



LDOS 5.1.4 Update - released 09/01/83

8970 N. 55th Street / P.O. Box 23956 / Milwaukee, WI 53223 / (414) 355-5454

---

LDOS 5.1.4 has had numerous internal changes made to most of the system files, and contains two new utilities: QFB and FED. To update your existing diskettes, we recommend the following procedure:

- 1) Make several mirror-image backups of your new master system diskette, then put it in a safe place. Never use your master diskette for any purpose other than the production of working copies.
- 2) Boot up with your new working system diskette. Place your old system diskette in drive one, and execute the following command:

```
BACKUP $:0 :1 (OLD)
```

This will transfer all of the updated files to your old diskette.

Note: Do not attempt to use any configuration file created by 5.1.3! Create a new configuration file under 5.1.4 if desired.

- 3) Model 1 owners should now place the new LDOSXTRA working diskette in drive one, and execute the following command:

```
BACKUP $:1 :0 (OLD)
```

Any files on your updated system disk that originally came from a LDOSXTRA diskette will now be updated.

- 4) If desired, the new utilities (QFB and FED) may now be copied to any diskette that has sufficient free space.

If your working diskettes are any older than the 5.1.1 release, you must move all your existing programs and files over to new working master system diskettes and formatted data diskettes. This will ensure your continued smooth operation.

Thank you for using LDOS. We hope that you will enjoy the new utility programs provided with the LDOS 5.1.4 release.

#### 5.1.4 CHANGES AND TECHNICAL INFORMATION

The 5.1.4 release of LDOS has all of the patches previously listed in the LDOS Quarterly newsletter applied, plus the following changes.

The floppy disk driver was adjusted to take care of running the Model 4 as a Model III at fast clock speeds, and to handle the new design of the FDC board on the Mod 4.

The KI/DVR program has the debounce adjusted to work better in the fast mode.

The method of allocating space on a disk has been changed, and will now search sequentially forward starting with the lower numbered tracks. If the old method of random allocation is desired, it may be patched back in with:

```
PATCH SYS8/SYS.SYSTEM (D00,FE=D5 CD 4E 44 D1 6C)
```

The DATE? prompt on powerup will now accept any delimiter character less than a "0" between the digits.

A problem in LCOMM that showed up during long sessions has been fixed.

A bug in PATCH/CMD was fixed, so X type patches that span sectors can now be fully YANKED.

Several small changes were made to format.

The DIR command has the A parameter ON. If this is not desirable, it may be changed with the following patch:

```
PATCH SYS6/SYS.SYSTEM (D05,D9=00 00)
```

On the MAX80, the delay for 5" drive startup has been fixed to properly give the 1 second and .5 second delay.

Two new programs were added to the disk, FED and QFB. Complete documentation is included with this update.

## F E D - T H E L D O S F I L E E D I T O R

FED is an all-purpose, screen oriented file editor to be used with the LDOS operating system. Its wide range of capabilities make it excellent for the advanced user, but its simplicity makes it easy to use for the novice. The editor supports both Model I and III, upper and lower case, and all drive types and sizes supported by LDOS. Some points need to be made concerning FED:

This is a file editor, NOT a file copier, text editor, or word processor. It is for displaying, printing, and modifying existing files. Fed works on a file level, not a track/sector level.

FED was not designed to repair damaged disks or recover lost files, but it could be used to do so by the experienced LDOS user.

You cannot create or extend files with FED, only modify existing ones.

FED is intended to run with the LDOS operating system only.

The following is a brief description of FED's capabilities:

- 1) Complete editing capabilities are supported, including Hexadecimal and ASCII modifying. Direct disk patching becomes a simple matter with FED. It is even possible to write machine language code directly to disk. Small changes in files can be made instantly. With FED, there is no need to read in a large source file and reassemble it just to change one character.
- 2) FED allows for record advancing, backspacing and positioning. You may page through a file quickly, either forward or backward. The user need not know any diskette information (density, number of sides, number of sectors per gran, etc.). The only thing that is required to use FED is knowledge of the proper filespec.
- 3) ASCII and Hex string searching can be performed, and a command exists which will allow you to position the cursor to the next occurrence of the search string. FED searches the entire file, not just the current edit record. It allows searching for upper/lower case ASCII strings (up to 30 characters in length), and Hex strings (up to 15 bytes in length). FED will retain a search string, so you can go to the next occurrence of that string from the currently displayed position in the file.
- 4) You will be allowed to locate a Hex load address in a load module format file, and calculate the load position of a specified byte. This feature will facilitate the inspection and editing of a load module file. Just type in the load address in question, and FED will position the display to that byte. Another extremely powerful feature is the reverse of the address location command. FED will calculate where in memory a specific byte pointed to by the cursor will load. With these two features it is possible to write machine language routines directly to disk. Direct patches are made quickly and easily. Even X-patches can be installed by the experienced programmer.

- 5) Complete listing of a file or individual record to a printer is supported. Many safeguards have been added to make it difficult to LOCK-UP the system if a printer is deselected, out of paper, etc.
- 6) FED includes a 256 byte display mode, and an extended 128 byte display. Editing utilities in the past allowed for 256 byte displays only. By using this format exclusively, the variations of an ASCII/HEX display are limited. But by having a 128 character display mode, the extra space makes it more visually appealing. The filespec, drivespec, record number, input & output can be displayed horizontally instead of vertically.

Here is a sample display of the 256 byte mode:

ASCII representation	Hexadecimal representation	Current Record
! .h . = . U X . S @ . . !	00> 21D8 6811 003D CD55 58ED 5320 4006 1721	0 F
. . @ . . . Z x . . > . . 3	10> ED60 CD40 00DA 945A 78B7 2005 3E13 C333	0 E
. . X . . . D ! . b .	20> 60CD 2058 11ED 60CD 1C44 20F0 210D 6206	0 D
. . \$ D . 3 ` . . 0 * . R w	30> 00CD 2444 C233 603A F360 C630 2A15 5277	D /
. K . . . C . R . . . C . R	40> ED4B F960 0BED 4313 5201 0000 ED43 0F52	C
! . a " . R > . . 3 . . U U . .	50> 210D 6122 0A52 3E1C CD33 00CD 5555 C93A	M
. R . . . Z . . . Z . . . B	60> 0152 B7CC 105A FE04 D410 5A11 ED60 CD42	D
D . 3 ` . . . 6 D . 3 ` ! . b	70> 44C2 3360 11ED 60CD 3644 C233 6021 0D62	:
. . [ . . . . . . . . . . ( U .	80> 110D 6101 0001 EDB0 CDE3 5BC9 CD28 55CD	5 - Drive #
. . ] . . K . R . . . . . B D .	90> DE5D C9ED 4B0F 52C5 11ED 60D5 CD42 44C2	
3 ` . 6 D . 3 ` ! . b . . a . .	A0> 3360 CD36 44C2 3360 210D 6211 0D61 0100	
. . . . . * . R . . B . . . B D	B0> 01ED B0D1 C12A 1352 B7ED 42C8 03CD 4244	
. 3 ` . 6 . . . U D . 3 ` . G :	C0> C233 60CD 3601 028E 5544 C233 60C9 473A	
. R . ( . ! . ? " @ . ! . = 6	D0> 0E52 B728 0721 CA3F 2220 40C9 21BD 3D36	
. # 6 . . @ . . 6 . . . 6 . + 6	E0> 8C23 36AC 1140 0019 36AA 10FB 3683 2B36	>82
. ! . = " @ . . : . R . ! . =	F0> 8321 FD3D 2220 40C9 3A0E 52B7 C021 BD3D	C:R

Index
Command

Here is a sample display of the 128 byte mode:

.K . . . C . R . . . C . R ! . a " . R > . . 3 . . U U . . : . R . . . Z . . . Z . . . B D . 3 ` . . . 6 D . 3 ` ! . b - ASCII  
 . . [ . . . . . . . . . . ( U . . ] . . K . R . . . . . B D . 3 ` . 6 D . 3 ` ! . b . . a . . . . . \* . R . . B . . . B D - Rep.

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
40>	ED	4B	F9	60	0B	ED	43	13	52	01	00	00	ED	43	0F	52
50>	21	0D	61	22	0A	52	3E	1C	CD	33	00	CD	55	55	C9	3A
60>	01	52	B7	CC	10	5A	FE	04	D4	10	5A	11	ED	60	CD	42
70>	44	C2	33	60	11	ED	60	CD	36	44	C2	33	60	21	0D	62
80>	11	0D	61	01	00	01	ED	B0	CD	E3	5B	C9	CD	28	55	CD
90>	DE	5D	C9	ED	4B	0F	52	C5	11	ED	60	D5	CD	42	44	C2
A0>	33	60	CD	36	44	C2	33	60	21	0D	62	11	0D	61	01	00
B0>	01	ED	B0	D1	C1	2A	13	52	B7	ED	42	C8	03	CD	42	44

FED/CMD Drive 5 Record 13 X'000D' Relative Byte >82  
 Command:R Values X'61'=97

## ENTERING FED

To enter FED, simply type FED <ENTER> at the LDOS Ready prompt. Doing so will cause FED to be loaded and executed. The first prompt you will see will ask you to enter a filespec. Answer this prompt by giving the filespec of the file you wish to examine/modify. If you wish to exit FED at this point, press the <BREAK> key, and you will be returned to the LDOS Ready prompt. If an illegal or improper filespec is given, the appropriate error message will appear, and you will be allowed to re-enter the filespec. The filespec prompt may be bypassed by entering FED using the syntax: FED filespec<ENTER>.

After a valid filespec has been given, the FED 256 character mode will appear on the screen, and the first record (record 0) will be contained in the "edit buffer" (The term "edit buffer" will refer to the record of the file currently in the computer's memory. The edit buffer will contain one 256 byte record at any given time). There will be two cursors flashing within the record (one cursor will be in the "ASCII" portion of the screen, the other cursor will be in the "Hex" display portion), and upon initially accessing a file, these cursors will be positioned over relative byte X'00' of record X'000'. Throughout this documentation, the term "relative byte" will be used, and will indicate the byte number (0-255) relative to the sector in question. Also, hexadecimal notation (X'nn') will be used to represent the current record number and relative byte number.

There will also be an input cursor located on the bottom right portion of the screen, following the message "Command". This will be referred to as the "command buffer", and will be the place on the screen where commands are entered. The current command in use will always be displayed there. When in the 128 character mode, the command buffer will appear on the lower left portion of the screen.

Also shown on the screen will be additional information which may be of importance to the user (such as current record number, filespec, relative byte within the sector, etc.). The sample displays on the previous page will show where on the screen this information will be displayed. For certain commands, inputs of several characters will be required. Depending on the mode you are in (256 or 128 character mode), these inputs will be taken in a different manner.

When in the 256 character mode, these types of inputs will be taken in an input box, and the input box will be positioned vertically along the right hand edge of the display.

When in the 128 character mode, these types of inputs will be taken directly to the right of the command buffer. No input box will appear, but a flashing cursor will be present, indicating that an input is requested.

It is advised that when using FED, the <BREAK> key should always remain enabled, as some FED commands are exited by the use of the <BREAK> key.

The remainder of this manual will be dedicated to the discussion and explanation of all commands available in the FED program.

## FED LIBRARY

<A> Enter ASCII character modify mode  
<B> Position to the Beginning record  
<C> cccccc ASCII Character string search for cccccc  
<D> Dump Disk File to printer (from current position)  
<E> Position to the Ending record  
<F> nnnnnn Find Hex string nnnnnn  
<G> Go to the next occurrence of last search (Hex or ASCII)  
<H> Enter Hex modify mode  
<L> nnnn Locate Hex load address nnnn  
<M> Memory location of a specified byte  
<N><ENTER> New File request (open a different file)  
<O> Output a top-of-form to printer (X'0C')  
<P> Print current record in edit buffer  
<R> nnnn Position to Record nnnn  
<S><ENTER> Save current record (sector) in edit buffer  
<T> Toggle between 256 and 128 display mode  
<X><ENTER> eXit FED and return to LDOS Ready  
<Z> "Zip" through File Load Blocks  
<BREAK> Cancel current FED command  
<ENTER> Display FED instruction set (Menu)  
<;> (+) Advance one record in the file  
<-> Backup one record in the file  
<SHIFT><=> Display binary representation of byte (128 byte mode only)

## CURSOR MOVEMENT

<←> Move cursor left.  
<→> Move cursor right.  
<↑> Move cursor up.  
<↓> Move cursor down.  
<SHIFT><↑> Position cursor to relative byte X'00' of the current record.

## MENU DISPLAY OF FED INSTRUCTION SET

<;>	Forward ONE Record	<BREAK>	Cancels command
<->	Backward ONE Record	<N><ENTER>	New File
<B>	Beginning Record of File	<S><ENTER>	Save Record
<E>	Ending Record of File	<X><ENTER>	Exit FED
<R>	Position to Record	<H>	Hexadecimal Modify
<Z>	Go to next Load Block	<A>	ASCII Modify
<M>	Calculate Load Address	<T>	Toggle Display modes
<C>	Find ASCII String	<F>	Find Hex string
<L>	Locate Hex Load Address	<G>	Go next occurrence
<D>	Dump File to Printer	<O>	Output top-of-form
<P>	Send Buffer to Printer	<=>	Display Binary Value

Press <ENTER> to Return to Display Mode

## FED MANIPULATION COMMANDS

- <;> Advance one record sequentially in the file. For example, if FED was currently displaying record X'000C' and <;> was pressed, the contents of record X'000D' would be displayed (provided that a record X'000D' existed in the file). An "\*" will be displayed directly below the record number when pointing to the last record in the file. Issuing the <;> command will not change the position of the relative byte cursors. A "+" will be shown in the command buffer to show positive motion in the file.
- <-> Back up one record in the file. If FED was currently displaying record X'0087' and <-> was pressed, the contents of record X'0086' would be displayed. Issuing the <-> command does not change the position of the relative byte cursors. The <-> command will be ignored if it is issued when record 0 is being displayed. A "-" will be shown in the command buffer to show negative motion in the file.
- <B> Position to the beginning of the file (record X'0000') and point cursors to relative byte X'00'.
- <E> Position to the Ending record of the file. An "\*" will appear directly below the record number, indicating that the record being displayed is the last record in the file. The relative byte cursors will be positioned on the last byte in the file (not necessarily relative byte X'FF'). Since LDOS uses sector I/O, the whole sector will be displayed, and any byte in the sector may be modified. Realize that any modifications made to bytes beyond the last byte will not cause the EOF marker of the file to be updated to reflect these changes.
- <R>nnnn Position to record X'nnnn', provided record X'nnnn' exists in the file. If the record does not exist, an "\*" will appear in the command buffer. After entering <R>, a box will appear below the record number display box. The input for the record number to retrieve will be taken in this box. Hex digits (0-F) must be entered, as any other characters will be ignored. You may press <BREAK> to cancel this command. The user may enter the record number without using the standard four digit (X'nnnn') format. Simply type in the record number and press <ENTER>. For example, if the desired record number is X'0021', type <R> <2> <1> <ENTER>. To position to record X'0007', type <R> <7> <ENTER>. The position of the relative byte cursors will remain unchanged after the new record is retrieved.

<Z> Points the cursors to the next "Type" byte (X'01', X'02', X'05', X'07', X'10', X'1F') of a Load Module File. This feature is designed to allow the user to ZIP through machine language files quickly. Place the cursors on a "Type" byte and press <Z>. After this has been done, the cursors will be positioned over the next "Type" byte. Encountering a X'02' will terminate a <Z>ip. Any string searching, address locating, or address calculating will disable an active <Z>ip. For more information on "Type" bytes, refer to FILE FORMATS in the Technical Information section of the LDOS manual.

#### FED MODIFICATION COMMANDS

<A> Enters the ASCII Modify Mode. In this mode, modifications can be made in ASCII. Anything you can type in from the keyboard (with the exceptions of the <BREAK> key and the arrow keys) can be sent to the edit buffer. Modifications can be made by positioning the cursor over the bytes to be changed. After the A command is issued, the command buffer will display an "A". From this point on, any characters entered will be taken as modifications to the bytes in the record. The arrow keys may be used to position the cursor for additional edits. To exit the ASCII modify mode, the <BREAK> key must be pressed.

To modify a byte: 1) Position the cursor to the desired byte to change. 2) Type in the ASCII character to replace the original. After making a modification, the relative byte cursors will move to the next byte of the record. Note - no changes are made to disk, only to the edit buffer. To make changes to disk, see the <S>ave command.

<H> Enter the Hex Modify Mode. In this mode, the user can modify bytes in the currently displayed record. Modifications can be made by positioning the cursor over the bytes to be changed. After the H command is issued, the command buffer will display an "H". From this point on, any characters entered will be taken as modifications to the bytes in the record. The arrow keys may be used to position the cursor for additional edits. To exit the Hex modify mode, the <BREAK> key must be pressed.

To modify a byte: 1) Position the cursor over the desired relative byte in the record. 2) Enter the hex digits that you wish to overwrite the current information with. As digits are entered, the previous hex digits will be replaced by the digits entered from the keyboard. The first hex digit entered will modify the first hex digit in the byte, and the second hex digit entered will modify the second hex digit in the byte. After an entire byte has been modified, the cursors will move to the next byte in the record. Note - no changes are made to the disk, only to the edit buffer. To make changes to disk, see <S>ave.

<S><ENTER> Save the contents of the current edit buffer to disk. The current record pointed to by FED will be overwritten by the contents of the edit buffer. Any changes made after the initial read of the record will be written to disk.

#### FED SEARCH COMMANDS

NOTE: The search commands described below may cause the information in the edit buffer to be overwritten by information contained in subsequent records of the file. If edits have been made to the information in the edit buffer, they should be saved to the disk prior to issuing a search command. In most cases, you should issue a "B" command prior to performing a search. This will assure that the entire file will be searched, and no occurrences of the search string will be missed.

<C>cccccc Find ASCII string "cccccc". Issuing the <C> command will cause a search to be performed for the string (cccccc). The search will start at the relative byte pointed to by the cursors. The search is identical to the <F>ind Hex string command, except that the search criteria is an ASCII string of 1 to 30 characters (depending on the display mode being used). Also, the number of characters to be searched for may be an even or an odd number. See the <F> command for further information.

<F>nnnnnn Find hex string "nn nn nn". The <F> command will perform a search for the hex string nn nn nn, starting at the relative byte pointed to by the cursors. (If in the 256 byte display mode, the length of the hex string may be from 2 to 6 characters long, and must be represented as an even number of characters. If in the 128 byte display mode, the length of the hex string may be from 2 to 30 characters long, and must be represented as an even number of characters). The search will begin from the byte over which the cursor is positioned, and will scan all records past the current record until the first occurrence of the string is encountered. If a match is found, the record containing the match will be displayed, and the cursors will be positioned over the first character of the record which matches the search string. To terminate any search, you may press the <BREAK> key. This will cause the record which was contained in the edit buffer prior to the search to be read back in from the disk. If a match is not found, an "\*" will appear in the command buffer, and the cursor will be positioned over relative byte X'FF' of the last record. Only hex bytes can be entered, not hex digits. An "\*" will appear in the command buffer if an odd number of hex digits are entered. If there are multiple occurrences of the specified string, you can "go" to each occurrence by means of the <G>o command.

<G> Go to the next occurrence of current search criteria (string or "L" address). The <G>o command performs a continuation of the last search. If the last search was for a string, it will go to the next occurrence of that string. If the last search was for an address, it will <G>o to the next occurrence of that address. Note - the <G>o works in conjunction with the last search! If the data searched for is not found, one of two things will happen. If the <G>o command is issued after an <L> command and the address is not located, the current record will be read in from disk, and the position of the relative cursors will be unaffected. If the <G>o command is issued after any other search command and the search criteria is not located, the last record will be displayed with the cursor pointing at relative byte X'FF'.

<L>nnnn Locate Hex load address X'nnnn'. The <L> command allows the user to find load address X'nnnn' in a load module file. Unlike the string searches, the <L>ocate command starts its search at record X'0000', rather than at the current cursor position. If the address is located, the record containing the byte at that load address will be displayed, and the cursors will be positioned over this byte. If the address is not located, an error message will be displayed, and you will be prompted to press <ENTER> to continue. After <ENTER> is pressed, the record which was in the edit buffer prior to issuing the <L> command will be retrieved, and the position of the cursors will be unaffected. If a <L>ocate is performed on a non-load module file, the appropriate error message will be displayed. The <G>o command may also be used in conjunction with the <L> command to locate multiple occurrences of the same load address.

#### FED OUTPUT COMMANDS

<D> List the file to the printer, in the same format as the <P>rint command. The <D> command will print all records in the file, starting from the current record number. All records to be printed will be read in from the disk. To halt the printing prior to its completion, depress the <BREAK> key. After the printing has been completed (or terminated), the record which was in the edit buffer prior to printing will be retrieved from disk and stored in the edit buffer, and the cursor position will remain unaffected. Realize that if changes have been made to the record in the edit buffer, these changes should be saved to the disk prior to issuing the <D> command. Several precautions have been taken to prevent computer lock-up during the printing of records. If the printer should become disabled for some reason during printing, FED will continue the printing process after the printer has been enabled. Please note that the LDOS spooler will work in conjunction with the printing operations of FED. Also note that all records will be printed in 20 lines, with a spacing of 2 lines between records. This will allow 3 records to be printed on 66 line/page paper.

- <O> Output a top-of-form character (X'0C') to the printer.
- <P> Send edit buffer contents to a printer in ASCII and Hex. The <P> command will print the contents of the edit buffer. After the <P> command has been issued, the record display on the screen will be sent to the printer. To terminate printing at any time, depress the <BREAK> key. The following is a sample of the output produced by the <P> command:

SPACE/CMD DRIVE 1 RECORD 22 X'0016'

0123456789ABCDEF	BYTE	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
<.2<...<2....D.	<00>	3C	09	32	3C	7F	3A	04	7F	3C	32	04	7F	C3	F6	44	A5
GAME OVER PLAYER	<10>	47	41	4D	45	20	4F	56	45	52	20	50	4C	41	59	45	52
< >NEW HIGH SCO	<20>	20	3C	20	3E	4E	45	57	20	48	49	47	48	20	53	43	4F
REEN....TER NAME	<30>	52	45	45	4E	01	00	B4	97	54	45	52	20	4E	41	4D	45
I.	<40>	20	20	20	20	20	20	20	20	20	20	20	20	20	20	21	C4
W:`...Ww:a...Ww:	<50>	57	3A	60	7F	CD	F1	57	77	3A	61	7F	CD	F7	57	77	3A
a...Ww:b...Ww:b.	<60>	61	7F	CD	F1	57	77	3A	62	7F	CD	F7	57	77	3A	62	7F
..Ww....0#..//.	<70>	CD	F1	57	77	C9	E6	0F	C6	30	23	C9	CB	2F	CB	2F	CB
/./...0#. PLAYE	<80>	2F	CB	2F	18	F0	C6	30	23	C9	20	20	50	4C	41	59	45
R < > .....	<90>	52	20	3C	20	3E	20	20	FF								
.....	<A0>	FF															
.....	<B0>	FF	FF	FF	FF	80	88	B7	B7	B7	B7	9D	80	AE	BB	BB	BB
..... INTRU	<C0>	BB	84	80	80	80	80	80	80	80	80	20	49	4E	54	52	55
DERS.....	<D0>	44	45	52	53	AE	9D	AE	9D	88	9B	A7	84	88	9E	AD	84
....0xH.x.0 H0.H	<E0>	A0	99	A6	90	30	78	48	B4	78	84	30	7C	48	30	F8	48
POINTS20 POINTS1	<F0>	50	4F	49	4E	54	53	32	30	20	50	4F	49	4E	54	53	31

### FED MISCELLANEOUS COMMANDS

- <ENTER> Display FED instruction menu.
- <X><ENTER> Exit FED and return to LDOS Ready.
- <N><ENTER> Open a New file for editing. A prompt for the filespec will be displayed. If you input an invalid or improper filespec, an error message will appear, and you will be allowed to re-enter the filespec. Note that FED will never close files, as files need not be closed with this type of editor.
- <BREAK> Clear command buffer. Pressing <BREAK> will cancel any partial command, and will cause the termination of any command being executed. It is also the only way to exit the ASCII and Hex modify modes. Anytime there is any doubt as to the operation being performed by FED, you may press <BREAK>, and the command buffer will be cleared.

- <SHIFT><=> Display binary representation of byte pointed to by the cursors. This command may only be used when in the 128 character mode, and will be ignored if issued in the 256 character mode. After depressing <SHIFT><=>, 8 binary digits will be displayed next to the command buffer. For example, if the cursors were positioned over relative byte X'27', and this byte of the edit buffer contained a X'F3', the binary digits 1111,0011 would be displayed.
- <M> Calculates the address in memory where the byte pointed to by the cursors will load. This command works with load module format files only. If the byte is contained in a load block, the load address will be displayed below the record number. If the byte is not in a load block (e.g. a comment line, file header, etc.) the error message "Byte not in load block" will be displayed.
- <T> Toggle between the regular 256 byte mode and the extended 128 byte mode. By pressing <T>, the user shifts to "the other" mode. The 128 character mode has all of the same commands as the 256 character mode. The display is a window of the 256 byte record, and 128 bytes will be displayed. By moving the cursors (usually with the <UP> and <DOWN> arrows) you will notice a scrolling effect. The ASCII display will be at the top of the screen instead of the 16 leftmost columns. The current record number is displayed in decimal as well as hexadecimal. All inputs will be taken horizontally instead of vertically. ASCII and hex search inputs will allow 30 characters instead of 6.

#### USING DRIVERS, FILTERS, OR PROGRAMS WITH FED

FED works harmoniously with other programs as long as FED is not tampered with. User programs should not use any memory below X'7700'. When returning to FED from some other function, the display may appear to be garbaged. Simply press <BREAK> and the FED display will be re-established. As far as drivers and filters are concerned, FED uses keyboard, video and printer DCB'S, so any vectors changed by another driver will be picked up by FED. NOTE: For FED to function properly, the user must maintain standard ASCII values and restore any registers, DCB's, devices, etc. to their original values. You may also use FED in conjunction with a machine language program. It is possible (although not recommended) to use FED from within LBASIC via the CMD"FED" command. In order to utilize FED in this manner, the user should make sure that at least 10,000 bytes are free in LBASIC. If the number of free bytes is less than 10,000, the system will most likely crash.

## Q F B

This utility is designed to allow for a backup with format to be performed. Only floppy drives may be used, and the backup performed must be mirror image. The syntax is:

```
=====
| QFB :s :d (parm,parm,parm)
|
| :s  is the Source drive. The colon is optional.
| :d  is the Destination drive. The colon is optional.
|
| The following optional parameters may be used:
|
| ALL= parameter used to specify whether all cylinders
|       of the source disk will be read and copied to
|       the destination disk, or only allocated
|       cylinders will be used. The switch ON or OFF
|       may be specified, with the default being OFF.
|
| V1=  parameter used to specify whether or not a
|       verify of the destination disk is to be
|       performed on the 1st pass. The switch ON or OFF
|       may be used, with the default being ON.
|
| V2=  parameter used to specify whether or not a
|       verify of the destination disk is to be
|       performed on the 2nd pass. The switch ON or OFF
|       may be used, with the default being OFF.
|
| QUERY= Query for parameters not specified. The switch
|         ON or OFF may be used. The default is OFF
|
| abbr:  ON=Y, OFF=N, QUERY=Q, ALL=A
|
=====
```

The QFB (Quick Format and Backup) utility will allow for the creation of a mirror image backup of a source disk without having to format the destination disk prior to executing the backup. The normal means by which a mirror image backup is made using LDOS is to first format a diskette using the FORMAT utility, and then use the BACKUP utility to perform the backup. The limitations of the QFB utility are as follows:

- 1.) Two distinct floppy drives must be used.
- 2.) The source diskette must have been formatted using the LDOS 5.1.x FORMAT utility, and cannot contain any non-standard format.
- 3.) QFB will run exclusively on LDOS 5.1.x, versions 5.1.3 or later.

QFB will perform a "single pass" format and backup. If QFB is entered with no drives specified, prompts will appear for them. If drive numbers are specified, the first drive number will represent the source drive, and the destination drive will be the second drive number. If no parameters are specified, the defaults will be used.

Consider the results of entering the following command.

QFB 1 2

Drive 1 will be used as the source drive, while drive 2 will be the destination drive. Prior to QFB performing any action, a prompt will appear to load the diskettes. Once the proper diskettes have been installed, press <ENTER>, and the backup will begin. The following actions will take place.

- 1.) The source diskette will be logged in, to determine the type of format.
- 2.) Cylinder 0 of the destination diskette will be formatted.
- 3.) If cylinder 0 of the source disk contains data, it will be read into memory.
- 4.) If cylinder 0 of the source diskette contains data, the information stored in memory (see Step 3) will be written out to the destination diskette.
- 5.) Cylinder 0 of the destination diskette will be verified.
- 6.) Steps 2-5 will be repeated for all remaining cylinders.
- 7.) The following message will appear after the last cylinder has been verified:

Duplication complete     1 disk created

Replace destination disks and press <ENTER> to repeat  
..<R> to restart with new parameters  
...or....<BREAK> to exit program.

- 8.) Press <ENTER> in response to this prompt to make another mirror image backup.  
Press <BREAK> to abort the QFB utility. The following prompt will appear:

Load SYSTEM diskette and hit <ENTER>

Place a system diskette in drive 0 and press <ENTER>, to return to LDOS Ready.

If it is desired to use QFB again with different parameters, press <R> in response to the prompt displayed in step 7. Doing so will cause the drives to be prompted for, and prompts will appear for all parameters.

If QFB is to be restarted, or the command QFB (Q=Y) is entered, the following prompts for the parameters will occur:

Duplicate unallocated tracks? (Y/N)  
Verify on same pass? (Y/N)  
Verify on second pass? (Y/N)

The first prompt relates to the ALL parameter. If it is answered with <Y>, all cylinders will be read from the source diskette and written to the destination diskette, regardless of whether or not the cylinder contains information. If this prompt is answered <N>, only cylinders containing information will be read and written.

The next prompt relates to the V1 parameter. If it is answered with <Y>, all cylinders on the destination diskette will be verified immediately after all writes. If answered <N>, no immediate verify will be done.

The final prompt corresponds to the V2 parameter. If it is answered with <Y>, all cylinders on the destination diskette will be verified upon completion of all writing to the diskette. If answered <N>, there will be no second pass verification.

If an error occurs, an appropriate error message will be displayed, and a prompt will appear requesting the course of action that is desired. During any QFB operation, the <BREAK> key will be active, and can be used to abort the process.

#### I M P O R T A N T

QFB assumes that a mirror image backup is desired, and performs no check on the destination diskette with respect to the existence of data. Any existing information on a destination diskette will ALWAYS be destroyed. Also, QFB will NOT clear the Mod Flags of files on the source diskette.