

SHOCK DYNAMICS LABORATORY

PDP-11/23 USERS GUIDE

TABLE OF CONTENTS

0. Introduction
1. Getting Started
2. Monitor Commands
3. Editor Commands
4. Using Fortran
5. Logging Off

APPENDICES

1. Programming Suggestions
2. Data Files
3. Command Files
4. Advanced Monitor Commands
5. Advanced Editor Usage
6. Special Printer Characters
7. Extended Memory Usage

INTRODUCTION

This manual is intended to introduce a programmer with little or no experience in the PDP-11/23 to some of its many applications. No knowledge of any computer's operating system is assumed, but you need to know Fortran, of course. The first 5 sections of this manual should be sufficient to allow you to run most programs on the PDP, and for those who want more advanced use of the computer there are appendices covering more topics.

To begin using the PDP, you must place a resident monitor in memory by bootstrapping it (section 1). This monitor controls program execution and the devices (printer, terminal, disk drives), among other things. To communicate with the monitor, you use monitor commands (section 2). To create new files or to change old ones, you use the monitor EDIT command to allow you to use the text editor commands and functions (section 3). Once you have a FORTRAN file that you would like to run, you use monitor commands to do the various steps necessary (section 4).

If you are unsure about how a particular monitor or editor command or an editor function works, try it how you think it should work. The monitor rarely allows you to do something devastating without asking you if that is really what you want to do, and when you use the text editor a backup file is normally created, containing the contents of the file before the beginning of the particular editing session.

Once you understand the first 5 sections reasonably well, you may want to read one or more of the appendices to increase the number of things that you can do with the PDP-11. The appendices do not always tell you everything about the given subject, but they should acquaint you with it and let you know where you can get further information.

Good luck with the PDP, and may the invisible gremlin that lives in all computers avoid your programs....

Greg Bakken
17 Sept. 1981

REFERENCES

RT11 manuals
RT11 pocket guide
keypad editor users guide (RT11 manual, not the chapter in the Intro. to RT11 manual)
keypad editor reference card
MX-80 operation manual
VT100 users guide

GETTING STARTED

Before using the PDP-11/23, first do the following:

1. Sign into the yellow log book.
2. Turn on power switch #4.
3. Put a system disk in the left disk drive.
4. Bootstrap the system by lifting the 3 switches on the computer, from left to right.
5. Place your own disk in the right disk drive, with the label up and out. If you do not have a disk or need another one, disks should be available in the box above the terminal marked "blank disks." A new disk must be properly formatted and initialized before use.
6. OPTIONAL : type in "DA DD-MMM-YY" so your files show today's date if you modify them or create new ones. DD (day) and YY (year) are given in 2 digit decimal numbers and MMM (month) is a 3 character abbreviation.
7. OPTIONAL: type in "TIME HH:MM:SS" where the time is given in 'military time' format, using 14:00 hours rather than 2:00 P.M.

USING THE MONITOR COMMANDS

Monitor commands are used to communicate with the system monitor. They are entered after a prompting "." on the terminal, are enacted by typing a carriage return, and have the following basic form:

COMMAND/OPTIONS (space) ARGUMENTLIST

All commands can be used without any option, and some do not even require an argument list. A "/" must separate the monitor command and the option, and all options must be also separated by a "/". If you are not exactly sure what argument the monitor command expects in what order, you can always type the command and any options with no argument list, and the monitor will let you know what type of argument it expects.

In cases where files are given as arguments, the proper format is:

DEV:FILE.TYP

where dev: stands for the input device. This will usually be DK: unless you use the system disk on drive SY:, and if you use DK: you do not have to specify DK:, as it is the default device.

The following is a list of monitor commands that should be sufficient to allow you to use the PDP to run FORTRAN IV programs and do simple file manipulation. In some cases options are given. If you would like to read further about these commands, other commands, these options, or other options that are available (there are many), look in chapter 4 of the RT11 System Users Guide manual, and for a brief summary look in the RT11 Pocket Guide. The commands and options can all be abbreviated, and their shortest abbreviations are shown in capitals, with the rest of the command given in lower case.

BASIC MONITOR COMMANDS

<u>command</u>	<u>option</u>	<u>arguments</u>	<u>description</u>
COPY		anyfile.typ newfile.typ	creates a new file identical to the old one.
	/Wait	from? SY:(filenames) to? DK: place input in DY0. OK to continue? (Y) place output in DY1. OK to continue? (Y) (files are copied) place system disk in DY0. OK to continue? (Y)	allows you to copy files from another disk to your own. when the computer tells you to, replace the system disk with the one you are copying from, then answer yes to the first two questions, let the computer do the copying, and replace the system disk and answer yes to the last question.

DATE	DD:MMM:YY	Sets the current date to the argument given. If no argument is given, it shows the last defined date.
DIRectory	none	lists the names of your files on the terminal.
	/ALPhabetize	lists in alphabetical order.
	/PRInter	prints a listing of your files.
DELeTe	DEV:FILE.TYP	eliminates a given file or group of files.
EDit	DEV:FILE.TYP	allows you to enter the editing mode and modify the given file.
	/Create	begins a new file with the given name.
	/Inspect	allows you to look at the file without being able to accidentally modify the file.
FORTran	DEV:FILE.TYP	tells the FORTRAN compiler to compile the given FORTRAN program and produce an object file.
	/LIST/WArn	prints out the FORTRAN file and gives a listing of any errors and warnings found.
	/UNITS:N	sets the number of logical units that may be open at any one time. The maximum is 16; the default is 6.
INItialize	DK:	erases all of the files on your disk.
LINK	DEV:FILE.OBJ	takes a previously compiled file and "links" it with the system library, producing a runnable program with a .SAV type.

Print	DEV:FILE.TYP	prints a copy of your file
REName	DEV:OLD.TYP DEV:NEW.TYP	gives a file a new name.
RUn	DEV:FILE.SAV	executes a given file.
SQueueze	DK:	consolidates all free space on your disk. Use this command if your directory listing says you have considerable space but are unable to save files, etc.
Time	HH:MM:SS	sets the time of day. If no argument is given, it displays the last defined time of day. Time is given in 'military time' format, using 14:00 rather than 2:00 P.M.

Note that in some cases it is helpful to use a "*", or wildcard in some monitor command arguments. A wildcard does whatever command you give to all files of the form you type in. For example:

DELETE *.BAK deletes all backup files

RENAME JONDOE.* JOEDOE.* renames all JONDOE files to JOEDOE files of the same type.

COPY A*.* B*.* copies all files that begin with A to the same files beginning with B.

Wildcards can seldom be harmful, because the monitor will not let you do something disastrous without first asking you if that was really what you wanted to do.

Parentheses can also be used to help reduce some typing in file specifications. For example:

SY: ((ONE, TWO, THREE).BAK, FOUR.MAC)

can be used to replace:

SY: ONE.BAK, SY: TWO.BAK, SY: THREE.BAK, SY: FOUR.MAC

Such nesting can be made up to a depth of three.

KEYPAD EDITOR

The keypad text editor is used to alter the contents of files or to create new ones. To enter the text editor, use the EDIT monitor command.

The terminal keyboard has a set of 4 arrows that can be used to move the cursor around in a file. To insert something into your text, simply type it and it will be placed immediately to the left of the cursor. To delete whatever is immediately to the left of the cursor, hit the delete key.

The keypad editor uses functions and commands to manipulate the text. A function is a process you control by typing 1 or 2 strokes on the set of keypad keys on the right side of the terminal's keyboard. A command is a process that you control by first using the command function, then typing in the command, and finally hitting the enter function. Some functions and commands that you might need to use are as follows:

FUNCTIONS

GOLD	sets the next function that you enter to be the lower of the two for the given key.
RESET	Cancels the GOLD function.
PAGE	advances the cursor by 1 page in the direction set by the advance or backup functions; page length can be set by a command or the default formfeed character, (inserted by CTRL L).
ADVANCE	sets the direction the cursor will move in some functions to be down.
BACKUP	sets the direction the cursor will move in some functions to be up.
FIND	asks you for a character string to look for. Hit FIND, give it a string, then hit either ADVANCE or BACKUP to let it know what way to search.
FINDNEXT	looks for the next specified string in the direction specified.
COMMAND	enables you to type a command.
ENTER	executes the command given.

COMMANDS

EXIT leaves the text editor, writes all changes into the edited file, and moves the previous file into it's backup file.

QUIT leaves the text editor and nullifies any changes made.

SET PAGE N sets page length to N lines.

SET TABS N sets tab length.

Below is a diagram of the keypad buttons and their functions:

GOLD	HELP	FINDNEXT	DELLINE
		FIND	UNDELLINE
PF1	PF2	PF3	PF4
PAGE	SECTION	APPEND	DELWORD
COMMAND	FILL	REPLACE	UNDELWORD
7	8	9	-
ADVANCE	BACKUP	CUT	DELCHAR
BOTTOM	TOP	PASTE	UNDELCHAR
4	5	6	,
WORD	EOL	CHAR	
CHNGCASE	DELEOL	SPECINS	ENTER
1	2	3	
BLINE		SELECT	
OPENLINE		RESET	SUBSTITUTE
	0	.	

The number or characters in the lower right corner of each box is the key that enables the function(s) listed.

USING FORTRAN

RT11 uses FORTRAN IV as it's main language. FORTRAN IV has fewer features and library subroutines than the WATFIV found on the computer center's Amdahl does, and you can check these differences in the RT11 FORTRAN IV Language Reference Manual, or the RT11 Pocket Guide on pages 41-47. Besides having slightly fewer features and a few other minor differences, there are two major differences:

1. No control cards, such as \$JOB,\$STOP are used.
2. Data cannot be placed at the end of a program.

To remedy the second difference, you can do one of the following:

1. Use TYPE statements to ask for the data on the terminal, and use ACCEPT statements to collect the data. This is slow for a lot of data, though.
2. Read the data from data files (see appendix). This is not often worth it if you have only a small amount of data, but it is usually the best way to input data, especially if you need to save the input data anyway.

Once you have your program written and any data problems resolved, do the following to execute your program:

1. Compile the program using the FORTRAN monitor command. If you have any errors or warnings get a listing and debug. The program will not execute with any errors present but will with some warnings. Some warnings turn out to be false, but you should check to see.
2. Link the program to the system subroutine library using the monitor LINK command.
3. Begin execution of the program, using the monitor RUN command.

If you wish to run the same program with different data and the data is not contained in the program, you do not have to compile and link it every time.

LOGGING OFF

When you have finished your work on the computer, do the following:

1. Turn the 3 switches to the computer off,
2. Remove disks and return to the proper boxes.
3. Turn off power switch #4.
4. Sign out of the yellow log book. Note any problems that you encountered.

PROGRAMMING SUGGESTIONS

The following may help shorten the time it takes to prepare and/or execute your FORTRAN programs:

1. Use DATA statements, rather than assignment statements, to assign values to variables that will always begin a program or routine with the same value.
2. If you plan to run a program more than once, your data is fairly short, and you don't need a copy of it on a file, use the TYPE and ACCEPT commands to read it in from the terminal.
3. If you need a copy of your data on file, or if your data is fairly long, use data files to read in your data (see appendix).
4. If your program is big and you might need to use extended memory, compile your subroutines separately and see the appendix on extended memory usage to overlay your program into extended memory.
5. If you need to access memory addresses, change the flow of control, or other advanced programming, look in chapter 2 of the RT11 Programmers Reference Manual for a thorough description of all the system subroutines.

DATA FILES

A FORTRAN program on RT11 can use logical unit numbers to read and write with many data files. The following chart shows what logical unit numbers will access what data files:

<u>LOGICAL UNIT NUMBER</u>	<u>DESCRIPTION</u>
2	ftn2.dat data file
3	ftn3.dat data file
4	ftn4.dat data file
5	terminal (read only)
6	printer
7	terminal (write only)
8	ftn8.dat data file
9	ftn9.dat data file
MM ; 10<=MM<=99	ftnMM.dat data file

The following program stub exemplifies the use of data files (in the example FN stands for the relevant format number):

```
READ (2,FN) VAR1,VAR2
READ (2,FN) VAR5
WRITE (9,FN) VAR1
WRITE (29,FN) VAR2
VAR3=VAR1+VAR2
VAR4=VAR1-VAR2
WRITE (9,FN) VAR3
WRITE (39,FN) VAR4,VAR5
```

Note that you can only have a limited number of logical units open at one time. This number is defaulted to 6, but you can make this as high as 16 with the FORTRAN/UNITS:N command and option.

COMMAND FILES

Command files can be used to execute an often used sequence of monitor commands or can be used to execute a series of monitor commands without having to wait for each to finish before entering the next. For example, the following command file runs two programs with different data, renames some data files, and prints the results:

```
rename inp1.dat ftn2.dat      (program input is ftn2.dat)
run   prog1.sav
rename ftn99.dat outp1.dat   (program output is ftn99.dat)
rename inp2.dat ftn2.dat
run   prog2.sav
rename ftn88.dat outp2.dat   (program output is ftn88.dat)
print outp1.dat
print outp2.dat
```

To execute a command file, simply type the name of the file with a "@" in front of it.

In some cases, it is not necessary to use command files because of RT11's 'full readahead' capability. That is, you can type a few commands, each followed by a carriage return, and they will be executed in order. You can only do this for a limited number of commands, though, as the terminal's input buffer has limited space.

ADVANCED MONITOR COMMANDS

The following monitor commands may prove useful as you use the PDP-11/23 more. Brief descriptions are given here; for a detailed description see chapter 4 of the RT11 System Users Guide.

DIRECTORY/ORD:TYP	lists directory by the filetypes.
/ORD:DAT	lists directory by dates.
/ORD:DAT/R	lists directory by dates with the most recent files listed first.
/OU:DEV: FILE.TYP	writes the directory to the given file. This option is good for use with command files.
/BRIEF	only lists files and types.
EXECUTE DEV:FILE.TYP	compiles, links, and runs a given program.
LIBR/(options)	allows you to create and modify libraries that contain often used subroutines. See chapter 4 of the RT11 System Users Guide for the use of the library monitor command or see chapter 12 for use of the librarian program.
TYPE DEV:FILE.TYP	types the contents of a file on the screen. Use the noscroll keyboard button to temporarily halt the typing on the screen.

ADVANCED EDITOR USAGE

Below are brief descriptions of some advanced editor functions and commands that you may find useful as you use the PDP more. For further information refer to the Keypad Editor Users Guide, the Keypad Editor Reference Card, or use the help command to get a diagram of the keypad and a list of all the commands.

FUNCTIONS

SELECT	starts the select range marker for use with certain commands and functions. To adjust select range, use any means of moving the cursor.
CUT	removes the current select range, clears paste buffer, and places the select range contents in the paste buffer.
APPEND	removes the current select range and adds it on to the end of the paste buffer.
PASTE	inserts the contents of the paste buffer immediately to the left of the cursor.

CURSOR MOVING FUNCTIONS

WORD	moves the cursor one word in the direction set by the last ADVANCE or BACKUP function.
EOL	moves the cursor to the next end of line.
CHAR	moves the cursor to the next character.
BLINE	moves the cursor to the beginning of the next line.

RIGHT HAND MARGIN COMMANDS

SET WRAP N	sets the line wraparound length to N characters. This allows you to type text without worrying about your right margin; if a word is too long to fit on the line the editor will move it to the beginning of the next line.
FILL	reformats text lines in the select range so none is larger than the current line length. Lines are left-hand justified.

INPUT filename.typ	opens the file given as the input file for input commands.
OUTPUT filename.typ	opens the file given as the output file for output commands given.
INCLUDE (options)	copies from the auxiliary input file.
SKIP (options)	skips down in the auxiliary input file.
WRITE (options)	writes to the auxiliary output file.

AUXILIARY FILE OPTIONS

N PAGES	processes request by your definition of pages or by the default formfeed character.
N LINES	processes request by text lines.
REST	process rest of the auxiliary file.
WRITE SELECT	writes a select range to the auxiliary output file.

For your further information, the keypad editor reference card is reproduced in following this page.

[TABS] JAD,JUST[[+]*number-of-levels*]

Adjusts the indentation of lines within the select range.

LOCAL[*starting_value* [*increment*]]

Reorders MACRO-11 and VAX-11 MACRO local symbols within a local symbol block so that the local symbols start with the starting value and are separated by the increment. The default starting value is 10. The default increment is also 10.

Miscellaneous Commands

CLEAR PASTE

Clears the paste buffer.

EXIT

Closes all files and restarts the keypad editor.

QUIT

Discards open output files and restarts the keypad editor.

CTRL Key Functions

CTRL C

When a response to the Model: or the Command: prompt is incomplete, cancels the prompt and repaints the screen; otherwise, CTRL C is an invalid character.

When a response to a keypad editor prompt is incomplete, returns to the monitor. On RSX, invokes MCR from keypad editor prompt; does not terminate editing.

CTRL C **CTRL O**

If the keypad editor is executing a function, cancels the process and displays the part of the file where the cursor stopped. For example, stops cursor movement; stops erasure; stops any command process.

CTRL O

Warning: alternate CTRL O keystrokes interrupt and reestablish the display. To avoid getting confused, do not use CTRL O.

CTRL O

Restarts the keypad editor after a CTRL S.

CTRL S

Warning: CTRL S interrupts the output to the screen. To avoid getting confused, do not use CTRL S.

CTRL U

Erases to the preceding line terminator and stores the erasure in the line buffer (buffer capacity is 132 characters); also corrects responses to prompts.

CTRL W

Repaints the screen after using the HELP function or when display is confused. Also restores keypad to alternate mode.

CTRL Z

When a response to the Model: or the Command: prompt is incomplete, cancels the prompt and repaints the screen. Otherwise, CTRL Z is an invalid character. On RSX, returns to monitor if response to keypad editor prompt is incomplete.

CTRL/L *Inserts FORMFEED*

Special Graphic Symbols

VT100 Symbol		VT52 Symbol		Usage
In prompts	In text	In prompts	In text	
 or 	 or 			The cursor
None		None		The end of file symbol
None		None	None	VT100 marking for characters in a select range
	None		None	The Horizontal Tab character
				The Vertical Tab character
			None	The Carriage Return character
	None		None	The Linefeed character
				The Formfeed character
None		None		The Escape character
	None			The New Line Terminator
None		None		Each wrapped line

Buffer Capacities

Character buffer

2 bytes (however, the buffer stores the result of only one DELCHAR or DELETE function)

Word buffer

80 bytes

Line buffer

132 bytes

Paste buffer

512 bytes minimum; the actual capacity is system-dependent.

PDP-11 keypad editor reference card

AV-H854A-TC

VT100 Keypad Functions

To use the lower function on a key, press the PF1 key (the GOLD function) first and then the function key.

GOLD PF1	HELP PF2	FINDNEXT FIND PF3	DELLINE UNDELLINE PF4
PAGE COMMAND 7	SECTION FILL 8	APPEND REPLACE 9	DELWORD UNDELWORD
ADVANCE BOTTOM 4	BACKUP TOP 5	CUT PASTE 6	DELCHAR UNDELCHAR
WORD CHNGCASE 1	EOL DELEOL 2	CHAR SPECINS 3	ENTER
BLINE OPENLINE 0		SELECT RESET	SUBSTITUTE

Note: The letters, numbers and characters in the lower right corner of the keys are what actually appear on the keys.

Common Keyboard Functions

DELETE	Erases the character to the left of the cursor.
LINEFEED	Erases the word to the left of the cursor.
CTRL/U	Erases the line to the left of the cursor.
CTRL/W	Repaints the screen.
CTRL/C	Cancels.
GOLD <i>integer</i>	Repeats any function except DELETE or SPECINS.

digital equipment corporation
COPYRIGHT © 1980

VT52 Keypad Functions

To use the lower function on a key, press the BLUE key (the GOLD function) first and then the function key.



Note: The letters, numbers and characters in the lower right corner of the keys are what actually appear on the keys.

Common Keyboard Functions

DELETE	Erases the character to the left of the cursor.
LINEFEED	Erases the word to the left of the cursor.
CTRL/U	Erases the line to the left of the cursor.
CTRL/W	Repaints the screen.
CTRL/C	Cancels.
GOLD <i>integer</i>	Repeats any function except DELETE or SPECINS.

Using KED and K52 Functions and Commands

To use a function, press the key that controls the function. The keypad diagram shows the locations of functions on the keypad. To use the lower function named on a keypad key, press the GOLD function key first.

To use a command, press the GOLD and COMMAND function keys on the keypad and type in the command. When the command is complete, press the ENTER function key.

The keypad editor sometimes displays the message WORKING... while it is processing commands and functions.

Starting the Keypad Editor

Check that your system is set properly for your VT52 or VT100 terminal. Use the RUN command or another valid system command to start the keypad editor.

The keypad editor prompt is:

* on RT-11 systems

KED> on RSX-11 systems
K52>

Respond to the prompt by typing a file specification string in one of the following forms:

To inspect a file:

filespec / I (RET) on RT-11 systems

filespec / IN (RET) on RSX-11 systems

To create a file:

filespec / C (RET) on RT-11 systems

filespec / CR (RET) on RSX-11 systems

To edit a file:

filespec (RET)

new filespec = *old filespec* (RET)

Stopping the Keypad Editor

To stop a session without saving any open output files, use the QUIT command.

To save all open output files, use the EXIT command.

In either case, when a keypad editor prompt appears, use (CTRL/C) or (CTRL/Z) to return to the monitor, or type another file specification string to start another keypad editor session.

Summary of Functions

Getting HELP

HELP If an error has occurred, the HELP function displays the error message once. If no error has occurred or if the keypad editor is displaying a message, the HELP function first displays the keypad diagram. Use the HELP function once or twice more to see a summary of the keypad editor commands and keyboard functions.

ENTER Restores the screen after using the HELP function.

Using the GOLD Function

GOLD Specifies the lower function of the two functions named on a keypad key repeats a function, or allows special character insertion.

RESET Cancels a GOLD function or a selection range.

Using Arrow Functions

→ Moves the cursor to the next character.

↓ Moves the cursor to the character below.

← Moves the cursor to the preceding character.

↑ Moves the cursor to the character above.

Changing Directional Mode

ADVANCE Changes the direction of cursor movement: moves to the right and downward until BACKUP.

BACKUP Changes the direction of cursor movement: moves to the left and upward until ADVANCE.

Adjusting Tabs (After the SET TABS Command)

(GOLD) A s structured tabs according to the cursor's position and the base indentation as specified with the SET TABS command.

(GOLD) D Decreases the level counter for structured tabs; moves the structured tab column toward the left margin.

(GOLD) E Increases the level counter for structured tabs; moves the structured tab column toward the right margin.

Using Macros (After the LEARN Command)

(GOLD) S Marks the end of a macro; the keypad editor stops storing the macro but does not execute it.

(GOLD) X Executes the current macro.

Summary of Commands

In the following summary, optional words are shown in brackets, for example: [word].

Auxiliary File Commands

[OPEN] INPUT file
Opens an auxiliary input file.

[OPEN] OUTPUT file
Opens an auxiliary output file.

CLOSE
Closes an auxiliary output file.

PURGE
Purges an open auxiliary output file.

INCLUDE options
Copies from an auxiliary input file.

SKIP options
Skips down in an auxiliary input file.

WRITE options
Writes to an auxiliary output file.

options can be as follows.

integer PAGES
Processes pages according to the latest definition you have provided or the default definition (the Formfeed character).

integer[LINES]
Processes text lines.

REST
Processes the rest of the file.

WRITE SELECT
Writes a select range to an auxiliary output file.

Page and Section Defining Commands

SET [ENTITY] PAGE "string"
Defines a page in terms of a marker string. Default: the Formfeed character.

SET [ENTITY] SECTION "string"
Defines a section in terms of a marker string.

SET [ENTITY] PAGE integer[LINES]
Defines a page as a number of lines.

SET [ENTITY] SECTION integer[LINES]
Defines a section as a number of lines. Default: 16 lines.

Search Modifying Commands

SET [SEARCH] GENERAL
Specifies that target and model letters match without regard for case (the default setting).

SET [SEARCH] EXACT
Specifies that target and model letters match exactly.

SET [SEARCH] BEGIN
Specifies that the cursor be placed at the beginning of the matching target (the default setting).

SET [SEARCH] END
Specifies that the cursor be placed to the right of the matching target.

SET [SEARCH] BOUNDED
Limits searches to one page according to the latest definition you have provided or the default definition (the Formfeed character).

SET [SEARCH] UNBOUNDED
Allows searches for a matching target between the cursor and the top or bottom of the file (the default setting).

Visual Control Commands (KED Only)

SET QUIET
Specifies that the screen display be reverse when an error occurs.

SET NOQUIET
Specifies that the bell be rung when an error occurs (the default setting).

SET [SCREEN] 80
Displays 78 characters on a screen line (the default setting).

SET [SCREEN] 132
Displays 130 characters on a screen line.

SET [SCREEN] DARK
Displays light characters on a dark background (the default setting).

SET [SCREEN] LIGHT
Displays dark characters on a light background.

Using Macros and Reformatting Text

LEARN
Erases the latest macro and stores the commands and functions that follow.

SET WRAP[line_length]
Sets the right margin for the word-wrap feature, the KED FILL function, and the K52 FILL command. The default line length is 78 characters when the screen width is 80 and 130 characters when the screen width is 132.

SET NOWRAP
Cancels the word-wrap feature but does not affect the FILL function.

FILL
Reformats the text lines within a select range so that none is longer than the current line length. The default line length is 78 characters.

Using Structured Tabs and Reordering MACRO-1 Local Symbols

SET TABS[indent]
Specifies the basic amount of indentation for the structured tab feature. The default indent is 4.

SET NOTABS
Cancels the structured tab feature.

EXPERIMENT 01
SHOCK TUBES LABORATORY
DEPARTMENT OF PHYSICS
WASHINGTON STATE UNIVERSITY

Moving the Cursor by Standard Units of Text

BOTTOM	Moves the cursor directly to the end-of-file symbol (␣).
TOP	Moves the cursor directly to the top of the file.
BLINE	Moves the cursor to the beginning of a line.
CHAR	Moves the cursor to the next or preceding character.
EOL	Moves the cursor to a line terminator character.
WORD	Moves the cursor to the beginning of a word.

Moving the Cursor by Pages and Sections

PAGE	Moves the cursor to the top of a page. The default definition is a Formfeed. To insert a Formfeed, type CTRL/L.
SECTION	Moves the cursor to the top of a section. The default definition is 16 lines.

Use the SET ENTITY commands to change the definitions of PAGE and SECTION.

Searching for Strings in the File

FIND	Type a new search model, then press the ADVANCE key or BACKUP key to specify the search direction.
FINDNEXT	Searches for another occurrence of a target that matches the latest search model.

Use the SET SEARCH commands to specify exact or general searching, the limit of the search, and the cursor's location at the search target.

Repeating Functions

If the VT100 autorepeat feature is enabled, you can repeat a single key function and single character insertion slowly by holding the key down. For the same effect on a VT52, hold down both a function key and the Repeat key.

GOLD *integer* Repeats quickly any function except SPECINS, DELETE, and CTRL/U.

Inserting Material into the Main File

To insert any printing characters on the keyboard, type the characters. To insert non-printing characters, use the SPECINS function.

SPECINS	Press the GOLD key, type the ASCII decimal equivalent of the non-printing character you want to insert, and then use the SPECINS function to insert the character.
OPENLINE	Breaks the current line by inserting a New Line terminator at the cursor's right.

Erasing and Restoring by Standard Units of Text

CTRL/U	Erases to the preceding line terminator and stores the string in the line buffer (buffer capacity is 132 characters); also corrects responses to prompts.
DELCHAR	Erases the cursor's current character and stores it in the character buffer.
DELEOL	Erases to the next line terminator and stores the string in the line buffer (buffer capacity is 132 characters).
DELETE	Erases the preceding character and stores it in the character buffer; also corrects responses to prompts.
DELLINE	Erases through the next line terminator and stores the string in the line buffer (buffer capacity is 132 characters).
DELWORD	Erases to the first character of the next word and stores the string in the word buffer (buffer capacity is 80 characters).
LINEFEED	Erases through the first character of the current or preceding word and stores the string in the word buffer (buffer capacity is 80 characters).
UNDELCHAR	Inserts the characters from the character buffer.
UNDELLINE	Inserts the string from the line buffer.
UNDELWORD	Inserts the string from the word buffer.

Building a Select Range

SELECT	Marks one end of a select range. When the cursor is moved, use the following functions and commands to process the select range: Functions: APPEND, CHNGCASE, CUT, FILL, REPLACE Commands: FILL, TABS, ADJUST, WRITE SELECT The RESET function cancels the select range.
---------------	---

Moving, Copying, and Reformatting Text

APPEND	Erases the select range and stores it at the end of the paste buffer.
CUT	Erases the select range and stores it in the paste buffer (buffer capacity is at least 512 characters).
FILL	Reformats the text lines in the select range (a function key for KED only).
PASTE	Inserts the string from the paste buffer.

Substituting

CHNGCASE	If there is a select range, changes all lower case letters to upper case and all upper case letters to lower case in the select range; otherwise, if the cursor is located properly, changes the case of letters in a search target. If no select range is being built, changes the case of the cursor's character and advances or backs up the cursor by one character, depending on the directional mode.
REPLACE	If there is a select range, erases and discards the select range; otherwise, if the cursor is located properly, erases and discards a search target. Then inserts the string that is in the paste buffer.
SUBSTITUTE	When the cursor is at a search target, completes the REPLACE function and then the FINDNEXT function.

SPECIAL PRINTER CHARACTERS

The EPSON MX-80 printer that the Shock Lab has is capable of using some special printing characters that are generally useless to program writers but may help those using the PDP for papers, notes, etc. Some of the most simple are shown on the following page. These codes are entered using the editor SPECINS function. For further information on these control codes, see the MX-80 Operation Manual.

SUMMARY OF MX-80 CONTROL CODES

Control code data is sent to the MX-80 along with character code data via the input data lines. Control codes are sent as data, and are interpreted as instructions by the Printer. The following is a summary of control codes and control code sequences recognized by the MX-80.

(1) Print action codes

Mnemonic	Decimal	Hex. code	Function
CR	13	0D	Performs the printing of a line stored in the print buffer.
LF	10	0A	Performs the printing and advances the paper one line.

(2) Paper movement codes

FF	12	0C	Advances the paper vertically to the next predetermined Top of Form position.
ESC C	27, 67	1B, 43	Specifies the form length per page. Execution command is FF.
HT	9	09	Advances the print head to the next predetermined tab stop position in a horizontal direction.
ESC D	27, 68	1B, 44	Specifies the horizontal tab stop positions. Execution command is HT.
VT	11	0B	Advances the paper to the next predetermined tab stop position in a vertical direction.
ESC B	27, 66	1B, 42	Specifies the vertical tab stop positions. Execution command is VT.
ESC 0	27, 48	1B, 30	Causes the line spacing to be set at 1/8 inch.
ESC 1	27, 49	1B, 31	Causes the line spacing to be set at 7/72 inch.
ESC 2	27, 50	1B, 32	Causes the line spacing to be set at 1/6 inch. This code is also a command to execute the predetermined line spacing mode.
ESC A	27, 65	1B, 41	Specifies the amount of line spacing in the Line Feed. Execution command is ESC 2.

(3) Character designation code

SO	14	0E	Instructs the Printer to print subsequently received characters in double width size.
DC 4	20	14	Cancels the SO (enlarged character) mode.
SI	15	0F	Instructs the Printer to print characters in condensed size.
DC 2	18	12	Cancels the SI (condensed character) mode.
ESC E	27, 69	1B, 45	Causes the Printer to print emphasized characters.
ESC F	27, 70	1B, 46	Cancels the ESC E mode.
ESC G	27, 71	1B, 47	Causes the Printer to perform double printing.
ESC H	27, 72	1B, 48	Cancels the ESC G mode.

(4) Other codes

NUL	0	00	Terminates the tabulation setting sequence.
BEL	7	07	Causes the buzzer to sound for about 3 seconds.
DC 1	17	11	Places the Printer in the selected (data receivable) state.
DC 3	19	13	Places the Printer in the deselected (data not receivable) state.
ESC 8	27, 56	1B, 38	Permits data to be transmitted even if there is no paper in the Printer.
ESC 9	27, 57	1B, 39	Cancels the ESC 8 condition.
CAN	24	18	Deletes from the print buffer all previously entered printable characters. (Print buffer clear command)
DEL	127	7F	Functions the same as 'CAN'.

EXTENDED MEMORY USAGE

The memory of the PDP-11/23 is essentially divided into 2 groups: low memory, which consists of the first 64 K addressable locations (bytes), and high memory, which stretches from the end of low memory to 256 K bytes. Most programs use low memory exclusively for both data and programs. The high memory can be used for both data and programs as follows:

DATA USE

Replace all arrays of any size with VIRTUAL arrays, if possible. If an array is defined by a DIMENSION statement, simply replace DIMENSION with VIRTUAL. If a given array is part of a COMMON statement, simply remove it from the COMMON statement and place it in a VIRTUAL statement, making sure that the particular VIRTUAL statement appears everywhere the particular COMMON statement does. There are some restrictions on the use of virtual arrays, however. The two main ones are as follows:

1. VIRTUAL arrays must not be used on COMMON, EQUIVALENCE, DATA, ENCODE, or DECODE statements.
2. If a VIRTUAL array is passed as an argument to a subprogram, its value must not be altered.

For further information on virtual arrays, see the RT11 FORTRAN IV Language Reference Manual, on pages 5-6 to 5-9.

PROGRAM USE

The use of programs in extended memory requires virtual overlays to place most of the program in extended memory and selectively call parts of it into extended memory for processing. This should not normally be done until after you have tried placing all the data in extended memory. For complete information on the use of VIRTUAL overlays, see section 1.4 of the RT11 FORTRAN IV Users Guide.

INTERNAL REPORTS - 1982

1. G.E. Duvall, "Transition Probabilities, Absorption Coefficients and Oscillator Strengths in CS₂", Internal Report 82-01, January, 1982.
2. G.E. Duvall, "Excitation of Bending Vibrations in CS₂", Internal Report 82-02, January, 1982.
3. R. Granholm, "Double-Diaphragm Breach for Atmospheric Pressure", Internal Report 82-03, February, 1982.
4. G.E. Duvall, "Equation of State Fluids. I. Nitromethane", Internal Report 82-04, November, 1982.
5. G.E. Duvall, "Equation of State: The Mie-Gruneisen Equation", Internal Report 82-05, December, 1982.
6. G.E. Duvall, "Approximations to the Hugoniot P-V Curve of Fused Silica", Internal Report 82-06, December, 1982.
7. P. Bellamy and G.E. Duvall, "Spall Pressure in Cobalt-Aluminum Systems", 1982.