUM2 defaults to RELNUM1. If RELNUM2 is specified, then RELNUM2 must be equal to or greater than RELNUM1.

EXAMPLE: {current line * 37}

```
D,.                {delete line37}
D                 {delete line37}
D,                {delete line37}
D.                {delete line37}
D.,               {delete line37}
D37               {delete line37}
D#, .             {delete lines 1 to 37 inclusive}
D., *             {delete lines 37 to the end of the text
                  buffer}
```

```
D43, .                         {results in}
.   <- parameter error {RELNUM2 cannot be less than
                       RELNUM1}
```

EXAMPLE:

```
D 43,22                {results in}
22 <- parameter error
```

EXAMPLE:

```
00075                RET
00076 LOOP        LD        A, {HL}
00077                INC       HL
00078                OR        A
00079                RET       Z
00080                DJNZ     LOOP
00081 DOG         LD        HL, MOKE
00082                CALL     BOTTLE
```

Lines 76 to 80 may be deleted by one of the following.

- 1) D76,80
- 2) D76,<81
- 3) D76,<DOG
- 4) D<77,80
- 5) D<77,<81
- 6) D<77,<DOG
- 7) DLOOP,80
- 8) DLOOP,<81
- 9) DLOOP,<DOG

ZEUS COMMAND DETAIL

E - Edit specific text line.

Syntax ERENUM<ENTER>

The E command will permit the user to modify the contents of a text line. If RELNUM is not specified, then the current line is used. When the E command is invoked, the line will be printed with a small transparent rectangle cursor at the first position, and a block at the end of the line. Concurrently the <DOWN-ARROW> and <CLEAR> keys become function keys, and the user will be in the overstrike mode. In the overstrike mode, any non-control key will replace the character at the cursor position.

In the insert mode, indicated by the small rectangle cursor becoming a large rectangle cursor, any non-control character will be inserted at the cursor position, and the balance of the line moved right one position. The insert mode will drop the 128th character if the number of characters in a line would exceed 127.

Control keys which do not overstrike or insert.

Edit functions:

The edit functions are implemented by pressing one of the function keys (<CLEAR> or <DOWN-ARROW>) with one appendage, and, while keeping this key held down, press one of the four keys listed below:

A	abort changes and start editing again,
D	delete character at cursor.
H	delete all characters from cursor to end of 1 line.
I	shift between overstrike and insert.

Cursor moves:

<RIGHT-ARROW>	move right 1 character position (to tab use <SHIFT><SPACE>).
<LEFT-ARROW>	move left 1 character position.
<SHIFT><RIGHT-ARROW>	move to end of line.
<SHIFT><LEFT-ARROW>	move to beginning of line.

Other control keys:

<BREAK>	abort EDIT and leave the line without any changes.
<ENTER>	process the modified line through the syntax check routine. If no error is found, replace the edited line with the modified line, and exit the E command. If an error is found, then the E command will disable <BREAK>, place the edit function in the overstrike mode, and position the cursor at the first character in the potentially erroneous symbolic field.

ZEUS COMMAND DETAIL

F - Print line with the specified label.

Syntax FLABEL<ENTER>

The F command will search from line one for LABEL in the label fields, and print the line if found. The LABEL is mandatory with the F command. If LABEL is null, then

Label not specified!

will be printed. If the LABEL does not exist in the label field, then

Label or line not found,

will be printed.

G - Set printer values.

Syntax GL,W,P<ENTER>
or
Syntax G<ENTER>

The G command sets up the parallel lineprinter characteristics for ZEUS forms. The parameters are: L = the number of printed text lines per page, W = the maximum number of characters printed before ZEUS forms will send a carriage return, and P = the number of physical lines per page.

G <ENTER> will print the current forms settings.

EXAMPLES:

G 50,80,52<ENTER> {Lprint 50 text lines, of 80 characters
 maximum on a 52 line page. (11" for
 QUICK PRINTER)}

G 80,132,88<ENTER> {Lprint 80 text lines, of 132 characters
 maximum on a 88 line page.}

G 60,80,6<ENTER> {Lprint 60 text lines, of 80 characters
 maximum on a 66 (see NOTE!) line page,
 (normal default for most parallel
 printers)}

NOTE: If P is less than L, then P is
added to L.

ZEUS COMMAND DETAIL

H - LPrint line(s).

Syntax HRELNUM1,RELNUM2<ENTER>

The H command will print and Lprint RELNUM1 to RELNUM2 inclusive. If RELNUM1 is not specified, then the current line is used. If RELNUM2 is not specified, then RELNUM2 defaults to RELNUM1. If RELNUM2 is specified, then RELNUM2 must be equal to or greater than RELNUM1.

The **H** commands **links** the lineprinter to the video monitor.

I - Insert lines.

Syntax IRELNUM<ENTER>

The I command is used to insert or add source code into the text buffer immediately behind RELNUM. If RELNUM is not specified then the current line is used. IO<ENTER> is used to insert a line ahead of the text buffer (new line one).

The I command will print, a five digit line number one unit higher than RELNUM followed by a space, and wait for the user to key in text. After a text line is keyed in and terminated with <ENTER>, the line is processed through the syntax check routine. If no error is found, then the line is inserted into the text buffer. If an error is found, the I command is suspended, <BREAK> disabled, the E command invoked, with the edit function in the overstrike mode, and the cursor positioned at the first character in the potentially erroneous symbolic field. When subsequent modifications correct the erroneous line, then the I command will resume.

The I command continues to print a line number one unit higher than the previously entered line and await user input until <BREAK>, which will return the user to the command mode.

ZEUS COMMAND DETAIL

J - Raw data to Lprint.

```
Syntax      J<ENTER>          {Software top of form.}
Syntax      Jdata<ENTER>     {will send data directly      to
printer.}
```

EXAMPLES:

```
J27,69      J 1BH,45H J 27,'E'      J 1BH,'E'
J 'Z','e','u','s',13<ENTER> {will Lprint "Zeus".}
J <ENTER>   {will determine how many software lines
              remain to top of form, and generate linefeeds
              to match.}
```

K - Kill filespec.

```
Syntax Kfilespec<ENTER> {auto "/ASM".}
```

L - Load source file.

```
Syntax      L[OPT1][filename][OPT2]<ENTER> {auto "/ASM".}
```

```
If OPT1 =#, then load a ASCII format file.
If OPT1 =*, then load a EDTASM format file.
If OPT1 =%, then load a EDTASM format file without header.
```

OPT2 is used in conjunction with OPT1 to tell ZEUS to skip over OPT2 number of lines before loading source code into the text buffer.

If text already in buffer, then

Append to buffer?

will be printed. Respond <Y>, <N>, or <BREAK>. If the filename was not entered in the command prompt, then

Filespec:

will be printed. Enter filespec or <BREAK>.

EXAMPLE:

```
L*SUPER 3000<ENTER>
```

This command directs ZEUS to load the EDTASM format file SUPER/ASM, starting with the 3001th source text line.

ZEUS COMMAND DETAIL

M - Move/duplicate text lines.

Syntax M[OPT]RELNUM1,RELNUM2,RELNUM3<ENTER>

The M command will move RELNUM1 to RELNUM2 inclusive behind RELNUM3. If RELNUM1 is not specified, then the current line is used. If RELNUM2 is not specified, then RELNUM2 defaults to RELNUM1. If RELNUM2 is specified, then RELNUM2 must be equal to or greater than RELNUM1. RELNUM3 must be greater than RELNUM2 or less than RELNUM1. If RELNUM3 is not specified then RELNUM3 defaults to RELNUM2 (error).

If OPT = 1, then the lines in the range of RELNUM1 to RELNUM2 inclusive are duplicated behind RELNUM3.

EXAMPLE:

M!,,< {duplicates current line at current line.}

N - New text buffer.

Syntax N<CLEAR><ENTER>

Resets all pointers. {Recoverable via "X"}

O - Opcode/operand reference.

Syntax O[OPT]STR1<ENTER>

The O command will search from line one for the first occurrence of STR1. STR1 can be any combination of labels, tabs, opcodes, or operands.

If OPT = #, then the O command will print all lines with STR1.

EXAMPLE:

O LD{TAB}A, (HL)<ENTER> {locates LD A, (HL).}

If STR1 is located, then subsequent O<ENTER>'s will find the next occurrence of STR1 until

Text end.

is printed, at which time the reference search will restart at line one.

If STR1 is located, then O#<ENTER> will print all remaining line numbers with STR1.

ZEUS COMMAND DETAIL

P - Print line(s)

Syntax PRELNUM1,RELNUM2<ENTER>

The P command will print text lines from RELNUM1 to RELNUM2 inclusive to the video monitor. If RELNUM1 is not specified, then the current line is used. If RELNUM2 is not specified, then RELNUM2 defaults to RELNUM1. If RELNUM2 is specified, then RELNUM2 must be equal to or greater than RELNUM1.

Q - Exit **ZEUS**.

Syntax Q<CLEAR><ENTER>

R - REFERENCE

Syntax R[OPT]STR1<ENTER>

The R command will search from line one for the first occurrence of STR1. STR1 can be ASCII text and not opcodes and/or operands. The 0 command is used to find opcodes and/or operands.

If OPT = #, then the R command will print all lines with STR1.

EXAMPLE:

R DOGGY<ENTER> {locates DOGGY.}

If STR1 is located, then subsequent R<ENTER>'s will find the next occurrence of STR1 until

Text end.

is printed, at which time the reference search will restart at line one.

If STR1 is located, then R#<ENTER> will print all **remaining** line numbers with STR1.

S - Save filespec.

Syntax S[OPT1][filename]<ENTER> {auto "/ASM"}

If OPT1 =#, then save is in ASCII format.

If OPT1 =*, then save is in EDTASM format.

If OPT1 =%, then save in EDTASM format without header.

ZEUS COMMAND DETAIL

T - Sorted label table printing.

Syntax T[nn]<ENTER>
 nn = 1 to 252

The T command will sort the label table in ascending alphanumeric order, then print its contents in the format:

```
LABEL 0000 LABEL 0000 LABEL 0000 LABEL 0000 LABEL 0000
```

nn is the width of device to receive the label table. If nn = 0 or nn is null, then the device width will default to 64.

Once an assembly has gone through the first phase, the label table is completed. However, a disk I/O error, or the completion of one of the following commands C, D, E, I, L, M, N, or Q will reset the label table.

U - Memory usage.

Syntax U<ENTER>

The U command will print the memory used by the resident source text; and, if the assembler has completed phase one, the memory used by the label table, and GET files. In addition, the amount of free memory is printed.

EXAMPLE: {File Z80E/ASM with ZEUS forms.}

U<ENTER>

```
09466 source                {source text usage.}
00045 table                 {memory for lables in label table.}
00000 GET                   {memory GET files used.}
25119 free                  {unused.}
```

X - Recover source text on reentry to ZEUS.

Syntax <X>

If "X" is the first key pressed upon entry to ZEUS, then

nnnnn text lines.

will be printed, with nnnnn being the number of recovered lines.

If the "X" key, inadvertently, was not the first key pressed, exit and try again.

ZEUS

ZEUS is an small disk operating system, with only one utility. This utility, \$/CMD (for file copying), is customized to work with *ZEUS* only

Although *ZEUS* is supplied single density diskette, *ZEUS* will communicate with double density media in a machine equipped with a double density disk controller. *ZEUS* has automatic hardware recognition, as well as automatic density recognition.

ZEUS has five single keystroke commands.

0 = DIR :0
1 = DIR :1
2 = DIR :2
3 = DIR :3
<ENTER> = repeat last DOS input.

These single keystroke commands only work if they are the first key pressed after receiving the "OK" prompt, or after <BREAK> is pressed.

LIBRARY COMMANDS

ZEUS has 24 library commands. These commands are listed below with a brief description. All commands are terminated with <ENTER>.

APPEND filespec1 to filespec2 - add filespec1 to the end of filespec2.

AUTO doscmd - automatic, invincible, operation of a command or executable file on powerup/reboot.

BOOT - reboot system.

CLEAR - zero memory from 5200H to TOPMEM.

CLOCK (ON) or CLOCK (OFF) - enable/disable time printing in the upper right-hand corner of the video.

CONFIG[[:][drn] (param[,param])] - set default power-up drive attributes.

Param may be:

STEP=nn where nn is the stepping speed of drive drn.
DENSITY=q where q is the default density of drive drn.
SIDES=ss where ss is the number of sides on drive drn.

CONFIG<ENTER> echo's system disk's configuration.

CONFIG (7)<ENTER> echo's RAM's current configuration.

CONFIG (X)<ENTER> writes RAM's configuration onto the system diskette and removes all SKIP's.

DATE[mm/dd/yy] - set date to mm/dd/yy. DATE<ENTER> prints time and

ZEUS

DIR[[:]d][(opt)[, ...]] - print directory of diskette in drive d. All visible files are displayed.

°Pt may be: A, I, K, P, or S.

- A - print directory with file attributes.
- I - include invisible files in directory printing.
- K - include killed files in directory printing.
- P - link printer to printer.
- S - include system files in directory printing.

DUMP filespec (START=X'ssss',END=X'eeee'[,TRA=X'tttt'][,CIM][,TITLE])
Transfer the contents of memory from ssss to eeee inclusive to disk. This command will insert the extension "/CMD" if none is specified, ssss, eeee and tttt are 4 digit hexadecimal addresses, eeee must be greater than ssss. tttt is optional, and specifies the entry point for execution of the file. CIM specifies directs transfer to diskette sectors without 'load marks'. TITLE will place a title block at the beginning of the file's contents.

FORMS[(param)[,param1][,param2]] - set printout parameters.

Where param:

- I initialize line counter and character counter to zero.
- W=www www=1 to 254 (the maximum width of print line in characters). If www = 255 infinite width.
- p=ppp PPP=1 to 255 (the page length in print lines).
- T=ttt ttt=1 to 255 (the printed TEXT in print lines).
- S=sss sss=1 to 127 (the blank spaces between blocks of printed TEXT).
- X Writes the current W, P, and T values to the system diskette for default parameters on subsequent power-ups.
- N=nn nn = nulls after linefeed.
- L send linefeed after carriage return.
- C directs printer output to RS-232-C (Model III only).

FREE - print the number of available file spaces, free granules, and kilobytes of disk space on all mounted diskettes.

KILL filespec - remove filespec from the active roll.

LIB - print *ZEUS* library commands.

LIST filespec - print filespec to the video monitor.

LOAD filespec - load filespec from diskette into RAM and return to OK prompt. The file is not executed.

ZEUS

\$/CMD - Duplicate a single file.

This utility will copy a single file from one diskette to another.

If the filespec for the destination diskette is the same as the source diskette, then the filename does not have to be entered twice.

\$ TEST/ASM:2 1

If the destination diskette's drivespec is the same as the first, then the user will be prompted to swap diskettes.

\$ TOKEN/ASM:2 TO PACK/ASM:2

Whenever drive zero is specified, a "\$" must precede the filespec, and the user will be double prompted to swap diskettes.

\$ \$ZEUS:0 0

The user will receive the message:

Press "ENTER" when a ZEUS system diskette is in drive 0.

or

Insert *ZEUS* <ENTER>•

This message is printed even .if the *ZEUS* diskete is in drive zero.