



DECUS

PROGRAM LIBRARY

DECUS NO.	8-466A
TITLE	RL MONITOR SYSTEM (WCFMPG VERSION) P?S-08-1.1A
AUTHOR	Richard Lary, et al Submitted by: Stanley Rabinowitz, et al
COMPANY	Polytechnic Question Society Brooklyn, New York
DATE	March 1, 1971
SOURCE LANGUAGE	PAL III

20330

LIBRARY



EXHIBIT

...

...

...

...

RL MONITOR - USER'S GUIDE

TO LOAD THE SYSTEM

1. Mount a system DEctape; space it up several feet; set the tape transport switches to "remote" and "write enabled" off (i.e., write protected).
2. Load the tape bootstrap into memory, locations 7600 and up.
3. Set keys to '7600.'
4. Press "LOAD ADD" key
5. Press "START."
6. When teletype prints a period and rings the bell the system is running and waiting for commands.

If the tape bootstrap is not already in memory, it can be loaded using the auto-bootstrap.

CREATING A FILE

A file is a contiguous set of blocks of information. Each of these blocks is 128₁₀ machine words long. A file is a maximum of 2044 words, or 4088 characters. Files may contain programs in symbolic code or machine language.

CREATING A FILE OF SYMBOLIC CODE

1. Load the system .
2. Type "SCR ↵" :this clears the file in core .
3. Type "SE_n_m ↵" :causes automatic line numbering from n by m's .
4. Type program line-by-line; a line is a set of characters followed by a " ↵ " .
5. When program is entered, or one desires to leave the automatic line numbering mode, type CNTRL "D" (hold down control key and type a "D").
6. If one tries to put too many characters into the file the monitor will respond with a "WHAT". At this point one should delete the last few lines which were accepted, close the present file, save it on tape, then create a new one and continue entering the program by going to step 2. When the assembler or other system programs use the files they are concatenated and their particular beginning and endings are ignored.

N.B. To delete a line but remain in sequence mode type ALT MODE. To delete the last character, a space is a character, type a "←". Line-feeds are ignored.

CREATING A PROGRAM

To convert a PAL III program to absolute code follow these steps:

1. Break the program into sections such that each section can fit into one file (approximately 4000 characters) being sure to leave room for corrections and insertions. Usually one places the code for a page or two of core into one file.
2. Make the files as described in "CREATING A FILE."
3. Assemble the program as described in "ASSEMBLING A PROGRAM."
4. Load and run the program as described in "LOADING AND RUNNING A PROGRAM."
5. Correct the files as described in "CORRECTING A FILE."
6. Repeat steps 5, 3, and 4 until the program works.

N.B. If the last file named in the "RUN PAL ..." command is not terminated with a "\$" the machine will enter an infinite loop waiting for characters from the teletype. At this time the user may type the "\$" and the assembly will continue. It will be necessary to type the "\$" after each pass of the assembly.

CORRECTING A FILE

1. Load the system.
2. Type "LOAD <name> ↵" as, for example, "LOAD MYJOB ↵".
3. Modify file in any of the following ways:
 - a. To delete a line, type line number and carriage return.
 - b. To alter a line, type line number, space, and new line followed by a carriage return.
 - c. To insert a line between two lines - same as b, except that the line number used should be between the two bracketing line numbers.

Example:

```
If initially  10 A and you want "AB"
              20B

between "A" and "B" then type "15 AB ↵"

to get  10 A
        15 AB
        20 B
```


EXAMINING A FILE

Files should always be examined after being typed. One must always keep in mind that errors in printout may be caused by a faulty teletypewriter. The listing process may be interrupted by typing any character; this character will not be printed. The line being typed will always be completed. Typing premature termination commands is meaningful only for listings of more than one line.

LISTING ONE LINE

Type "LIST < number > ↵" as in "LIST 340 ↵".

The computer responds by typing the line.

LISTING A FILE

1. Load the file if it is not already in core.
2. Type "LIST ↵".

LISTING A SEGMENT OF THE FILE

1. Type "LIST < number > - ↵" and the computer will type all lines from the number given to the end of the file.
2. Type "LIST - < number > ↵" and the computer will type all lines from the beginning of the file to the given number.
3. Type "LIST < number₁ > - < number₂ > ↵" to list all lines from number₁ to number₂ inclusive.

SAVING A FILE

1. We assume that the file of interest is in core as discussed in "CREATING A FILE" and "CORRECTING A FILE."
2. If you are not out of automatic sequence mode type a control D to leave it.
3. Enable DECTape writing, left light on TU55 transport on.
4. Type "SAVE < name > " as "SAVE MYJOB ↵". This causes the present file in core to be saved on DECTape under the label < name >. For the example this would be the label "MYJOB."

If the name is new it is added to the catalog in the first free location and the file is put on DECTape. If the name has been used before, and hasn't been deleted, the present file in core will be stored over the information previously associated with that name. The old contents of the file will be destroyed. Once a file has been stored under some name it may be referenced by that name by any command which requires a < name > field (argument).

N.B. Be particularly careful not to store a blank file on top of a useful file. This would cause the useful file to be completely lost. This error is most often committed when one wishes to correct a file and forgets to load it before making the corrections. Any time the system is loaded or an "SCR ↵" or "MO ↵" command is given the file is blank.

DELETING AND LOADING A FILE

DELETING A FILE

A file name is deleted, hence freed, by removing it from the catalog. This is done by typing "DE <name> ↵". If the name does not exist the monitor will type "NF" for 'not found.' If it does exist a period will be printed indicating the successful completion of the erasure.

LOADING A FILE

To load a file from tape into the working store of the system type "LO <name> ↵ as in "LO MYJOB ↵". If successful the TTY will reply with a period. If the file does not exist the machine will reply "NF."

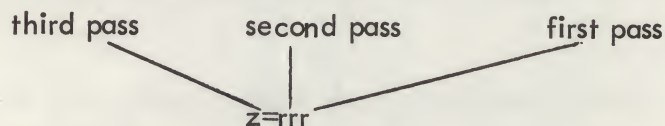
ASSEMBLING A PROGRAM (RUNNING PAL III)

1. Under system control type the command:

"RUN PAL=x0 yz, <name₁>, <name₂>, . . . , <name_n> ↵"

where the symbols above have the following interpretation:

x=0	Print symbol table
x=4	Do not print symbol table
y=0	Put binary output on paper tape
y=1	Put binary output on DECTape in special file named "\$"
z=rrr	Where each r is a 0 or 1. Zero means do not perform this pass of the assembly and one means perform the pass



x, y, and z are written as octal numbers, not binary.

The use of the passes is as follows:

- Pass 1 - Make the internal symbol table for the assembly
- Pass 2 - Using the symbol table of pass one, create binary
- Pass 3 - Using symbol table of pass one, make full listing of the program

The symbol table controlled by "x" is printed at the end of pass two. If DECTape is used for output, y=1, it must be write enabled. If it is not, the machine will halt. Recover by enabling the tape write; press continue on the console.

The names above are file names. They are taken as the source of symbolic code, in order, one at a time. If the last is not terminated by a "\$" PAL goes to the TTY for input. It will loop on the TTY waiting for characters, so if you forget a "\$" remember this source of difficulty when the system seems tied up. The maximum number of files is fifteen, n=15.

2. Example:

```
"RUN _ PAL=13,MYJOB ↓ "
```

LOADING A BINARY PROGRAM

The most recent assembled program is available in a special file named "\$". "\$" may accommodate a maximum of about fifteen binary files. The "\$" file may be saved under an alternate name by using the "BSAVE" command. "BSAVE" 'd and "\$" files are called binary files. To load up to five binary files type "RUN _ BIN=xxxx, <name₁> , <name₂> , ..., <name_n> " where the symbols have the following meanings:

name ₁	represents a binary file
n	an integer up to five
xxxx	the starting address in octal of the program. This location will be branched to upon entering the program

Unless one is using location overlay the files may be in any order. If the "\$" file is to be used it must be first. If no starting address is given the machine halts after loading the binary files.

Example:

```
"RUN _ BIN, $ ↓ "
```

Notational update: Since writing the above, the following notational change has been made. PAL binary files are saved in a special file called THE BIN. It may be saved in a user file by the command BSAVE <filename>. It can be run by either the command, RUN BIN or RUN BIN,\$. When running several binary files at once, THE BIN can be included only if it occurs first and is indicated with a dollar sign, e.g., RUN BIN,\$,MYJOB. The command RUN BIN, MYJOB does not run the binary file stored in THE BIN.

There is also a special user file known as "\$" which shares space with the end of THE BIN. It can be referenced by commands such as LOAD \$, SAVE \$, and RUN PAL=13,A,\$,B. Care must be taken with this file since it can be destroyed by large assemblies. Since it is at the beginning of the tape, it can be used for quick editing and once it is correct can be saved in a different named file.

TRANSFERRING FILES

A symbolic file may be transferred from one tape to another. It is also possible to rename a file. Any attempt to manipulate a binary file, IN ANY WAY, except to load it, will cause the file to be changed randomly (not quite, but close enough). The only command to manipulate a binary file is "BSAVE _ <name> ↓ " which is used to move a binary file from the "\$" file to another location so that the results of an assembly may be permanently saved. Perform an "SCR" before executing any of the following.

* See next page

TO SAVE A FILE UNDER A NEW NAME

1. Load the file "LO < name > ↵ "
2. Save under new name "SA < name > ↵ "

TO RENAME A FILE

1. Load the file "LO < name > ↵ "
2. Delete the file under the old name "DE < name > ↵ "
3. Save under new name "SA < new name > ↵ "

TO TRANSFER A FILE FROM ONE TAPE TO ANOTHER

1. Load the file "LO < name > ↵ "
2. Mount new tape; enable tape writing
3. Type "SYS" to read in the new tape's directory "SYS ↵ "
without changing the file stored in core
4. Save file under any name "SA < name > ↵ "

DUPLICATING THE SYSTEM ONTO ANOTHER DECTAPE

To create a fresh copy of the system on a formatted DECTape do the following. The old contents of the tape will be destroyed. As a safety precaution keep the write disabled, write enable off whenever the original DECTape is on the tape drive. We will refer to the tape on which the system is being placed as the "new tape."

1. Load the system from any tape.
2. Type "DUP ↵" and wait for the machine to halt.
3. Mount new tape and space forward a few dozen feet.
4. Press the "Continue" lever on the console.
5. When the machine halts put on the original system tape.

* Correction: Binary files may now be safely manipulated. Thus the sequence

BSAVE BFILE
LOAD BFILE
SAVE BFILE2

is permissible. Binary files can thus be transferred from one tape to another. A binary file cannot be listed. If you do so by mistake, hit stop and rebootstrap.

6. Space up the tape a few dozen feet.
7. Press "Continue" lever on console.
8. When machine stops mount the new tape etc. as in step 3. Keep performing 3, 4, 5, 6 and 7 until the system bootstraps itself with the new tape on the tape transport. The TTY will go "ker-chunk."

The system is usually transferred in three segments. BE SURE TO PROTECT THE ORIGINAL SYSTEM TAPE WHEN DUPLICATING. KEEP THE WRITE DISABLED. If the write is accidentally disabled when the new tape is on the transport, merely enable it and press Continue again. This is generally true of this system.

Note: Some tapes have more systems on them than other tapes. It may take more than 3 segments to transfer them. The DUP command will automatically bootstrap onto the new tape no matter how many segments are required.

N.B. All files on the new tape will be destroyed.

SUMMARY OF THE SYSTEM

The monitor system for the PDP-8 was written by Richard Lary in the early months of 1968. It was designed to run on a PDP-8 with limited hardware; 4K of core, teletype, and one DECTape (TC01 controller and TU55 transport). The following description of the system commands is accurate; nothing happens except what is stated. Nothing is assumed or understood.

The general action of the system is as follows. We wish to be able to run PAL, the assembly language, on programs. Then we wish to load and run them. To do this efficiently we must be able to enter programs in symbolic form and make corrections. Users of the DATANET system, GE's, will recognize our goals and methods of accomplishing them. Our environment is limited by our small amount of core and the fact that we have one DECTape. Essentially this only restricts the numbers, limits, of certain quantities, not the type of commands available. The monitor permits the entering of characters, storing them on the DECTape, getting them back and changing them, and the ability to cause PAL to run with these characters as input, i.e., as assembly. This is accomplished as follows.

The DECTape is divided up into two sections, a system section and a section for storing the user's programs. The system section contains the system, the monitor, and several languages. These languages are run on various user programs and the results are available in several ways. The user's section contains a set of files. These are of fixed length, 2044 words, and are in a fixed position on the DECTape. Associated with each file is an entry in a catalog which is kept in the system sections of the tape and also in core when the monitor is running. In this catalog are kept the names of the files. These names, a maximum of six alphanumeric characters, are used to reference the file. The action is as follows: If a name is given in a command, it is looked up in the catalog. Let us say it was the n'th name, rather, the name associated with the n'th position in the catalog. Then the monitor knows that the user is indicating the n'th file in the user section of the tape.

In this system the monitor and the editor are together. So, when the monitor is running, room for editing must be available. The monitor and editor use only about half of core leaving the other half as a buffer for the editor. This buffer is exactly the size of one file. It is referred to as the working file because it is the only file which may be changed or created. Once one has made this file it may be stored or retrieved from tape. Thus, the system commands for editing break down into two types, those which are used to manipulate the working file and those which manipulate the tape by moving the working file or altering the catalog. The remainder of the commands are true monitor commands. They effect the running of system and user programs. We now may describe the details of using the monitor.

IN CASE OF FAILURE

There are several reasons why the monitor may fail to do what is described below. Sometimes one may recover, other times all is lost. For this reason it is important to keep copies of the source files which are useful. For very important work two copies may be kept, one as a backup and one as a master. Barring machine errors we have the following considerations. When the system was being read in an error may have been made and the monitor was not loaded correctly. This is extremely rare unless the tape drive is broken. Similarly, the system could have been duplicated incorrectly. There may be bugs in the system. Although this is unlikely, because we have used the monitor extensively for some time, it is possible. However, all the really useful commands have been thoroughly tested.

GENERAL SYSTEM NOTES

- a. If the system tries to write one tape but cannot because tape writing is disabled, it will stop ---- recover by enabling tape writing and pressing "Continue" on the console.
- b. At any time in the typing of a line an ALT MODE character will cause the line to be ignored. This does not delete a line in a file.
- c. All lines in files must be entered with line numbers.
- d. A line in the working file is deleted by typing the linenummer followed by a "↵".
- e. If two lines are entered with the same line number, only the last entered is present.
- f. b. applies to system commands also.
- g. To delete the last character typed in a string type a "←." This applies to commands and inputs to the editor.
- h. An EOT (control D) will act as an ALT MODE but will take one out of line numbering mode if he is in it.
- i. Keep writing disabled until you need to do it. This can prevent many a headache.
- j. Be very sure the working file is empty (null) at such times when this state is necessary, particularly
 1. When correcting a file, before loading it.
 2. When creating a new file, before entering it.

k. When correcting a file be sure the file being corrected is in the working file space before making the corrections, otherwise, you may store the corrections over the original file causing it to be destroyed and only the corrections will remain.

l. To stop a listing - at the end of the present line, type any character ; immediately press the "stop" switch on the console; "Continue" continues.

m. Hitting control tape causes the tape to rewind.

SUMMARY OF COMMANDS

MONITOR COMMANDS

(Abbreviated form in parentheses)

MOUNT (MO) transfers to 7600 to bootstrap this system, or, if the tape has been changed, to load a DEC system.

CATALOG (CAT) lists the names of saved files.

SYS reads a catalog from the tape presently mounted; there is no other action.

DUP - used to duplicate a system and all its subsystems. DUP was designed for a one tape installation. It reads a segment, halts for tape switching, on hitting Continue writes that segment, halts for tape switching, on Continue loops until all segments have been duplicated whereupon it bootstraps the new tape.

BIN - loads a standard binary loader for use with the TTY, then transfers control to it.

BSAVE < name > saves binary file in "\$" under an identifier name, in the user file area. This is the only command which may be used to move a binary file (See "SAVE")

RUN (RU) - run is used to call in a subsystem, specifying options and input files. The standard form is RUN sysname-param, file1, file 2, ..., file 15 . "sysname" is the name of the subsystem being used. This is either BIN, PAL, LISP, SNOBOL, or EDIT. Param is an optional octal number between 0000 and 7776 and is assumed zero if omitted. The list of files is treated by the subsystem as if it were one long file; up to fifteen files can be strung together (subsystems may have tighter limits).

EDITOR COMMANDS

LOAD (LO) name - loads the tape file designated by name into the working file.

SAVE (SA) name - saves the working file into the space on the user's part of the tape reserved for that name. If no space is reserved then the first available space is used.

DELETE (DE) name - deletes the file named from the catalog in core and on tape.

PUNCH (PU) n-m - same as list below except line numbers are not listed. Before and after the file the editor will place 18 inches of leader or trailer; to terminate this prematurely type exactly one character (any one) when sufficient leader or trailer has been generated. Do

not type characters while the file is being punched or it will be prematurely terminated.

LIST (LI) \square n-m - there are several options to control listing part or all of the working file.

1. LIST (LI) - lists entire working file.
2. LIST \square N (LI N) - lists line number n, if it exists, otherwise lists nothing.
3. LIST-N (LI-N) - list through line N from the beginning of the working file.
4. LIST \square N- (LI N-) - list from line N to the end of the working file.
5. LIST \square N-M (LI N-M) - list from line N through line M. During a listing any input from the teletype will terminate the listing at the end of the line being typed. A character used in this way is lost and will not appear in the listing.

SEQUENCE \square N \square M (SEQ \square N \square M) - initiates automatic line numbering from n in steps of m. This mode is terminated by an EOT (control D) character. If M is omitted, 1 is assumed.

RESEQUENCE \square N \square M (RE \square N \square M) - resequences the working file from n in steps of m.

ZONK - puts the entire working file on one line, using the first line number in the present working file; this is a space saver. Line numbers take up two words per number. If this is used one must change the ENTIRE line in order to make any changes.

SCRATCH (SCR)- initializes working file to a null file .

ECHO (EC) - normally the system echoes, types out all commands given it. Each time ECHO is typed the echo switch is reversed, from on to off and back. This is used to keep the listing free from monitor commands.

RADIX (RA)- like ECHO this command controls a switch. Normally the system uses decimal numbers. By typing RADIX the user may cause the system to employ octal numbers.

LAST (LA) - returns the line number of the last statement in the working file.

LEFT (LE) - returns the number of unused words in the file in the present radix (initially 2044_{10}). There are two symbolic characters per word.

When the system types:

WHAT - it means that the working file is full or syntax error occurred in command.

NF - it means that the file name used in a command is not in the catalog.

TOO BIG - means binary file in THE BIN is too large to be BSAVE'd.

MISCELLANEOUS SYSTEM NOTES

This section of this manual describes miscellaneous features of the RL Monitor which are either new or which are not adequately described in any other publication.

1. The system prints a dot (.) whenever it is ready for more input (assuming the echo switch is on).
2. Whenever a scratch is done, the system rings the teletype bell.
3. Whenever auto-sequence mode is terminated, either because control D is typed, or because a backslash is encountered in the text, the echo switch is turned on.
4. The system, EDIT, performs a scratch before it starts.
5. When typing a line into the editor beginning with a statement number, the statement number should be followed by a space and then the text starts. This space is not part of the text, so if you want your text to start with a space, you must then type an additional space. When in auto-sequence mode, the system automatically types this space for you. If a non-digit immediately follows the line number (and is not a space) the space is automatically inserted into the listing. When in octal line-numbering mode, the characters 8 and 9 are considered to be non-digits. This space following the line number is useful only to make listings neater, but is not part of the source file and is not passed as text to any systems.
6. The starting address for the RL Monitor is location 400. The system automatically does a scratch when it starts. The restarting address is location 601. You may branch there at any time manually, by hitting load address (with 601 in the switch register) and then hitting continue (not start). Your active work space is saved.
7. The editor ignores rub-outs and line feeds. It changes tabs to spaces.
8. You may hit the key, control R, at almost any time and this will cause the tape to rewind to its beginning after the current tape operation has been completed. Commands may still be entered while this rewind operation is in progress. If a new tape command is entered while it is rewinding, the new operation has precedence and the tape will not be rewound.
9. Hitting the key, control T, from a system from a tape which has been blessed by Mario DeNobili, will cause the tape to rewind and then unload. If another command using the tape is entered before the tape is unloaded, the new operation takes precedence. If the tape has not been blessed by Mario, hitting control T has no noticeable affect. All tapes sent by the P?S to DECUS have been blessed by Mr. DeNobili. If you want a tape blessed, you may pay a personal visit to the P?S secret headquarters at the Polytechnic Institute of Brooklyn, or mail it to Mario, c/o Box D at the Polytechnic Institute of Brooklyn, 333 Jay Street, Brooklyn, New York 11201. Please enclose return postage.
10. An empty file should not be zonked.
11. The command, BSAVE, destroys the active work space since it first transfers the BIN to core and then to the specified file on the tape.

12. Comments can be inserted on most monitor commands by following the command with a space and then typing the comment. For example, the following commands are legal:

```
CATALOG SMITH'S TAPE  
LIST 1000-5000 FILE FOO
```

13. The largest legal line number is $4094_{10} = 7776_8$. Larger numbers are taken modulo 4096 and may cause errors in your file of irreparable damage. During a listing, leading zeros of statement numbers are not printed when in decimal radix. When in octal radix, all statement numbers are four octal digits long. The largest legal parameter which may be passed to a system is 7776 . This must be expressed in octal.

14. The parameter \emptyset is not legal for the system, BIN. Thus your program cannot have \emptyset as its starting address. The command `RUN BIN= \emptyset` will load your program but execution will not start.

15. The parameter part of a RUN command may appear anywhere that a comma could appear. (It must start with an equal sign.) Thus the following commands are all legal:

```
RUN BIN=3400,BPROG,BPROG2  
RUN BIN,BPROG,BPROG2=3400  
RUN BIN,BPROG=3400,BPROG2
```

and all perform the same function. If two or more parameters are included, only the right-most one is passed to the system. Other punctuation characters are legal in this command (such as a period instead of a comma) but are not recommended.

16. A legal source file name consists of from 1 to 6 characters. Each character may be either a letter or a digit or certain other special characters, such as brackets.

17. When in auto-sequence mode, if you use the back-arrow character to erase characters on the line, you may erase all or part of the statement number. The new statement number is then the one that is effective. The original statement number is not used and no changes are made to the line containing this statement number. After that line is completed, sequencing continues from the number after the original statement number, before changed. If the entire statement number is deleted, a monitor command may be entered. After completion of this command, auto-sequencing is continued, if possible. If you attempt to erase more characters than are on the line, the system types a carriage return line feed and reprints the original statement number.

18. The command, `HR m n`, has recently been added to the system but is not supported by the P?S. `m` and `n` are two numbers which perform the same function as in the `SEQUENCE` command. `HR` stands for high speed reader. This command initiates auto-sequencing, from `m` in steps of `n`, and further source input is then taken from the high speed reader, if the system has one. If the system has no high speed reader, or if it is off or not ready, the system goes back to the usual teletype input (low speed reader). It stays in sequence mode unless input directed otherwise. When input tape is done, further input is taken from the teletype.

Summary of Commands

The first two characters of the keyword are sufficient to identify the command.

Bin - Loads a (paper tape) binary loader and transfers to it.

BSave Name - Saves the binary program currently in the bin in a binary file named Name. If Name is already on tape, new version stores over old; otherwise a new file is created with appropriate catalog entry.

CAlog - Lists the names of the saved files in the catalog. No indication is given whether the file is a source file or binary file.

DElete Name - Deletes the catalog entry for the file named Name. This space is thus made available for future saved files. No compression occurs.

DUplicate - Used to duplicate the RL Monitor System and all systems onto a new (formatted) tape. It reads a segment, halts for tape switching; on hitting continue, writes that segment, halts for tape switching; reads the next segment, etc., until all systems have been duplicated, whereupon it bootstraps onto the new tape. No user saved files are copied and the catalog on the new tape is cleared.

ECho - Flips the echo switch (initially set to echo).

HAlt - Halts the monitor; may be restarted by hitting continue. Used under EDIT control to give operator time to change tapes.

LAst - Returns the line number of the last statement in the current work space.

LEft - Returns the number of words left unused in the current work space. It is initially 2044₁₀. One word stores two characters.

Llst [n] [-m] - Lists lines n through m inclusive in the current work space.

If -m is omitted, it is assumed that m=n.

If m is omitted, it is assumed to be the last line in the file.

If n is omitted, it is assumed to be the first line in the file.

LOad Name - Loads the saved file named Name into the work space, destroying the previous contents.

MOnitor - Transfers to location 7600, i.e. bootstraps.

PUnch [n] [-m] - Same as Llst except that line numbers are not printed, and listing is preceded and followed by 18 inches of leader/trailer.

Note: Name may be any sequence of letters or digits. Only the first six characters are significant. Name may also be \$. \$ cannot be deleted and its contents may be destroyed by certain systems since the \$ file is overlapped by the bin.

RAdix - Flips the radix switch. (Choice of octal or decimal line numbering and referencing. It is initially decimal.)

REsequence *n* [*m*] - Resequences the entire current work space from *n* in steps of *m*. If *m* is omitted, it is assumed to be 1. Resequencing must not cause new statement numbers to be larger than 4094.

RUn System [= param,] [File1,File2,...] - **RU**n is used to execute a system, specifying options and input files if any. System is the name of the system to be executed; only the first two characters are significant. param is an optional octal number from 0 to 7777 which is passed to the system; if omitted, it is assumed to be 0. The list of files are strung together and passed to the system if it requests them. A maximum of 15₁₀ files may be so strung together.

SAve Name - Saves the current work space on the tape as a user file with the name Name. If name is already on tape, the new version stores over the old, otherwise a new file is created with appropriate catalog entry. The contents of the current work space are not affected.

SCatch - Clears the current work space, i.e. replaces it by an empty file.

SEquence *n* [*m*] - Initiates automatic line sequential numbering from *n* in steps of *m*. If *m* is omitted, 1 is assumed. Hitting ALT MODE deletes any changes in the current line and allows you to retype it. This mode is ended by hitting Control D (an EOT character). The last line number printed by the sequencing is not affected. This mode is ended during an EDIT by a backward slash (\).

SYstem - Rereads the catalog and system catalog into core. Used to tell the monitor that a new tape has been mounted. The current work space is not affected.

ZOnk - Permanently removes the line numbers from the file in the current work space. Used to save space for completed programs.

Notes: Listing may be stopped at any time by hitting any character.

ALT MODE or Control D may be hit to remove a line that has been typed (before carriage return has been hit). No change is made to the file in the current work space.

Back-arrows (←) may be used to delete the last character typed.

m is not optional in the **RE**sequence command. Only a simplified structure for the **RU**n command is shown.

Error Messages: WHAT? Syntax of monitor command is incorrect or file in **RUN** command cannot be found.

NF Not Found. Name in **LOAD** or **DELETE** command cannot be found.

TOO BIG Binary file is too large to be **B**saved.

RL MONITOR SYSTEMS PROGRAMMERS' GUIDE

TABLE OF CONTENTS

February 1971

	PAGE
I Introduction and acknowledgement	16
II Security Classification	16
III Format of RL Monitor Tape	16
IV Changes from the original system	16
V The user catalog	17
VI Systems catalog	17
VII Mechanics of the RUN command	18
VIII Dump of blocks 0, 20, and 21	19
IX Dump of Blocks 15 and 16 of a typical tape	20
X Dump of block 17 on a typical tape	21
XI Core allocation for RL Monitor	22
XII Format of active work space	22
XIII Example of a typical work space	23
XIV Format of an RL binary file	25
XV Example of a binary file	26
XVI Format of a zonked file	27
XVII Protecting a file	27
XVIII Deleting a system	27
XIX Adding a system	28
XX Standard Packed Format	28
XXI Auto starting	29
XXII Generating the monitor from scratch	29

RL Monitor Systems Programmer's Guide

I. Introduction and acknowledgement.

The RL Monitor (MS/8) is a multi-purpose monitor system for a PDP-8 system with 4K of core, one teletype, and one Dectape.

The P?S wishes to thank that famed, omnipotent programmer, Mr. Richard Lary of the WCFMPG, for writing this system.

II. Security Classification.

This guide may not be read by any person with a P?S security rating lower than class 3.

III. Format of RL Monitor Tape.

<u>Block(s)</u>	<u>Contents or use</u>
Ø	Bootstrap and tape routines
1-14	core image of monitor
15-16	user source catalog
17	systems catalog
20	core image of Dec binary loader
21	core image of system duplicator
22-57	binary file (BIN)
4Ø-57	"\$" user file
6Ø-27Ø1	available for systems and user files

Note that what was originally called "\$" is now called the BIN and is stored on blocks 40-57. The "\$" in the RUN BIN,\$,file1,...,fileN refers to the original meaning and refers to the binary file starting at block 22. RUN BIN,\$ is equivalent to RUN BIN. A bsave transfers the BIN to a user file. The "\$" in LOAD \$ and SAVE \$ refer to blocks 40-57. This file should be used for temporary storage and quick access time but users should be cautioned that large binary files produced by PAL and loaded into the BIN might overwrite blocks 40-57. A BIN file longer than 20 blocks cannot be bsaved since a user file can be at most 20 blocks long. An attempt to do so produces the error message "TOO BIG"; the binary file is not written, but an erroneous catalog entry is made which should be deleted.

IV. Changes from the original system.

- (a) The \$ notation is slightly different as explained above.
- (b) The ?? error message has been replaced by WHAT?
- (c) Several minor changes have been made in the ECHO and PRINT routines in order to allow the monitor to be easily modified to print slower. * To get the monitor to print 20% slower, core location 2174 should be changed from a NOP (7000) to a JMP .-1 (5373). This can be zapped on the tape if required. Its location is relative address 174 on block 11.
- (d) The period and bell have been added.
- (e) Internal handling of certain characters has changed.
- (f) The bootstrap has been slightly modified.
- (g) The HR command (Kluge) has been added.
- (h) The unload feature has been added to tapes.

* See top of next page

* Note: The following patches in core can also be made if you want to use the high speed punch: In the RL monitor, with echo off, change location 2145 from 6041 to a 6021 and change location 2175 from a 6046 to a 6026. In PAL, when it is running, after leader starts coming out on TTY, hit stop, change location 365 from a 6041 to a 6021 and change location 367 from a 6046 to a 6026. Restart PAL at location 367. (Restart monitor at location 2175).

V. The user catalog.

This is the catalog of the user's saved files which he gets by the CAT command. Each entry is 3 words long, containing the six characters of the name of the saved file in the standard packed format (2 characters per word). There is room in the catalog and on the tape for $124_8 = 84_{10}$ saved files.

Each user saved file occupies exactly 20 blocks on the tape and they occur consecutively. The last location in block 16 (relative address 177) gives the block number of the first block of the first user file.

The beginning of user saved files should not be after block 200.

The end of the catalog is denoted by an entry whose first word is 0.

A file which has been deleted from the catalog has its first word in the catalog entry replaced by a packed "??" (7777), but the source on the tape is not lost until another file is saved over it. Thus an important file which is accidentally deleted can be recovered by zapping the "??" in the first word of the correct catalog entry and changing it to some 2-letter packed ASCII code such as AA (0101).

A deleted catalog entry prints as a linefeed with the command CAT.

If x denotes the beginning of the saved files (the contents of the last word in block 16), then the n th saved file begins at block $x + (n-1)*20$ on the tape, and the catalog entry starts at relative location $(n-1)*3$ relative to block 15.

VI. Systems catalog.

In addition to user saved files, RL monitor tapes may contain programs known as systems which can not be referenced by the user except through the RUN command. Almost all tapes contain PAL, BIN, and EDIT as systems and many tapes contain other systems also, such as SNOBOL, DUMP, etc. Some tapes (called systems tapes) contain systems programs almost exclusively and little or no saved files. These tapes contain frequently used programs and demonstration programs stored as RL systems.

RL systems may be stored anywhere on the tape in consecutive blocks but usually are all located before the start of the saved files so that the user can not write over them. They are stored as core images beginning with page 0 of core. One block on the tape corresponds to one page of core since relative address 200 of each block is not used. The systems catalog in block 17 contains pointers to the beginning of each system on the tape. Unlike the saved file catalog, each entry specifies the length and location of each system program; thus the order of the catalog entries need not be the same as the physical order of the locations of the systems programs on the tape. Each entry in the systems catalog is 4 words long. The end of the systems catalog is denoted by an entry whose first word is 0. There may not be gaps in the systems catalog. If a system is removed from a tape, the systems catalog entries should be compressed.

VI. Systems Catalog (continued)

Each 4-word catalog entry is comprised of the following:

- word 1: 2-character name of the system represented in the usual packed format. In the RUN command, only the first two letters of the system name are significant. This is different from names of user saved files in which 6 characters are significant.
- word 2: Block number of beginning of program on tape.
- word 3: Negative of the length of the program in blocks.
(This number would be 7773 for a program which took up 5 blocks.)
- word 4: Absolute starting address of the program.

VII. Mechanics of the RUN command.

The RUN command in the RL monitor has the general form

RUN system=param,file₁,file₂,...,file_n

When this command is encountered, the first two letters of system are looked up in the systems catalog. If they are not found, the WHAT? error message results. If the system is found, it is loaded directly into core (provided the syntax of the rest of the RUN command is OK). The param, if it is included, is placed in absolute core location 7756 to possibly be later used by the system. If param is not specified, location 7756 is loaded with a \emptyset . The n files specified are looked up in the saved file catalog. If any of them are not found, the error message, WHAT?, results and control passes back to the monitor without ever having the system loaded. n must be less than or equal to 15 (base 10). If all the files are found, the block numbers of their beginning are computed and stored in locations 7757 and up consecutively. That is, the block number of the start of file₁ is put in location 7757, the block number of the start of file₂ is put in location 7760, etc. After all the block numbers for all the n specified files are stored in core, a \emptyset is stored in the next core location. Thus if there were 15₁₀=17₈ files specified, file₁₅'s starting block number would be placed in absolute core location 7775 and location 7776 would be zeroed. Note that it is solely up to the system program to make use of these block numbers. Thus PAL-III had to be modified when it was turned into an RL system, so that it looked at location 7756 for information about which passes to perform, so that it looked at the following locations to determine where on the tape its input was coming from, etc. For this reason, some systems may not allow files to be passed to it, or may not require a parameter, or may allow files, but may not be able to handle a maximum of 15 files. Indeed, the system, BIN, allows only a maximum of 5 files to be passed to it. It ignores the rest (if any). After the system is loaded into core and the file information stored, the starting address is obtained from the systems catalog and control is transferred directly to this location.

VIII. Dump of blocks 0, 20, and 21.

BLOCK = 0000

0000	6224	6774	1221	4213	1222	3355	1223	4213
0010	0000	0000	0000	7624	6766	3354	6771	5216
0020	5613	0600	7577	0220	1322	6766	6774	1317
0030	3355	4325	6772	7006	7710	5311	6772	7710
0040	5260	6761	7006	7006	7600	1320	7041	1316
0050	7450	5266	7041	7420	7001	7620	1320	5276
0060	6761	0323	1324	6764	6772	7402	7220	5276
0070	1315	6764	7240	3355	1302	3325	6734	1244
0100	3354	5326	7703	6772	7710	5311	2314	5276
0110	5731	1323	1321	5276	7760	0000	0001	7720
0120	0000	0400	0614	0200	0000	0000	6771	5326
0130	5725	0400	3014	1623	0000	1142	1423	2203
0140	1420	1146	2214	1623	2203	1161	1416	2325
0150	1500	1175	1420	2325	7777	7756	1714	1116
0160	0516	2510	1720	0314	1617	2550	3016	0115
0170	0500	1200	1624	1520	6100	1260	1624	1520

```
*2T7<JQ"KJR[-JS"K*
*@@@@@>I76[,79*N*
*.KF@=?BPKR767<K0*
*[-#U7:8F?H+I7:7H*
**0718F8F>@KPE!KN*
*<(*68!<PEA>PKQ*>*
*71CSKI747:<b>P*>*
*KM74: [-KB[U74J$*
*[,+V?C7:7H*0SL*>*
*/YKSKQ*>?4@X@A?P*
*@@DEFLB@B@79+V*
*/UD@XLNSRCI"LSRC*
*LPI&RLNSRCIILNSU*
*M@I=LPSU7???.OLIN*
*ENUHOPCLNOU(XNAM*
*E@J@NTMP1@J@NTMP*
```

BLOCK = 0020

0000	6224	6774	1221	4213	1222	3355	1223	4213
0010	0000	0000	0000	0000	6766	3354	6771	5216
0020	5613	0600	7577	0220	0200	7703	7100	4344
0030	4262	7041	1325	7402	6032	4272	5235	3325
0040	1326	3376	4314	3271	4272	5230	4262	7420
0050	5256	3324	1376	1271	1325	5237	3724	2324
0060	7600	5252	7631	1376	7106	7006	7006	1271
0070	5662	0000	7636	3327	4314	1233	7750	5303
0100	2327	7040	5273	1327	7640	5274	1326	0323
0110	1260	7710	2272	5672	7675	6031	5315	6036
0120	3326	1326	5714	0300	0142	0000	0200	0000
0130	7600	5321	0000	1351	7106	7106	7106	1352
0140	5732	7401	5252	0006	7630	5252	0000	6201
0150	0000	0000	0000	0173	7755	7755	6032	6031
0160	5357	6036	7106	7006	7510	5357	7006	6031
0170	5367	6034	7420	3776	3376	5356	0000	5234

```
*2T7<JQ"KJR[-JS"K*
*@@@@@>76[,79*N*
*.KF@=?BPB@?C@#S*
*"28!KU<B0Z":*][U*
*KV[>#LZ9":*X"2<P*
**.[TK>J9KU*←←TST*
*>@**>YK>9F8F8FJ@*
*.2@>↑[W#LJ[?(+C*
*SWE*;KW>*<KVCS*
*J0?HR:.:>=0Y+M0↑*
*[VKV/LC@A"@@B@C@*
*>@+Q@C@K)9F0F0FK**
*/Z<A**@F>X**@2A*
*@@@@@A;?-?-0Z0Y*
*+/0↑0F8F=H+/8F0Y*
*+70\<P<=>[>+.@C*\*
```

BLOCK = 0021

0000	0000	4335	1200	3206	7201	5210	0000	4335
0010	1272	3271	7240	4331	3200	1327	6756	5774
0020	4331	3320	4331	3323	1324	3355	4345	6772
0030	7006	7710	5315	6772	7710	5255	6761	7006
0040	7006	7600	1325	7041	1323	7450	5267	7041
0050	7420	7001	7620	1326	5302	6761	0330	1344
0060	6764	6772	7402	7346	1206	3206	5210	7620
0070	5302	5000	5273	1322	1321	6764	1200	3355
0100	1306	3345	6764	1241	3354	5346	7707	6772
0110	7710	5255	2320	5302	5606	1330	1326	5302
0120	0000	0030	0020	0000	7725	0000	0400	0614
0130	0200	0000	1606	2206	5731	0000	6761	6761
0140	0330	7640	5336	5735	0002	0000	6771	5346
0150	5745	3237	1162	7000	7755	7755	0100	0122
0160	4200	7400	7777	0010	5761	5360	4224	2327
0170	4224	1356	7650	7402	5756	0400	7600	5360

```
*@@#]J@ZF:A*H@#]*
*J:Z9: #YZ@KW767<*
*#Y[P#Y[SKT[-#Z7:*
*8F?H+M7:7H*-718F*
*8F>@KU@!KS<(*78!*
*<P8A>PKV+B71CXX$*
*747:<B:&JFZF*H>P*
*+B(@*;KRKQ74J@[-*
*KF[774J![,+&?G7:*
*?H*-SP+B.FKXKV+B*
*@@@X@P@?U@D@FL*
*B@@@NFRF/Y@7171*
*C>+↑/]@B@79+&*
*/Z7+12@?7-A@AR*
*"@<@??@H/1+0"TSW*
*"TK.>(<B/.D@>@+2*
```


IX. Dump of Blocks 15 and 16 of a typical tape.

BLOCK = 0015

0000	2014	0107	1700	0220	1401	0717	0623	0000	*PLAGO@BPLAGOFSS@*
0010	0000	2405	2424	0000	0114	0116	2400	0320	*@@TETT@ALANT@CP*
0020	2362	0000	0320	2361	0000	2001	2514	6061	*S2@@CPS1@PAUL01*
0030	0206	1417	0124	7777	1520	0000	0520	2215	*BFLOAT??MP@EPRM*
0040	0516	2401	0000	0000	0320	2363	0000	0320	*ENTA@@@CPS3@CP*
0050	2364	0000	0320	2311	1700	2025	1603	1000	*S4@@CPS10@PUNCH@*
0060	0405	0324	0120	2205	0104	0000	1011	2320	*DECTAPREAD@HISP*
0070	2204	2205	0104	6100	1011	2320	2516	2001	*RDREAD1@HISPUNPA*
0100	2200	0000	1722	0700	0000	0600	0000	0000	*R@@@ORG@@@F@@@@@*
0110	0417	1414	0122	7777	1520	1722	0000	0000	*DOLLAR??MPOR@@@@*
0120	0000	0000	0000	0000	0000	0000	0000	0000	*@@@@@@@@@@@@@@@@@*
0130	0000	0000	0000	0000	0000	0000	0000	0000	*@@@@@@@@@@@@@@@@@*
0140	0000	0000	0000	0000	0000	0000	0000	0000	*@@@@@@@@@@@@@@@@@*
0150	0000	0000	0000	0000	0000	0000	0000	0000	*@@@@@@@@@@@@@@@@@*
0160	0000	0000	0000	0000	0000	0000	0000	0000	*@@@@@@@@@@@@@@@@@*
0170	0000	0000	0000	0000	0000	0000	0000	0000	*@@@@@@@@@@@@@@@@@*

BLOCK = 0016

0000	0000	0000	0000	0000	0000	0000	0000	0000	*@@@@@@@@@@@@@@@@@*
0010	0000	0000	0000	0000	0000	0000	0000	0000	*@@@@@@@@@@@@@@@@@*
0020	0000	0000	0000	0000	0000	0000	0000	0000	*@@@@@@@@@@@@@@@@@*
0030	0000	0000	0000	0000	0000	0000	0000	0000	*@@@@@@@@@@@@@@@@@*
0040	0000	0000	0000	0000	0000	0000	0000	0000	*@@@@@@@@@@@@@@@@@*
0050	0000	0000	0000	0000	0000	0000	0000	0000	*@@@@@@@@@@@@@@@@@*
0060	0000	0000	0000	0000	0000	0000	0000	0000	*@@@@@@@@@@@@@@@@@*
0070	0000	0000	0000	0000	0000	0000	0000	0000	*@@@@@@@@@@@@@@@@@*
0100	0000	0000	0000	0000	0000	0000	0000	0000	*@@@@@@@@@@@@@@@@@*
0110	0000	0000	0000	0000	0000	0000	0000	0000	*@@@@@@@@@@@@@@@@@*
0120	0000	0000	0000	0000	0000	0000	0000	0000	*@@@@@@@@@@@@@@@@@*
0130	0000	0000	0000	0000	0000	0000	0000	0000	*@@@@@@@@@@@@@@@@@*
0140	0000	0000	0000	0000	0000	0000	0000	0000	*@@@@@@@@@@@@@@@@@*
0150	0000	0000	0000	0000	0000	0000	0000	0000	*@@@@@@@@@@@@@@@@@*
0160	0000	0000	0000	0000	0000	0000	0000	0000	*@@@@@@@@@@@@@@@@@*
0170	0000	0000	0000	0000	0000	0000	0000	0300	*@@@@@@@@@@@@@@@@@*

This dump illustrates a user's catalog containing files named, PLAGO, BPLAGO, FS, TETT, ALANT, CPS2, CPS1, PAUL01, BFLOAT, EPRMEN, TA, CPS3, CPS4, CPS10, PUNCH, DECTAP, READ, HISPUN, READ1, HISPUN, PAR, ORG, F, and DOLLAR. It also shows the remnants of two deleted files, one between BFLOAT and EPRMEN, and one after DOLLAR. The user's files begin in block 300 as given by the last word of block 16. (Such a high starting block is not recommended.)

X. Dump of block 17 on a typical tape.

BLOCK = 0017

0000	2001	0223	7761	3400	0211	0220	7776	0240	*PABS?I\@BIBP?>B *
0010	0504	0221	7776	0000	2316	0242	7754	2400	*EDBQ?>@@SNB"? ,I@*
0020	0425	0266	7774	0200	0123	0272	7776	0200	*DUB6?<B@ASB: ?>B@*
0030	0000	0000	0000	0000	0000	0000	0000	0000	*@@@@@@@@@@@@@@@@@@@@*
0040	0000	0000	0000	0000	0000	0000	0000	0000	*@@@@@@@@@@@@@@@@@@@@*
0050	0000	0000	0000	0000	0000	0000	0000	0000	*@@@@@@@@@@@@@@@@@@@@*
0060	0000	0000	0000	0000	0000	0000	0000	0000	*@@@@@@@@@@@@@@@@@@@@*
0070	0000	0000	0000	0000	0000	0000	0000	0000	*@@@@@@@@@@@@@@@@@@@@*
0100	0000	0000	0000	0000	0000	0000	0000	0000	*@@@@@@@@@@@@@@@@@@@@*
0110	0000	0000	0000	0000	0000	0000	0000	0000	*@@@@@@@@@@@@@@@@@@@@*
0120	0000	0000	0000	0000	0000	0000	0000	0000	*@@@@@@@@@@@@@@@@@@@@*
0130	0000	0000	0000	0000	0000	0000	0000	0000	*@@@@@@@@@@@@@@@@@@@@*
0140	0000	0000	0000	0000	0000	0000	0000	0000	*@@@@@@@@@@@@@@@@@@@@*
0150	0000	0000	0000	0000	0000	0000	0000	0000	*@@@@@@@@@@@@@@@@@@@@*
0160	0000	0000	0000	0000	0000	0000	0000	0000	*@@@@@@@@@@@@@@@@@@@@*
0170	0000	0000	0000	0000	0000	0000	0000	0000	*@@@@@@@@@@@@@@@@@@@@*

This dump shows the system catalog on a typical tape. The information contained in this catalog is summarized below:

<u>System Name</u>	<u>Block number of start</u>	<u>length</u>	<u>Starting address</u>
PA1	223	17	3400
BIn	220	2	240
EDit	221	2	0
SNobol	242	24	2400
DUmp	266	4	200
ASsembler	272	2	200

Comments:

1. The systems, BIn and EDIT, share block 221 on the tape. When running BIn, block 221 gets loaded into page 200, and when running EDIT, block 221 gets loaded into page 0.
2. The systems on this tape occur so high up on the tape because the system, ASsembler, uses blocks 60 through 217 for data storage. On most tapes, the systems would start around block 61.

NOTE: The system Assembler is not currently available for distribution.

XI. Core allocation for RL Monitor

<u>Core locations</u>	<u>Contents or use</u>
0-2777	Monitor. Read in from blocks 1-14 on tape
3000-6777	20 pages used as current workspace
7000-7377	Resident user's file catalog, read in from blocks 15-16 on tape.
7400-7577	Resident systems catalog, read in from block 17 on tape.
7600-7777	Bootstrap and tape routines usually reside here, are read in from block 0 on tape. When executing the command BIN or DUF, this page of core will contain the appropriate routine, read in from the tape, block 20 or block 21 respectively.

During normal operation, location 7600 contains the start of a standard dectape bootstrap routine. Manual start at location 7600 (or execution of MONITOR command) can thus be used to bootstrap onto another RL tape or other types of tapes. When transferring to another RL tape, the SYSTEM command is usually preferred since it retains the user's current active workspace in core.

XII. Format of active workspace.

The user's current workspace, including source text and line numbers, is stored in the 20 pages in locations 3000 through 6777 inclusive. The source text starts at location 3000 and works up towards higher core; the corresponding sequence line numbers begin at 6775 and work down towards lower core. When these two meet, the workspace is full and further attempts to add additional text result in the error message, "WHAT?".

The source text is stored in the standard sixbit packed notation. The end of a source line is indicated by the back-arrow character, code 37. (This code can never appear as source, since back-arrows are used to delete typed characters. Carriage returns typed in are not saved; they merely indicate the termination of the line and are thus changed to code 37. Line feeds are ignored by the monitor.) Each source line begins in the left half of a word. If the previous line ended in the left half of a word, the right half is packed with a 0. The end of source text is indicated by a word of 0's.

XII. Format of active workspace. (continued)

Looking at the active workspace from the end (high core) down towards the beginning (low core), starting at location 6777, we find the following:

Location 6777 points to the absolute core location which is next available for a line number entry. (See below.)

Location 6776 points to the absolute core location which is next available for a source text line entry. (See above.)

Location 6777 thus always points to a word preceding a word of 7's, while location 6776 always points to a word of 0's.

Locations 6775 and down contain line number information for the current workspace. Each entry consists of two words. The first word (in the higher core location of the two) contains the actual line number, which may be anything (in octal) from 0 to 7776 inclusive. The second word (in the lower core location) contains the absolute core address of the beginning of the source line whose statement number is being referred to by this entry. The line number entries are terminated by an entry whose first word is 7777. The user may not use 7777 (octal) or 4095 (decimal) as a line number.

The user's line numbers are always taken modulo 4096 (decimal). Resequencing the file so that line numbers become larger than this will thus screw up the editor.

XIII. Example of a typical workspace.

Here is a typical workspace listing containing source text:

LIST

```
100 ABCDEF
110 GHI
120 JKL MNOP
130 QRST U V W X Y Z
140 QRS T U V W X YZ
150 0123456789
160 THIS IS LINE 160***
200 ABC
215 1+1=2
400 EIEIO
566 /////
1000 E=M*C2
2048 D=R*T IS AN IMPORTANT EQUATION
2049 IT GIVES YOU DISTANCE IN TERMS OF RATE AND TIME
3037 END OF PROGRAM
```


XIII. Example of a typical workspace. (continued)

Here is the same workspace with line numbers listed in octal:

RADIX
LIST

```
0144 ABCDEF
0156 GHI
0170 JKL MNOP
0202 QRST U V W X Y Z
0214 QRS T U V W X YZ
0226 0123456789
0240 THIS IS LINE 160***
0310 ABC
0327 1+1=2
0620 EIEIO
1066 //
1750 E=M*C2
4000 D=R*T IS AN IMPORTANT EQUATION
4001 IT GIVES YOU DISTANCE IN TERMS OF RATE AND TIME
5735 END OF PROGRAM
```

This workspace was saved on a tape and by examining its position in the user's catalog, it was determined that it occupied blocks 1173 to 1212 inclusive on the tape. Block 1212 containing the line number entries and block 1173 containing all the source entries are dumped below. Remember that to get absolute core locations, 3000 must be added to a word's relative address from the beginning of block 1173.

BLOCK = 1212

0000	3700	2401	0440	3002	1417	0313	5561	3700	*←@TAD XBLOCK-1←@*
0010	0403	0140	5653	6537	2401	0440	3002	1417	*DCA .+5←TAD XBLO*
0020	0313	3700	0403	0140	5653	6437	1215	2340	*CK←@DCA .+4←JMS *
0030	0427	2211	2405	3700	0225	0606	0522	3700	*DWRITE←@BUFFER←@*
0040	5561	3700	6037	1215	2040	1140	3020	2524	*-1←@0←JMP I XPUT*
0050	1124	3700	2001	0705	6275	5637	4437	0000	*IT←@PAGE2=.←\$←@@*
0060	0000	4437	0000	0107	5662	7777	5655	7777	*@@\$←@@AG.2??.-??*
0070	5657	7777	5656	5656	5652	5644	5643	5632	*./??.....*.\$#.Z*
0100	5642	5620	5640	5606	5634	5574	5626	5562	*."P. .F.\-<.V-2*
0110	5622	5550	5614	5536	5610	5524	5601	5512	*.R-(.L-↑.H-T.A-J*
0120	5574	5500	5565	5466	5560	5454	5556	5442	*-<-@-5,6-0,. -."*
0130	5552	5430	5545	5416	5541	5404	3156	7777	*-*,X-?,N-!,DY.??*
0140	3146	5735	3116	4001	3076	4000	3072	1750	*Y&/JYN AX> @X:0(*
0150	3056	1066	3063	0620	3060	0327	3056	0310	*XG6X3FPX0CWX.CH*
0160	3044	0240	3036	0226	3025	0214	3014	0202	*X\$B X↑BVXUBLXLEL*
0170	3005	0170	3004	0156	3000	0144	3156	6736	*XFAEXDA.X@A\$Y.7↑*

XIII. Example of a typical workspace. (continued)

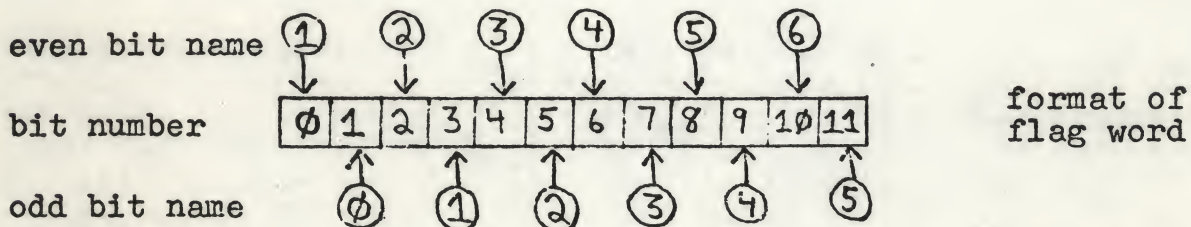
BLOCK = 1173

0000	0102	0304	0506	3700	0710	1137	1213	1440	*ABCDEF←@GHI←JKL *
0010	4040	1516	1720	3700	2122	2324	4025	4026	* MNOP←@QRST U V*
0020	4027	4030	4031	4032	3700	2122	2340	2440	* W X Y Z←@QRS T *
0030	2540	2640	2740	3040	3132	3700	6061	6263	*U V W X YZ←@0123*
0040	6465	6667	7071	3700	2410	1123	4011	2340	*456789←@THIS IS *
0050	1411	1505	4061	6660	5252	5237	0102	0337	*LINE 160***←ABC←*
0060	6153	6175	6237	0511	0511	1737	5757	5757	*1+1=2←EIEIO←////*
0070	5757	3700	0575	1552	0336	6237	0475	2252	*//←@E=M*C↑2←D=R**
0100	2440	1123	4001	1640	1115	2017	2224	0116	*T IS AN IMPORTAN*
0110	2440	0521	2501	2411	1716	3700	1124	4007	*T EQUATION←@IT G*
0120	1126	0523	4031	1725	4004	1123	2401	1603	*IVES YOU DISTANC*
0130	0540	1116	4024	0522	1523	4017	0640	2201	*E IN TERMS OF RA*
0140	2405	4001	1604	4024	1115	0537	0516	0440	*TE AND TIME←END *
0150	1726	4020	2217	0722	0115	3700	0000	0301	*OF PROGRAM←@@@CA*
0160	3156	0261	3160	0266	3162	0265	3164	0264	*Y.B1Y2B6Y2B5Y4B4*
0170	3166	0264	3170	0260	3172	0240	3174	0240	*Y6B4Y8B0Y:B Y←B *

XIV. Format of an RL binary file.

Binary files are not stored in the standard DEC binary format. They can not be loaded by the standard DEC binary loader stored on block 20 of the tape. Instead, they must be loaded by a special system program, named BIN, which is stored as a system on all RL system tapes.

The RL binary format consists of a sequence of 7 word entries. Each entry consists of one flag entry followed by 6 words of data. The flag entry consists of 12 bits. The odd and even bits serve different purposes, and will be referred to by the following notation:



As a notational convenience, the even bits are numbered from 1 to 6 and the odd bits are numbered from 0 to 5 as shown in the illustration above.

XIV. Format of an RL binary file. (continued)

The even bits (numbered 1 to 6) are used to specify which of the following 6 data words are to be loaded into core and which are merely origin settings, specifying where in core the following data words are to be loaded. Once an origin has been set, all future words (until the next origin setting) are loaded into consecutive core locations beginning at the specified origin. If even bit number *i* is a 1, this indicates that the corresponding data word which follows (data word *i*) is an origin setting. If even bit *i* is a 0, this indicates that the corresponding data word is data to be loaded into core.

The odd bits (numbered 0 to 5) are usually all 0's except for the last flag word. Each group in a binary file consists of 7 word entries except for the last group which consists of a flag word and from 0 to 5 data words. The exact number of data words in the last group are specified by the odd bits in the last flag word. To indicate that the last group contains only *i* data words (instead of the usual number of 6), odd bits numbered *i* through 5 are set to ones. Thus, to indicate that there are only 3 data words in the last group, odd bits 0, 1, and 2 are zeros and odd bits 3, 4, and 5 are ones in the last flag word.

When these 7-word groups are placed in blocks on a tape, 22 such groups are stored in each block, using relative addresses 0-175. Relative locations 176 and 177 are not used. If more than one block is required, the next group of 7 words begins at the beginning of the next block.

XV. Example of a binary file.

The following is a dump of a typical RL binary file. The flag words are underscored and the origin settings are enclosed in circles. The last 7-word group consists of only 5 data words.

BLOCK = 0022

0000	4000	<u>0200</u>	1111	2222	3333	4444	5555	<u>0200</u>	* @B@IIRRI[133--3@*
0010	6666	7777	<u>0300</u>	1234	2345	3456	<u>0200</u>	4567	*66??C@J\SZ\,beZ7*
0020	5678	<u>0247</u>	0000	1221	3443	<u>0000</u>	6666	7775	*.8J'@@JQ\#@65?=@*
0030	3344	2234	1234	4321	<u>4200</u>	<u>6540</u>	3336	<u>3200</u>	*[S\J\#Q"65 [1Z@*
0040	5674	2346	2222	<u>0001</u>	1234	1357	2460	0246	*.<S&R@AJ\K/TZb&*
0050	3571	0000	0411	0000	0000	0000	0000	0000	*19@@@I@@@@@@@@@*
0060	0224	2201	1661	0224	2201	1662	2010	1724	*BTRANIBTRAN2PHOT*
0070	1716	2017	1714	6100	2017	1714	6200	2617	*ONPOOLI@P00L2@V0*
0100	2011	2401	1405	2405	2261	1405	2405	2262	*PITALETERILETER2*
0110	2024	0424	2400	1102	1500	0000	2301	2605	*PTDTT@IEM@@@SAVE*
0120	1124	2401	2005	1124	2024	1417	0104	0223	*ITTAPEITPTLOADLS*
0130	0126	0500	0624	0120	0561	0624	0120	0562	*AVE@FTAPEIFTAPE2*
0140	1101	1524	1005	2316	1714	2324	2405	1520	*IAMTHESNOLSTTEMP*
0150	1722	0104	0400	0000	1525	2311	0300	1411	*ORADJ@@@MUSIC@LI*
0160	2302	0000	2316	1714	2400	0000	0000	0000	*S@@SNOL1@@@@@@@*
0170	0000	0000	0000	0000	0000	0000	0000	0000	*@@@@@@@@@@@@@@@@@*

XV. Example of a binary file. (continued)

The actual PAL source file which produced this binary file is shown to the right hand side of this page.

XVI. Format of a zonked file.

When a file is zonked, all the line number entries ^{/except one} at the end of the file are removed. These can not be recreated except by extensive zaps. If additional sequenced lines are added to the file, new line number entries will be created. Putting line numbers into a zonked file can probably best be done using SEQUENCE under the EDIT system. Zonking is not recommended except to save space for programs which are definitely complete, in which no future modifications are expected.

XVII. Protecting a file.

In some cases, a program may wish to use part of an RL system tape for data storage or for swapping purposes. If possible, he should use a section of the tape after all the systems but before the user's saved files. If there is not enough room there, he can use space in the midst of the user's files. However, to ensure that he (or other people) do not destroy the data that is stored here, he can protect the corresponding user file(s) so that they can not accidentally be deleted and/or another file stored over them. A convenient way to do this is to change the appropriate catalog entry name so that the monitor cannot reference it because of syntactic considerations. One way, is to zap in a catalog name with an illegal character or delimiter, or an embedded space. In this way, the user cannot LOAD, SAVE, BSAVE, or DELETE from or into this file. However, the only way to then remove this file when it is no longer needed, is to make another zap.

XVIII. Deleting a system.

A system can be deleted from an RL tape merely by zapping the corresponding catalog entry in the systems catalog and either moving all the following entries down, or replacing it by a dummy entry.

100 1111
110 2222
120 3333
130 4444
140 5555
150 6666
160 7777
170 *300
180 1234
190 2345
200 3456
210 4567
220 5670
230 *1247
240 0
250 1221
260 3443
270 6665
280 7775
290 3344
300 2234
310 1234
320 4321
330 *5540
340 3336
350 *3200
360 5674
370 2346
380 2222
390 1234
400 1357
410 2450
420 3246
430 3571
440 0

XIX. Adding a system.

The following steps (or their equivalent) must be made to add a new system to an RL system tape. Similar instructions apply to modifying an existing system.

1. Assemble the program and examine it to see how much core it uses and where a free page of core exists.
2. Examine the systems catalog to see where the existing systems are located and examine the last word of block 16 to see where the user's files start. Thus determine the (consecutive) blocks where you want to put the system. Remember that the system is stored in these blocks as a direct core image beginning with page 0.
3. Load the one-page DECTAPE routines onto your tape and add a short program at its beginning (about 6 locations should suffice) which writes the correct core pages onto the correct tape blocks. Give this file an origin setting which will not conflict with the space used by the system. At the conclusion, they should bootstrap.
4. Assemble these routines and save as a binary file.
5. Run BIN with the system and the DECTAPE routines. Start at the beginning of the DECTAPE routines.
6. This will put the system onto the tape and then bootstrap. Using DUMP, examine the newly formed blocks to see that everything went as planned; then zap the new system catalog entry into the systems catalog.

XX. Standard Packed Format.

The standard packed (sixbit) format used by the RL monitor is summarized by the following table.

	0	1	2	3	4	5	6	7
0	@	A	B	C	D	E	F	G
1	H	I	J	K	L	M	N	O
2	P	Q	R	S	T	U	V	W
3	X	Y	Z	[\]	↑	←
4	space	:	"	#	\$	%	&	'
5	()	*	+	,	-	.	/
6	Ø	1	2	3	4	5	6	7
7	8	9	:	;	<	=	>	?

To convert an ASCII character to the sixbit code, merely do an AND L77 .

To convert a character in sixbit code in the right half of a word, perform the following sequence, to change it to ASCII:

```
TAD L4Ø
AND L77
TAD L24Ø
```

Sixbit characters are usually packed two to a word.

XXI. Auto starting.

It is possible to make a binary file (not a subsystem) automatically start when loaded by including in the source somewhere a section of code consisting of one line containing the starting address and preceded by an origin setting of 7756. For example, if a program contains the code

```
*7756  
6000
```

when this program is assembled and then loaded by the command

```
RUN BIN
```

it will load and then automatically start from location 6000 even though no explicit starting address was given in the RUN BIN command. The starting address may not be 0 either explicitly or implicitly.

Caution: Unless you know what you're doing, your source file should not contain any other data or instructions to be loaded into page 7600, because that is where the binary loader resides while loading and if you load over it, serious results may happen.

XXII. Generating the monitor from scratch.

To generate the RL system from its numerous source files is a long and tedious procedure which we will not go into here in full detail. We will give a brief outline on how to do a partial gen after summarizing what each file does.

The source for the editor is stored in ten files called EDIT1 through EDIT10. The editor itself, without any peripherals, is stored in files EDIT1 through EDIT5. The purpose of the other files is as follows: EDIT6 consists of what gets loaded into block 0 of an RL tape. EDIT7 and EDIT8 contain the overlay to PAL-III. EDIT9 contains the source for the system duplicator (DUP) and the DEC binary loader (BIN) which get loaded onto blocks 20 and 21 of an RL tape. EDIT10 contains the source for the systems known as EDIT and BIN. The overlay to FOCAL (and its patches) are in the two files, FOCGEN and GENFOC, which are self-explanatory.

For the user who wishes to make a minor change in the editor, the following procedure is recommended: Make the source change with documentation in files EDIT1 through EDIT5. Assemble and load the object code into core, then place it as a core image (locations 0-2777) onto tape blocks 1-14. Then copy block 11 from a good tape onto block 11 of this tape. (Block 11 on the tape contains code required for the DUP command but its source is not in EDIT1-EDIT5. The corresponding core location, page 2000 is used as Lary's input buffer.) Making extensive changes in the editor is strongly discouraged by the P?S inasmuch as the P?S version of the RL Monitor will shortly be ready !

