

Getting Started

with



1953
1954

1955

1956
1957
1958

1959
1960
1961
1962

1963
1964
1965
1966

1967
1968
1969

1970
1971
1972
1973

1974
1975
1976
1977

1978

1979
1980
1981
1982
1983

1984
1985
1986
1987
1988

1989
1990
1991
1992
1993
1994

1995
1996
1997
1998
1999
2000

CHAPTER ONE
INTRODUCTION



Getting Started With PDOS will familiarize you with PDOS so that you will feel comfortable using some of the most common features. It is designed to make your job easier by teaching you how to use PDOS effectively. It is a beginner's guide to PDOS.

As you read along, you will learn how to steer the operating system in the direction you want it to go. This manual was designed to be read from beginning to end since one concept builds upon another. So, you might become confused if you skip around.

This manual is NOT a comprehensive list of all the PDOS functions and is intended only as a supplement to the PDOS Reference Manual. With that manual you can learn the details of the operating system. You will find that this manual refers to the PDOS Reference Manual quite frequently. It will help you understand PDOS better if you also familiarize yourself with that manual.

This manual also does not discuss any of the high level languages supported by PDOS (C, Fortran, Basic, Pascal) as each language has its own manual. This manual will use assembly programs as examples in the fourth chapter. The examples are simple so experience in assembly language programming is not a prerequisite, though it is helpful.

(INTRODUCTION Cont.)

Before you begin with this manual, you should already have installed PDOS on your system. You should be able to boot it up and see the PDOS prompt (>) on your screen. If you are able to follow along with the manual and try out commands as they are discussed, it will help you learn faster. You will need to have your backup utility disk in your system to follow along. For information on how to install PDOS on your system, refer to the Installation and System Management guide for your hardware.

A few conventions have been used in this and other manuals of which you should be aware before you begin. When an example is given, it is indented from the margin and everything that you must enter from the keyboard appears in boldface. The following symbols are used:

- > The PDOS prompt. You do not type this, but when you see it, you know that you are at the PDOS monitor.
- [] Key cap. Press the key enclosed in the braces, ie. [ESC] is the escape key, [CR] is the return key. NOTE: Though control characters will be identified in this manual as [CTRL-C], PDOS will often output the symbol ^ instead. [CTRL] and ^ are interchangeable.
- <> Monitor parameter. When you use a monitor command, you must tell PDOS the information enclosed.
- { } Optional monitor parameter.

About PDOS

PDOS is a realtime operating system for the 68000 micro-processor family. It is both multi-tasking and multi-user and comes with all the development tools you need for an application. You don't need to buy expensive options such as an editor, a kernel, a file manager, and a debugger. They all come standard.

With PDOS, the development cycle is simple. You can develop your application right on the target hardware because the system takes little memory. Then, you can choose the pieces of the operating system that you need (called run modules), put them together with your application, and burn the whole thing into PROM.

So, when you buy PDOS, you get all you need to develop your application. (You will have to purchase the compiler of your choice separately). Then, when you are ready to generate run modules, you may purchase the PDOS Run Module Generation package which includes five sets of run modules licenses for your use. After you have debugged and tested the application, you may purchase the additional run modules you need for production.

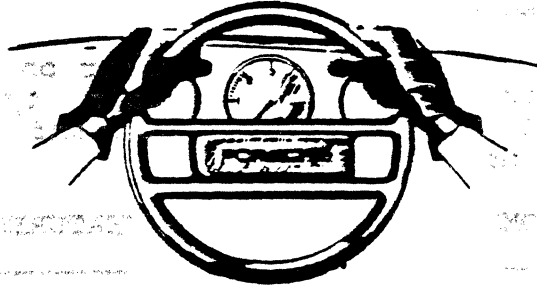
PDOS Hotline and More

The PDOS manuals have all been checked for thoroughness and accuracy. They have been designed for easy reference so that most problems and questions can be answered by using them. Learn how to use them.

If you do run into trouble or need some help beyond what is in the manuals, the PDOS hotline is available for you to use. The PDOS hotline phone number for Eyring customers is (801) 375-2434 during business hours (MST) or you may send a telex to 882000. If you purchased your PDOS from a dealer, support should come primarily from the dealer.

Eyring also offers a full complement of training seminars and consulting services. Training seminars are offered for each compiler as well as 68000 assembly, for beginning PDOS users, and other specialized subjects. Consulting may be arranged on an individual basis with Eyring.

CHAPTER TWO THE PDOS MONITOR



The PDOS monitor is your interface to the operating system. It's easy to see that you are working at the monitor level because the PDOS prompt (>) will be at the left of your screen with the cursor directly after it. If you want to try the commands as they are discussed here, you should already have the prompt on your screen. (If you have not installed your PDOS system, refer to the Installation and System Management guide).

At the monitor, you are in the driver's seat and you can enter commands to steer the operating in the direction you wish it to go. You will learn some useful commands in this chapter. Besides monitor commands, utilities, procedure files, and programs may all be called up from the monitor by simply typing the name of the file after the prompt and terminating with a carriage return.

× MONITOR COMMANDS

PDOS monitor commands are two-character commands which allow you to perform the most common functions with your computer -- functions such as copying and deleting files, listing directories, etc. These commands use memory-resident routines.

All of the commands are shown in upper case. Some must be followed by parameters which specify what you want PDOS to do. The parameters that you are allowed to enter with the command are specified in the command format.

Commands are entered directly after the PDOS prompt. If the command requires no parameters, then it is terminated with a carriage return ([CR]). If you are using parameters, enter them according to the format and terminate with a [CR].

A complete table of all of the monitor commands and their formats is found in the PDOS Reference Manual along with descriptions of all of the monitor commands. Some of the commands you will use most are covered in this chapter.

COMMAND LINE EDITING

To correct an error on the PDOS monitor line is simple. If you wish to begin again, type [ESC] and you will receive a new PDOS prompt and command line. [CTRL-C] performs the same function.

COMMAND LINE EDITING FEATURES

[ESC]	Cancels current line
[CTRL-C]	Cancels current line
[CTRL-I]	Enters insert mode
[CTRL-A]	Recalls last line entered
[CTRL-L]	Moves the cursor right one character
[CTRL-H]	Moves the cursor left one character
[CTRL-D]	Deletes one character to the right of the cursor
[RUB]	Deletes one character to the left of the cursor

If you were to type an invalid command at the monitor and terminate it with a [CR], "PDOS ERR 53 File not Defined" would appear on the screen. All of the error codes are listed in the PDOS Reference Manual.

MONITOR SYNTAX

The PDOS monitor recognizes certain symbols as having special meanings. These symbols can greatly simplify the input you need to make to the monitor. For instance, using a wildcard can relieve you of searching through directories to find a specific filename and it can perform multiple functions quickly.

The wildcard symbols are as follows:

- @ -- wildcard; match 0 or more characters.
- * -- wildcard; match 1 character.

Other symbols accepted by the monitor include:

- # -- auto-define file.
- . -- multiple command separator.
- () -- accept the enclosed argument as a single argument.

The auto-define file (#) symbol saves you the extra step of defining a file before you write characters to it. You will find many ways to use this feature later on.

(MONITOR SYNTAX Cont.)

The multiple command separator (.) allows you to enter more than one monitor command on one line. For instance, typing

```
>CF FILE1,FILE2.LS
```

copies FILE1 to FILE2 and lists out the directory on the screen.

The single argument delimiter (()) is used when a symbol recognized by the monitor is part of an argument. Enclose the argument in parentheses and the monitor will ignore any other meanings of the symbol.

```
>CF (FIL.23:01),(FIL.23:BAK)
```

In the above example, the "." was used as part of the filenames. In order to copy the file without the monitor treating the "." as a new command, the filenames have been enclosed in parentheses.

PDOS FILE HANDLING

A PDOS file is the starting point for any program or text you may create using PDOS. When you write a program or input text for a document, you are working in a file. A filename is used to identify a PDOS file.

Legal PDOS filenames are up to eight characters in length with an optional three character extension. The first character in the filename must be alphabetic (A). The next seven plus the extension can be alphabetic or numeric (C).

PDOS filename format: ACCCCCCC:CCC

For example, FILE1, A123, and FILENAME:01 are legal filenames. 1FILE is not legal because it starts with a numeric character.

(PDOS FILE HANDLING (Cont.))

The filename extension after the colon has no special significance to PDOS; however, there are several standard extensions that will help you to clarify the name of your file. Some common extensions include the following:

COMMON FILENAME EXTENSIONS

```
-----  
:SR      68000 assembly source file  
:OBJ     Object file  
:BAS     BASIC file  
:PAS     Pascal file  
:FOR     FORTRAN file  
:C       C file  
:TMP     Temporary file  
:BAK     Backup file  
none    Executable program file  
-----
```

Making Files

The MAKE FILE (>MF) command allows you to enter ASCII text from the keyboard into a file. (You will learn how to use the editor for large files later. MAKE FILE is only for small quantities of text). The file must have been previously defined using the DEFINE FILE command or preceded by the auto-define (#) symbol. Use the monitor line editing commands to edit the text you are entering (see page GS-5).

>MF <FILE>

Once you enter a [CR], the text is entered into the file. A [CTRL-C] cancels the line without writing it to the file and an [ESC] terminates the process, closes the file, and returns to the PDOS monitor. For instance, if you want to make two files with some text (you can use these files in later examples), you enter the following information:

```
>MF #FIL2[CR]  
PDOS is the realtime standard[CR]  
[ESC]  
>MF #FIL1[CR]  
for the VMEbus[CR]  
[ESC]  
>
```

Copying Files

The COPY FILE (>CF) monitor command can be used to perform the following functions:

1. Make backups of existing files,
2. Move a file to another disk or directory while leaving the original unaffected,
3. Or, print files.

Using the COPY FILE command retains the first file just as it was and produces another copy of that same file.

The COPY FILE command consists of the name of the monitor command followed by the name of the file to be copied (or source file), a comma, and the name of the file to which the first file is to be copied (destination file).

```
>CF <SOURCE>,<DESTINATION>
```

For instance, if you want to back up your file "FILL" to a new file called "FILL:BAK", you type

```
>CF FILL,#FILL:BAK
```

Notice that you must use the auto-define (#) feature to create the file FILL:BAK. If instead, you had chosen to copy the first file to another already existing file, you would eliminate the auto-define feature. In that case, the first file would write over the top of the contents of the second file.

An easy way to print files is to use the COPY FILE command. Actually, what you do is copy the file that you want to print to a driver (TTA is a printer driver provided with PDOS) which outputs text to your auxiliary output port. Your auxiliary output port number is set using the BAUD PORT (>BP) monitor command.

For example, if you want to connect your printer to port 3, you select that port as your auxiliary output port. You can print subsequent files without selecting the port again. (For more information about the BAUD PORT command see the PDOS Reference Manual). An example of using the COPY FILE command to print is as follows:

```
>BP -3  
>CF FILL,TTA
```

Transferring Multiple Files

The COPY FILE command works well if you are copying a single file, but often you want to move more than one file at a time. In that case, you can use the TRANSFER FILE (>TF) monitor command.

This command lets you transfer a group of files at a time while preserving both the creation and modification dates and times. The format to transfer files is as follows:

```
>TF <FILELIST>,<DESTINATION DISK NUMBER>{,A}
```

You can use wildcard symbols in the filelist to name the blocks of files that you wish to transfer. The destination disk number is the number of the disk (either floppy or Winchester) to which you want the files transferred.

Normally, the TRANSFER FILE command queries on each file to transfer. You respond with a "Y" to transfer the file, a "N" to not transfer the file, and an "A" to transfer all remaining files. If you already know that you want to transfer all of the files in the filelist when you type in the command at the monitor, you may specify the "A" as part of the command. If you choose that option, you will not receive queries on the individual files. For example,

```
>TF F1@:SR,3,A
```

transfers all files that begin with 'F1' and have an 'SR' extension to disk 3 without a query. The following:

```
>TF ;@,4
```

transfers all files on your default disk to disk 4 and queries you for "Y," "N," or "A." (The use of the semi-colon is discussed on page GS-14).

Appending Files

The APPEND FILE (>AF) command places the contents of one file at the end of another file. It leaves the first file in its original state. The file format to append a file is the same as to copy a file:

```
>AF <SOURCE>,<DESTINATION>
```

For instance, to append FIL1 and FIL2 used in the MAKE FILE example, you type the following:

```
>AF FIL1,FIL2
```

The contents of FIL2 are then "PDOS is the realtime standard for the VMEbus" and FIL1 still contains "for the VMEbus."

x Showing Files

The SHOW FILE (>SF) command places the contents of a file on your screen. This command is useful if you just want to view the contents of a specific file.

```
>SF (-)<FILE>
```

A few things to remember in conjunction with this command are that you only see one screenful at a time and the lines are automatically truncated to 78 characters per line.

If you would rather not scroll through the screen page by page, place a minus sign in front of the filename. You can stop and start the scrolling by pressing any key. Terminate the command at any point by pressing [ESC].

As an example, if you want to see what is in FIL2, you type in and see the following characters on your screen:

```
>SF FIL2
PDOS is the realtime standard for the VMEbus
>
```

Renaming Files

The RENAME FILE (>RN) command does just what it says -- changes the name of an existing file. The format for the command is also very simple:

```
>RN <OLDFILE>,<NEWFILE>
```

It is not necessary to predefine the newfile when you are renaming files.

```
>RN FIL2,FIL2:NEW
```

Defining Files

The DEFINE FILE (>DF) command is used to define the name and a location in memory for a specific file. You can do a lot more with this command than is covered here, so you will need to consult the PDOS Reference Manual for specific details. However, you may define a file by simply typing the command and the name of the file you wish to define:

```
>DF <FILE>
```

It's easy to define new files.

```
>DF ABC  
>DF ABC:TX  
>DF XYZ
```

X Deleting Files

Deleting files from the monitor is very easy -- so be careful when using these commands. There are two commands that are used to delete files:

```
>DL <FILE> -- which deletes a single file, and  
>DM <FILE LIST> -- which deletes multiple files.
```

To delete a single file, you would type

```
>DL XYZ
```

(Deleting Files Cont.)

The file list parameter in the DELETE MULTIPLE FILES >DM command usually consists of a combination of the filenames and wildcards. For instance, if all of the files you wish to delete have the same beginning name with a different extension, you type

```
>DM ABC:*
```

The first filename with the ABC match in the filename appears on the screen. You are provided with three choices: "Y" which will delete the file, "N" which will leave the file intact, and "A" which will delete all files in the list.

X DISK COMMANDS

PDOS disk commands allow you to change working areas on a disk or find out information about the disk you are on.

System Disks

PDOS allows you to access up to 255 system disks. These defined disks may be a floppy drive, a portion of your Winchester drive, or a portion of RAM. You can define your system disks to be any size you wish. (Instructions on defining disks, partitioning Winchester drives, and assigning system disk numbers are found in the Installation and System Management guide for your PDOS system.)

(System Disks Cont.)

The numbers you see before the monitor prompt are the default system disk numbers that you are accessing. For instance, if you are running off a single floppy drive, your prompt will say 0> (or whatever number your drive has been assigned to). If you are running a PDOS system with multiple disks, you can reference up to four different disks when files are accessed. For instance, 4,2,0> means that you are doing most of your work on disk number four but can access files on disks two and zero. This method of access is helpful when you need utilities or programs from another disk, but want to keep your work separate and in one place.

To change from one disk to another, you must use the SYSTEM DISK (>SY) monitor command. You only need to type in the number of the disk(s) you wish to access. There is a maximum of four disks that you can access at one time.

```
>SY {<DISK>{, <DISK2>{, <DISK3>{, <DISK4>}}}}
```

If you type >SY without any parameters, the disk(s) that are your current defaults are listed. (They always appear before the monitor prompt as well.) You can try changing your system disk defaults and see what happens.

```
0>SY 2,0[CR]
2,0>
```

Directory Levels

System disks are further broken down into directory levels. These levels are not recognized by PDOS as a unique part of the filename; however, they give you a means to sort files.

You may change your default level by using the DIRECTORY LEVEL (>LV) monitor command and typing in the number of the level you wish to use >LV {<LEVELNO>}. If you type >LV without any parameters, the monitor will display the level you are currently on. Unlike system disk defaults, the level number is not displayed unless you ask for it with the DIRECTORY LEVEL monitor command. When PDOS is booted, it will default to level one. You can access a file that is on any level (up to 255 levels) on your system disk.

(Directory Levels Cont.)

PDOS files are found on different levels. Some levels used for different types of files are as follows:

LEVEL	DESCRIPTION
0	Help and system start files
1	Software development tools (MASM/MEDIT/QLINK)
2	Utilities
3	Libraries and include files
12	Source to utilities
14	Source to drivers
18	Benchmarks and demonstration files
20-98	User defined
100-149	PDOS system and hardware dependent files
199	Temporary files

Disks and Levels as Part of the Filename

Both disk numbers and directory levels may be specified when creating and defining a file. If you do not specify a disk number or directory level when you create a file, the file will be written to the current default disk and level. The filename format can be expanded when specifying a disk and level number and is as follows:

ACCCCCC:CCC;NNN/NNN

The semi-colon specifies the level number and the back-slash specifies the disk number.

For example, if you want to back up your file FIL2 from your current location to disk number 4 on level 7, you type in the following from the monitor:

>CF FIL2,#FIL2:BAK;7/4

Listing File Directories

With the LIST DIRECTORY command (>LS), you can list the directory of any system disk or level that you wish. The parameters are the same as those for a filename. Using the LIST DIRECTORY command without any parameters will list out the directory of the system disk and level that you are currently on.

```
>LS {FILE}{:EXT}{;LEVEL}{/DISK#}
```

If you want to list files that are on level 2 of your same disk and you are on level one, you type

```
>LS ;2
```

Wildcards may be used in place of the actual level number to list all files on the disk. For instance, if you want to list all the files on disk number four, you type

```
>LS ;e/4
```

Disk Space

To view the amount of disk space left on your storage device, use the DISK SPACE (>SP) monitor command. You can type >SP to get information for the disk you are on or you can specify a disk number.

```
>SP {<DISK#>}
```

The information you receive from PDOS tells you the number of sectors still available for storage and the largest number of contiguous sectors available. Contiguous sector information is helpful if you wish to define a large file.

```
FREE=NNNN,NNNN  
(Available sectors,contiguous sectors)
```

You also learn the actual amount of space used on the logical disk and the total amount of storage allocated to files on that disk.

```
USED=NNNN/NNNN  
(Used/allocated)
```

MEMORY COMMANDS

Free Memory

The FREE MEMORY (>FM) command lists out the available memory (in Kbytes) for your current task. It also deletes memory from your current task. To list out available memory, do not use any parameters. To delete memory, type in the number of Kbytes that you wish to delete.

```
>FM (<SIZE>)
```

Get Memory

The GET MEMORY (>GM) command adds memory to your current task. Again, the memory is specified in Kbytes.

```
>GM (<SIZE>)
```

The following example shows you several ways to use the FREE MEMORY and GET MEMORY commands:

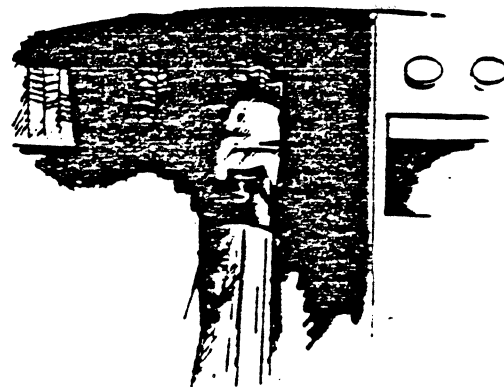
```
>FM                (See what memory is available for your
FREE=0             current task)
>FM 32            (Delete 32 Kbytes from your memory)
>FM                (See what memory is available)
FREE=32
>GM 10            (Add 10 Kbytes to your task memory)
>FM                (See what memory is available)
FREE=22
>GM                (Get all of your memory back)
>FM                (See what memory is available)
FREE=0
```

OTHER MONITOR COMMANDS

There are many other monitor commands other than those you have just learned. Some of them will be discussed later in this manual; the rest can be found in the PDOS Reference Manual.

However, there are a few more monitor commands that are worthy of mention here.

CHAPTER FOUR ASSEMBLE & LINK



Before you can assemble and link programs, you must first have some programs to work with. If you wish to follow along with this chapter, you need to create the following two assembly programs using MEDIT. Name the first program "PROG1:SR" and the second program "PROG2:SR."

```
*PROG1:SR
*
      IDNT  1,0          ;THIS IS VERSION 1.0
      XREF  STRING      ;REF. LABEL 'STRING' IN FILE PROG2:SR
*
START  XPMC  STRING     ;WRITE('HELLO THERE!') USING LABEL STRING
      XEXT                      ;GO BACK TO THE PDOS MONITOR
*
      END   START      ;END OF FILE AND START LOCATION

*PROG2:SR
*
      IDNT  1,0          ;THIS IS VERSION 1.0
      XDEF  STRING      ;ENTRY ON LABEL 'STRING' FOR FILE PROG1:SR
*
STRING DC.B  $0A,$0D     ;OUTPUT A [CR][LF]
      DC.B  'HELLO THERE!',0 ;STRING W/NULL END BYTE FOR PDOS 'XPMC'
      EVEN                      ;ADJUST TO END ON EVEN BYTE
*
      END
```

MASM

After you have edited the two programs, saved them, and exited from MEDIT, the next step is to assemble them. The PDOS assembler is MASM and it is called from the PDOS monitor.

```
>MASM <SOURCE>{,<OBJECT>,<LIST>,<ERROR>,<XREF>}
```

The SOURCE argument is the input text. The OBJECT is a file that will contain the tag object output that may be executed if fully resolved, or become input for QLINK, the PDOS linker. LIST is the full listing of the program. ERROR is a list of errors (these will default to the console if no file is specified). XREF is a list of symbols and lines where used.

(MASM Cont.)

Fortunately, you do not have to remember all of the parameters when you enter the assembler. If you only type >MASM, then you will be prompted for the filenames. Using extensions to delineate object (:OBJ) and listing (:LST) files will help you keep the filenames straight. REMEMBER: PDOS will not create files that have not been defined unless you prefix the filenames with the auto-define (#) symbol.

```
>MASM PROG1:SR,#PROG1:OBJ,#PROG1:LST[CR]
68K PDOS Assembler
ERII, Copyright 1983-85
SRC=PROG1:SR
OBJ=#PROG1:OBJ
LST=#PROG1:LST
ERR=
XRF=
END OF PASS 1
END OF PASS 2
```

If you received errors, then you probably made a mistake typing in the program. Go back into MEDIT and double check your work. Now, since you can only assemble one program at a time, you need to run MASM again for your second program.

```
>MASM PROG2:SR,#PROG2:OBJ,#PROG2:LST[CR]
68K PDOS Assembler
ERII, Copyright 1983-85
SRC=PROG2:SR
OBJ=#PROG2:OBJ
LST=#PROG2:LST
ERR=
XRF=
END OF PASS 1
END OF PASS 2
```

You may want to take a look at the listing files created by the assembler (use the show file command from the PDOS monitor >SF PROG1:LST or >SF PROG2:LST). As you can see, line numbers, section addresses, and data fields have been added to the program along with a summary of symbols at the end. This information is useful when you are debugging your program.

If you had one complete program, you would now be able to run it; however, since these programs are pieces of a larger program, you will need to link them together first.

QLINK

Oftentimes, applications software is developed in modules, so you need a utility that hooks them all together. The PDOS Quick Linker links together assembled programs (object files), libraries, etc. to make an application run. It is called from the PDOS monitor.

```
>QLINK[CR]
```

As you enter the linker, there is an asterisk (*) indicating that QLINK is ready to accept commands. If you enter a [CR], the QLINK command summary and the formats you use when entering commands are displayed on your screen. A list of the most used commands and their definitions are listed below:

*HELP	Display all QLINK commands
*INPUT	Input a single object file
*ZERO	Zero out the memory buffer
*LIBRARY	Load in only those object files that are needed from a library file
*MAP	Display the location and status of memory
*OUTPUT	Select for output the memory buffer to a file
*SRECORD	Specify srecord format in the output file
*SYFILE	Specify binary image in the output file
*END	Output the memory buffer to the file
*QUIT	Return to the PDOS monitor

For a complete listing of QLINK commands, see the PDOS Reference Manual, chapter four.

To link your two object files, PROG1:OBJ and PROG2:OBJ, you would use the following commands:

```
>QLINK[CR]
PDOS 68k Quick Linker 06/12/85
ERII, Copyright 1983-85
*ZERO[CR]
  (This command makes sure that the linking memory buffer is
   all set to zeros before you start to load files.)
*INPUT PROG1:OBJ[CR]
  (Input your first object file.)
ENTRY ADDRESS=00000000
*INPUT PROG2:OBJ[CR]
  (Input your second object file and all subsequent files in
   the same way.)
*MAP[CR]
  (The map tells you the section addresses of each file
   and tell you of any unresolved external definitions or
   references.)
```

(QLINK Cont.)

INPUT FILE MAP:

INDEX	FILE NAME	TYP	IDNT	R	V	DATE	TIME	SECTION	ADDRESSES
1	PROG1:OBJ	A	PROG1:SR	1	0	00/00/85	13:31	0/00000000	00000005
2	PROG2:OBJ	A	PROG2:SR	1	0	00/00/85	13:31	0/00000006	00000015

SECTION GROUPS: NONE

OVERFLOW REFERENCE VALUES: NONE

SECTION	BASE	LOWEST	HIGHEST
0	00000000	00000000	00000016

UNRESOLVED EXTERNAL DEFINITIONS: NONE

UNRESOLVED EXTERNAL REFERENCES: NONE

*SYFILE[CR]

(With this command, you define your program as a system file so that it may be called directly from the PDOS monitor.)

*OUTPUT #PROG[CR]

(This is the name of the file that you will call from the monitor).

*END[CR]

(Tell QLINK to take what has been linked and write it out to the output file and close the file.)

SY FILE: BASE=\$00000000, LENGTH=56

*QUIT[CR]

(Exit to the PDOS monitor.)

RUNNING YOUR PROGRAM

You are now ready to run your program. All you have to do now is type in the name of your program (the output filename you chose in QLINK) at the PDOS monitor.

>PROG[CR]

HELLO THERE!

PROCEDURE FILES

If you find that you are repeatedly reassembling and/or linking your programs, there is an easy way to automate the input you must make from the keyboard. Using procedure files, you can make a file which contains all of the keyboard input in the correct order.

For example, if you want to create a procedure file to perform the tasks that you just did (assemble and link your files together), you need to enter MEDIT and create the following file:

```
>MEDIT[CR]
MASM PROG1:SR,#PROG1:OBJ,#PROG1:LST[CR]
MASM PROG2:SR,#PROG2:OBJ,#PROG2:LST[CR]
QLINK[CR]
ZERO[CR]
INPUT PROG01:OBJ[CR]
INPUT PROG02:OBJ[CR]
MAP[CR]
SYFILE[CR]
OUTPUT #PROG[CR]
END[CR]
QUIT[CR]
RC[CR]
```

The final command (>RC) must always be placed at the end of a procedure file to return the input to the keyboard (Reset Console).

You should now save your file (you might want to name it RUNPRG) and exit MEDIT. Then, using a special monitor command called Set Attributes (>SA), you need to tell PDOS that this file is a procedure file (or an Assigned Console file). For more information on the Set Attributes monitor command, see the PDOS Reference Manual.

```
>SA RUNPRG,AC
```

PDOS then sets an AC attribute for your file and recognizes the file as a procedure file. Now, all you have to do is type >RUNPRG from the PDOS monitor and the assembler and linker will proceed just as if you were typing in the commands directly from the keyboard.

SUMMARY

Now that you've learned some of the fundamentals of PDOS, you can feel free to experiment with other monitor commands and different ways of accomplishing the same task. With PDOS, there is usually more than one way of doing anything. And if you try new ways, you will soon learn shortcuts.

If after you've gone through this manual, you still feel as though you need help, you might consider attending the hands-on "Introduction to PDOS" training seminar held at Eyring. There, you can get individual help and it is more in-depth than this manual. Or, you can keep practicing on your own.

Check your work carefully, but if you think that something may be wrong with your PDOS disks, don't hesitate to call the PDOS hotline so that your frustration can be minimized. And if you haven't already backed up your original PDOS disks, do it right away (see the Installation and Systems Management guide for step-by-step instructions).

Good luck with your applications development.

Help

The PDOS monitor provides a help menu for quick reference. All you need to do is type >HE and a menu of PDOS helps will appear. If you already know in which category you need help, type the name of the category after the HE on the monitor line.

```
>HE (<PARAMETER>)
```

For instance, if you want help with monitor commands, you type in >HE MONITOR; a list of monitor commands and their parameters appear on your screen.

For a list of other helps available, see the PDOS Reference Manual.

Execute

There are many times when you may have exited a program or a utility and wish to return to it. Before you type anything else from the PDOS monitor, type

```
>GO
```

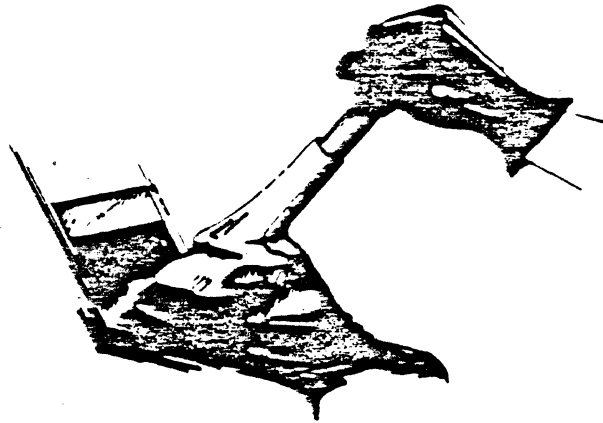
and PDOS returns you to the last program you were in. This command is particularly helpful with MEDIT (see page GS-25).

Set Date and Time

To set the date and time on your PDOS system, you may either do it on boot up or with the set date and time (>ID) monitor command. PDOS displays the current date and time according to its 24-hour clock and you replace it with the actual date and time followed by a [CR]. If the date and time are correct, a [CR] retains the default.

```
>ID
PDOS/68000 R3.0 NN/NN/NN
ERII, COPYRIGHT 1985
          BIOS NN/NN/NN
DATE=00/00/00  4/15/86
TIME=00:00:00  10 31
```

Any delimiter (comma, space, slash, etc.) may be used to separate date and time parameters.

CHAPTER THREEMEDIT -- The PDOS Screen Editor

Let's shift gears for a little while. Instead of working from the PDOS monitor, you can learn about one of the essentials of programming -- using the editor. The PDOS editor is MEDIT (pronounced em-edit) and you will find it to be a useful and flexible tool.

MEDIT is a screen-oriented editor designed for editing files on a CRT terminal. And because MEDIT is a screen editor, you can print out exactly what is on your screen. The screen is constantly updated to reflect the current image of the text that you are editing. Since you can always see what your file looks like, you are less likely to become lost or confused than with line- or character-oriented editors.

By just learning a few of the commands, you can proficiently edit files. There are also many advanced features that can cut your editing time considerably.

MEDIT may be configured to allow you to use your function and arrow keys. You may configure the editor for your terminal using the MEDITCON utility described at the end of this chapter.

PRELIMINARIES

There are some preliminaries you must do before you can enter the editor. You must make sure that your terminal characteristics are set properly with the MTERM utility (see PDOS Reference Manual).

You only need to set your terminal characteristics on boot up. Later in chapter four, you will learn how to write a procedure file. You can then write one which will perform MTERM for you automatically on boot up, but for now, just run it once.

Because the editor functions are configurable, this manual only refers to the name of the function and the default key sequence in parentheses. An example is QUIT ([ESC]^V). You can substitute the key sequence for the function keys you select later.

ENTERING MEDIT AND GETTING A FILE

To enter the editor, you must at the PDOS monitor. Just type >MEDIT. This will give you a blank screen and you can enter any text you wish. MEDIT has no facility for word wrap or automatic carriage returns, so you need to enter a carriage return at the end of each line of text. To edit an existing file, you may retrieve the file from the PDOS disk by using the FILE RETRIEVE ([CTRL-G]) command.

```
[CTRL-G]
'Get File B01:SR'   (a benchmark file on the utility disk)
Verify
V
```

The file then appears on your screen. Another (faster) way to edit an existing file is to enter the filename directly on the PDOS command line >MEDIT {FILENAME}.

The first 23 lines (or a screenful) of the file you retrieved will be on your screen. The bottom line is the MEDIT command line. It is used to display status messages, search strings, prompts, and the clock.

HELP AND LISTING FILES

When you first enter the editor, the status line contains the version number of the editor and the advice "For help, enter [ESC][CTRL-A]."

Call up the help table. Displaying the table will not affect the file that is on your screen. When you have configured MEDIT, your commands will appear in place of the default commands. This same table also appears in the PDOS Reference Manual. Return to your file by pressing any key.

Listing the files on your disk is just as easy. When you use the LIST FILES ([ESC][CTRL-F]) command, the editor will prompt you with "List files '". You may then enter a file specification just as you would with the LIST FILES (>LS) PDOS monitor command. If you enter a [CR], the editor will list out the directory of the current disk unit and level. The directory appears on your screen just as it would from the PDOS monitor. List the directory of the level and unit you are on. After you are through examining the files, you can press any key to return to the screen just as it was before.

THE CLOCK

During editing, the current time is kept in the lower right hand corner of your screen (on the status line). It is only updated when no other processing is occurring and serves as an indicator that the system is alive. When the editor is performing a lengthy process such as writing out a large file, you can watch the clock to see when the process is complete. When it starts ticking again, the editor will accept new commands.

MOVING THE CURSOR AROUND THE SCREEN

The cursor is at the top left hand corner of your screen when you first arrive in MEDIT. As you read through the different ways to move your cursor around in MEDIT, you can try each one out to become familiar with the different commands.

To move the cursor down one character at a time, press MOVE DOWN ([CTRL-J]). To move the cursor up one character at a time, use MOVE UP ([CTRL-K]). The MOVE RIGHT ([CTRL-L]) and MOVE LEFT ([CTRL-H]) keys are on the right and left of the MOVE UP and MOVE DOWN keys. Their location on the keyboard makes them easy to remember. Later, you can configure the arrow keys on your keyboard to do the same thing if you want.

To move the cursor to the end or the beginning of the line you are on, just press the escape key before selecting right or left: JUMP RIGHT ([ESC][CTRL-L]) and JUMP LEFT ([ESC][CTRL-H]).

JUMP DOWN ([ESC][CTRL-J]) and JUMP UP ([ESC][CTRL-K]) move the cursor down or up a certain number of lines. This number of lines can be set by you. The MEDIT default is half of a screenful or 11 lines. To set the jump count, use JUMP COUNT SET ([ESC][CTRL-W]). It will display the current number of lines and allow you to enter a new number.

When you want to move rapidly through your file by using a series of jump commands, you only need to press the escape key once. Subsequent MOVE UP ([CTRL-K]) or MOVE DOWN ([CTRL-J]) commands will be interpreted as jump commands. To cancel this operation, type any command except MOVE UP or MOVE DOWN.

To jump to the beginning of your file, use the JUMP TO TOP OF FILE ([CTRL-T]) command. To get to the end of your file, use the JUMP TO BOTTOM OF FILE ([CTRL-Z]) command.

(MOVING THE CURSOR AROUND THE SCREEN Cont.)

If you know the number of the line where you want to be, you can position the cursor there with the command JUMP TO LINE ([ESC][CTRL-G]). You will then receive the prompt "Goto line" on the status line. Enter the number of the line and terminate with a [CR]. The cursor will then move to that line.

If you would like to move temporarily to another location in your file but return to your current location, mark your present location with the PLACE POINTER ([CTRL-P]) command. Then, after you have moved elsewhere and want to return, you can find your place with the POSITION TO POINTER ([ESC][CTRL-P]) function.

INSERTING CHARACTERS

Inserting characters is as simple as typing. The characters are actually inserted into the buffer while they are displayed on your screen. MEDIT defaults to insert mode which means that if you type characters between other characters, the line is extended and the characters under and to the right of your cursor move over.

If you want to type over the top of the characters under your cursor, the USE REPLACE MODE ([ESC][CTRL-R]) function will allow you to stay in that mode until you return to USE INSERT MODE ([ESC][CTRL-I]).

Normally, only characters that are to be printed are inserted into the text. Most control characters are not inserted because they are either defined as functions or are ignored. Exceptions are the [CR] and tab. The tab is inserted by the INSERT TAB ([CTRL-I]) function.

If you need to insert other control characters (as printer commands, etc.), you must first tell the editor to be ready to accept them or they will be ignored. The INSERT CONTROL CHARACTER ([CTRL-V]) function should always precede any control character that you want to place in the text. The editor will then ignore any significance that the control character may have and insert it into the file.

(INSERTING CHARACTERS Cont.)

Inserted characters are displayed as a single character -- the value of the character plus 32. So, a line feed or control J (10) is displayed as an asterisk (42) and an escape or control [(27) is displayed as a semi-colon (59). This is only apparent on the screen display and the actual control character is stored in the file. For a full list of control characters and their equivalent displays, see the PDOS Reference Manual.

DELETING CHARACTERS

Here is a list and a brief explanation of the different ways to delete characters. You might want to try them out as you read along.

DELETE LEFT ([RUB]) -- This command deletes the character to the left of your cursor, or the character you just typed. You will probably use it the most. If you do not have a key called "rubout" on your terminal, it may be called delete or backspace.

DELETE UNDER CURSOR ([CTRL-]) -- This command deletes the character underneath your cursor. The underline key is usually close to the rubout key. (You do not have to use shift if the underline is upper case).

DELETE LINE ([CTRL-^]) -- This function deletes all the characters to the right of the cursor up to and including the next [CR].

DELETE TO EOL ([CTRL-]) -- This function deletes all the characters to the right of the cursor up to, but not including the next [CR].

FINDING TEXT

There are several ways to search for strings with MEDIT. When you type FIND ([CTRL-F]), the editor will prompt you with "Find string '". Type in the text you wish to search for and end it with a [CR].

The editor then searched for the string from the position of your cursor to the end of the file. When it finds the string, the cursor is repositioned at the end of the string. If the string is not in the file, "Not found" and the string you requested appears on the status line.

(FINDING TEXT Cont.)

If you wish to search from the bottom to the top of your file, use the FIND BEFORE ([CTRL-B]) function. It operates the same as the FIND function only backwards. The editor prompts you to type in your string with "- Find string '".

Both FIND and FIND BEFORE save your last string in a common buffer. You may call it up again by typing either function ([CTRL-F] or [CTRL-B]) twice. Typing the function twice calls the string from the buffer. Then enter a [CR] to complete the command.

If you want to call up all instances of the string one by one, first use either the FIND or FIND BEFORE commands. Then, use the FIND AGAIN ([CTRL-A]) function for all successive searches. You have to do either a FIND or FIND BEFORE command before you can use FIND AGAIN.

There are a few things to keep in mind with search arguments:

1. A [CTRL-Z] that is specified as part of the search string will be interpreted as a single-letter wild card. So, 'F[CTRL-Z]><[CTRL-Z]>D' would match both FEED and FIND.

2. A [CR] (which usually delimits the string) may be searched for by preceding the [CR] with INSERT CONTROL CHAR ([CTRL-V]). Other control characters may be inserted in the same fashion.

3. If you want to quit in the middle of a FIND or FIND BEFORE function, type CANCEL ([CTRL-C]). You will return to normal editing mode.

CUTTING AND PASTING

Most small cutting and pasting operations can be done by using the save buffer.

Go to the end of the portion of text that you wish to copy or cut and mark the location with the PLACE POINTER ([CTRL-P]) function. Then, place your cursor at the beginning of the text and type BUFFER FILL ([CTRL-U]). Either "I got it" or "Overflow" appears on the command line of your screen.

(CUTTING AND PASTING Cont.)

If your screen says "I got it" then the text has been stored in the buffer. You can then feel safe deleting the text from its current position if you want by using the DELETE TO POINTER ([CTRL-]) command. Then, position your cursor in the place where you want the text that is in the buffer to be and use the BUFFER RETRIEVE ([ESC][CTRL-U]) command. The text stays in the buffer until you write over the top of it with other text. So, if you want to repeat the text, it's easy.

As a quick review, let's run through the process again.

1. [CTRL-P] Mark end of text with a pointer.
2. Place cursor at beginning of text.
3. [CTRL-U] Put text in buffer ("I got it").
4. [CTRL-] Delete text.
5. Place cursor at beginning of where you want the text to be.
6. [ESC][CTRL-U] Retrieve text from buffer.

If you get a buffer overflow, you can either divide your text so that it won't overflow and follow the procedure above, or you can write the block of text out to a file.

To write it out to a file, keep your cursor in the same position (the beginning of your block of text) and use the FILE EXCERPT ([CTRL-O]) command.

The editor then prompts you for the name of the file. Name the temporary file a legal PDOS file name (see page GS-6). Be sure that you do not name it the name of the file you are editing or any other file name that is currently used or it will write over the top of that file. (Using "TEMP" or ":TMP" will help you keep your temporary files separate from other files.) Terminate the name of the file with a [CR].

You are asked to verify the function. If your file is a new one, the monitor command line says "Create Verify." If you selected a name that is already a file, it says "Verify." This warns you that you are about to write over the top of that file. If you want to go ahead with the command, type V. Pressing any other key aborts the function.

(CUTTING AND PASTING Cont.)

After your text block has been saved in the other file, delete it using the DELETE TO POINTER ([CTRL-\]) function. When you are ready to retrieve the text, place your cursor where you want the text to begin and use the FILE INSERT ([CTRL-Y]) command. The editor command line prompts you for the name of the file, type the file name and terminate with a [CR]. If you typed in a file name that does not exist, the editor will display "PDOS error #53."

You are asked to verify the command. If you wish to continue, type V and the text appears in your file. Otherwise, press any other key.

SAVING FILES AND EXITING

It is always a good idea to save your files periodically while you are editing. To save or write your file, use the FILE SAVE ([CTRL-W]) command. The editor then prompts you for the name of the file with "Write file ". Type the name (valid filenames are discussed in chapter two) and terminate with a [CR]. The editor then prompts you with a "Verify" or "Create Verify" if the file is new. To verify the command, type V.

MEDIT does not automatically save files when you exit. So, you must always remember to save your file before you leave the editor. To exit MEDIT, use the QUIT ([ESC][CTRL-V]). You are then returned to the PDOS monitor. If you forgot to save your file, you can get back to where you were from the PDOS monitor by typing >GO. You will be right back to where you were editing and then you can save the file. Don't call up MEDIT.

MACROS

Macros are a series of commands that the editor performs repeatedly for you. When you are defining a macro, it remembers every step you make and it will perform those steps as many times as you want. If you want to change all instances of some text to some other text, a macro will save you a lot of work.

Defining a macro with MEDIT is really very simple. First, you must decide exactly what it is you want to do and how you will achieve it. Especially if it is a complex macro, you might want to try it out first. Then, use the MACRO DEFINE ([CTRL-D]) function. You know that MEDIT is recording your steps because the word "MACRO" appears in place of the clock.

(MACROS Cont.)

Now, you can type the sequence of commands you wish to use. The editor shows you exactly what you are doing by performing the functions one time. When you have finished the sequence, press MACRO DEFINE ([CTRL-D]) again.

To perform your macro again, press MACRO EXECUTE ([CTRL-E]). The sequence of events is displayed very rapidly on the status line, but the screen is not updated until the macro has finished executing.

To execute your macro a number of times, press MACRO MULTIPLE EXECUTE ([ESC][CTRL-Z]). The editor prompts you with "Execute macro '". Type in the number of times you wish to perform the macro and terminate the number with a [CR]. If you type in -1, the macro will execute until the operation fails. To stop the macro after it has begun executing, type CANCEL ([CTRL-C]).

If you want to save your macro for future use, write it out to a file with the MACRO SAVE ([ESC][CTRL-O]) function. The status line prompts you with "Macro to file '". You type in a valid PDOS filename that you wish to call your macro and terminate your entry with a [CR]. If the filename has not been used before, the status line reads "Create Verify." Type V to complete the save.

To recall the macro, use the MACRO RETRIEVE ([ESC][CTRL-Y]) function. The editor prompts you with "Get macro file '". All you do is type in the name of the macro you wish to retrieve, terminate with a [CR], and verify the action. If you typed in the file name of a file that does not exist, the status line gives you the message "PDOS error #53." Begin the MACRO RETRIEVE function again and type in the correct name.

MISCELLANEOUS COMMANDS

There are some other MEDIT commands which are worthy of mention. You might want to try some of them out.

RECENTER ([CTRL-R]) -- This function causes the editor to rewrite the screen so that the cursor is centered on the screen. This command is useful when you need to see text before and after the location you are editing.

STATISTICS ([ESC][CTRL-B]) -- If you ask for statistics, the editor displays the number of free bytes, the total number of lines, and the current line number on the status line of the screen.

TOGGLE UPPER/LOWER CASE ([ESC][CTRL-T]) -- This function switches all lower case characters to upper case and all upper case characters to lower case. Mark one end of a block of text with the PLACE POINTER ([CTRL-P]) function, move the cursor to the beginning of the block, and press TOGGLE UPPER/LOWER CASE. The editor asks you to verify the command with a V.

DELETE CONTROL CHARACTERS ([ESC][CTRL-N]) -- When you have a file that has garbage characters in it caused by transmission error, this function eliminates all control characters (and/or characters with the eighth bit set). Sometimes these control characters are difficult to locate with normal find functions. When you press DELETE CONTROL CHARACTERS, the editor asks you to verify the command with a V and then proceeds to eliminate ALL control characters except for tabs and carriage returns.

CLEAR EDITOR ([CTRL-N]) -- This function clears the buffers and memory so that you have a clean screen to begin editing again. Remember that this function erases anything you may have on the screen. For that reason, the editor asks you to verify the command with a V. This command is useful after you have saved the file you are working on and wish to begin a new file without exiting the editor.

CONFIGURING MEDIT

Configuring the editor is really just a matter of running the PDOS utility "MEDITCON" that you can find on your utility disk. So, naturally, to run the utility you must be at the PDOS monitor level.

Before you run MEDITCON, it is a good idea to think out what commands you want for each function. Keep in mind which functions you use the most. A couple of things of which you should be aware are:

1. CANCEL is best left as [CTRL-C] because of its general use in PDOS as CANCEL.
2. [CTRL-X] cannot be used as it is used to clear the type-ahead buffer.
3. [CTRL-S] and [CTRL-Q] are used for handshaking on some terminals. Because PDOS supports this convention as an option, do not use these control characters as editor functions.
4. One command cannot be used as part of another command. For instance, if your F1 key produces [CTRL-A][ESC], then you can't use [CTRL-A] as part of another command.

On the next page is a table which you can fill in before you begin.

MEDIT FUNCTIONS

<u>FUNCTION</u>	<u>DEFAULT</u>	<u>CONFIGURED</u>
Buffer Fill	[CTRL-U]	_____
Buffer Retrieve	[ESC][CTRL-U]	_____
Cancel	[CTRL-C]	_____
Clear Editor	[CTRL-N]	_____
Command Mode	[ESC][CTRL-C]	_____
Delete Control Chars	[ESC][CTRL-N]	_____
Delete Left	[RUB]	_____
Delete Line	[CTRL-^]	_____
Delete Right	[CTRL-]	_____
Delete to EOL	[CTRL-]	_____
Delete to Pointer	[CTRL-\]	_____
File Excerpt	[CTRL-O]	_____
File Insert	[CTRL-Y]	_____
File Retrieve	[CTRL-G]	_____
File Save	[CTRL-W]	_____
Find	[CTRL-F]	_____
Find Again	[CTRL-A]	_____
Find Before	[CTRL-B]	_____
Help	[ESC][CTRL-A]	_____
Insert Control Char	[CTRL-V]	_____
Insert Tab	[CTRL-I]	_____
Jump Count Set	[ESC][CTRL-W]	_____
Jump Down	[ESC][CTRL-J]	_____
Jump Left	[ESC][CTRL-H]	_____
Jump Right	[ESC][CTRL-L]	_____
Jump to Bottom of File	[ESC][CTRL-Z]	_____
Jump to Line	[ESC][CTRL-G]	_____
Jump to Top of File	[ESC][CTRL-T]	_____
Jump Up	[ESC][CTRL-K]	_____
List Files	[ESC][CTRL-F]	_____
Macro Define	[CTRL-D]	_____
Macro Execute	[CTRL-E]	_____
Macro Multiple Execute	[ESC][CTRL-Z]	_____
Macro Retrieve	[ESC][CTRL-Y]	_____
Macro Save	[ESC][CTRL-O]	_____
Move Down	[CTRL-J]	_____
Move Left	[CTRL-H]	_____
Move Right	[CTRL-L]	_____
Move Up	[CTRL-K]	_____
Place Pointer	[CTRL-P]	_____
Position to Pointer	[ESC][CTRL-P]	_____
Quit	[ESC][CTRL-V]	_____
Recenter	[CTRL-R]	_____
Statistics	[ESC][CTRL-B]	_____
Toggle Upper/Lower Case	[ESC][CTRL-T]	_____
Use Insert Mode	[ESC][CTRL-I]	_____
Use Replace Mode	[ESC][CTRL-R]	_____

(CONFIGURING MEDIT Cont.)

Now, you are ready to begin. Type >MEDITCON.

The following message should appear on your screen though the revision number and dates may be different:

```
68000 PDOS 1.1 MEDIT Configurator
Copyright 1985 Eyring Research Institute, Inc.
```

The MEDITCON utility generates a new version of MEDIT which you configure for your terminal. Each function will appear along with the default command for that function. If you want to retain the default command, enter a [CR].

To change the command, press the key or series of keys you wish to use. A few seconds after your last character, MEDITCON will ask for the name of the keys you pressed. You type the name followed by a [CR]. This name will be used in the MEDIT help menu.

If you make a mistake, you can step back to the previous function by using [CTRL-C]. The [ESC] key returns you to the PDOS monitor only if pressed when naming the key.

```
ENTER NAME OF NEW EDITOR
([CR] will name the new editor MEDIT):
```

(This is the name of the editor that you will call up from the PDOS monitor. Remember to use the auto-define (#) feature for new files. If you use a variety of terminals on your system or anticipate using other terminal types, it is a good idea to name your configured editor(s) something other than MEDIT thus saving an unconfigured version of MEDIT. This is because each terminal has a different and often complex sequence of commands for each function key. One way to keep your editors straight is to use the name of your terminal in the editor name. For this example, use the name "NEWEDIT").

```
ENTER NAME OF NEW EDITOR #NEWEDIT[CR]
```

```
ENTER PROCEDURE FILE NAME #NEWEDIT:DO[CR]
([CR] will use the name MEDIT:DO):
```

(This is the procedure file that will build your new editor. Call it "NEWEDIT:DO").

(CONFIGURING MEDIT Cont.)

The configurator now steps you through a list of commands like the table you have already filled out. Your cursor is located at the beginning of the default command. If you wish to continue to use the default commands, simply press the carriage return. For instance, the following will appear on your screen with your cursor located where the underline appears:

Buffer Fill.....[ESC]^U[CR]

(Remember that a control character will show up as a ^). After you type the [CR], the next command will appear as you move through the list.

Move Down.....^J

If you wish to change MOVE DOWN to your down arrow key, simply press your down arrow key. The command sequence used by that key then appears in place of the default command sequence. DO NOT PRESS THE CARRIAGE RETURN or MEDITCON will interpret a [CR] as part of the command sequence.

Move Down.....[ESC]P (output from the down arrow key)

Within a few seconds, MEDITCON asks you the name of the function key. This is the name of the key(s) that you just pressed.

Move Down.....[ESC]P Function key name?Down Arrow[CR]

Just type a description of the key (F1, SHIFT F1, Up Arrow, etc.) and terminate the entry with a carriage return. The next command in the list then appear.

If you wish to return to the previous command, press [CTRL-C] and you can step backwards through the command list. After you have gone through the entire list, the configurator will ask you the following question:

BUILD NEW EDITOR (Y/[N])?

(Answer Y if you are happy with the way you have configured the editor. If you wish to start over, type N. Answering yes will produce similar results on your screen to those below. All you have to do is sit and watch).

(CONFIGURING MEDIT Cont.)

```
>IFNEO;IGNORE
>
>MASM NEWEDIT:DO,#MEDIT:TMP
68K PDOS Assembler
ERII, Copyright 1983-85
SRC=NEWEDIT:DO
OBJ=#MEDIT:TMP
LST=
ERR=
XRF=
END OF PASS 1
END OF PASS 2
>QLINK
PDOS 68k Quick Linker
ERII, Copyright 1983-85
*ZERO
*INPUT MEDIT:OB
ENTRY ADDRESS=00000000
*INPUT MEDIT:TMP
*MAP
```

INPUT FILE MAP:

INDEX	FILE NAME	TYP	IDNT	R	V	DATE	TIME	SECTION	ADDRESSES
1	MEDIT:OB	A	MEDIT:SR	1	8	05/31/85	12:56	0/00000000	000014F1
2	MEDIT:TMP	x	NEWEDIT:D	1	1	05/31/85	13:20	0/000014F2	00001929

SECTION GROUPS: NONE

OVERFLOW REFERENCE VALUES: NONE

SECTION	BASE	LOWEST	HIGHEST
0	00000000	00000000	0000192A

UNRESOLVED EXTERNAL DEFINITIONS: NONE

UNRESOLVED EXTERNAL REFERENCES: NONE

```
*SYFILE
*OUTPUT #NEWEDIT
*END
SY FILE: BASE=$00000000, LENGTH=6410
*QUIT
>DL MEDIT:TMP
>RC
```

Voila! You now have a new editor. You can call it up from the PDOS monitor and try it out by typing >NEWEDIT. You can double-check to make sure that your commands are what you chose by calling up the help table.