DEC-10-ULNDA-A-D

decsystem10 LINED
LINE EDITOR FOR DISK FILES

digital equipment corporation · maynard. massachusetts

1st Printing June 1971
2nd Printing (Rev) July 1972

CONTENTS

LINED

A LINE EDITOR FOR DECsystem-10 FILES

LINED is a line editor for disk files. It is used to create and edit source program files which are written on disk in ASCII code with line sequence numbers appended. LINED has the ability to reference any line at any time without the user having to close and reopen the file. LINED is a reentrant program and loads in 2K pure and 2K impure segments of core.

NOTE

In this document, computer typeouts are indicated by underscoring. The symbol) represents the RETURN key. The symbol (Ⓢ) represents the ALTMODE key.

1.0 MONITOR COMMANDS

The MONITOR commands CREATE and EDIT may be used to select a file for editing with LINED. A temporary disk file, called ###EDT.TMP, is created for passing the commands to LINED.

1.1 The CREATE Command

The CREATE command calls in LINED and opens the specified new disk file for editing. The CREATE command is of the form:

.CREATE filename.ext)

1.2 The EDIT Command

The EDIT command calls in LINED and opens the specified existing disk file for editing. The EDIT command is of the form:

.EDIT filename.ext)

2.0 LINED COMMANDS

LINED indicates its readiness to receive commands by typing an asterisk. At this time LINED is said to be in command mode. The user may then type in the following LINED commands.

2.1 Inserting or Replacing a Line

```
*Innnnn
nnnn   aaaa.....a
nnnxx  $
*
```

Insert or replace the following typed line at line number nnnnn of the currently open file; nnnnn can be specified as a line sequence number or a point (.), or it can be omitted entirely. A point refers to the last line which was typed, or the last line deleted, or the last line inserted. If nnnnn is omitted, it is assumed to be 10.

When LINED has typed a line sequence number, the program enters text mode. In the text mode, characters typed by the user are understood to be text for the insertion. Following the user's typein of the line to be inserted, LINED types out the next sequential line number (nnnnn+10) following which the user presses the ALTMODE key (sometimes labeled PREFIX or ESC) to terminate the insert process and return to LINED command level.

If there already exists a line at nnnnn, it will be replaced. A single quote following the line number indicates that insertion at this line number will cause the existing line to be replaced.

2.2 Inserting Multiple Lines

```
*Innnnn,iiiii
nnnnn   aaaaa.....a
nnnxx   bbbbb.....b
  .
  .
nnnyy   $
*
```

Insert the following typed lines, beginning at line number nnnnn (which can be specified as either a line number or a point) of the currently open file. Each time a line is entered, nnnnn is increased by the specified increment, iiiii. If iiiii is omitted, it is assumed to be 10 (if iiiii has never been specified previously), or the previous increment specified.

If nnnnn is omitted, it is assumed to be 10, and the result becomes the line number of the next insertion. Type ALTMODE on the line following the last insertion to return to LINED command mode. LINED then awaits another command.

A double quote following a line number indicates that the increment specified for the current insert instruction has resulted in an existing line being skipped.

2.3 Deleting a Line

```
*Dnnnnn
```

Delete a line number nnnnn from the currently open file; nnnnn can be specified as either a line sequence number or a point.

2.4 Deleting Multiple Lines

*Dmmmmm,nnnnn — Delete Lines mmmmm through nnnnn from the currently open file; mmmmm must be less than nnnnn. Either mmmmm or nnnnn may be specified as a point as long as mmmmm is less than nnnnn.

2.5 Printing a Line

*Pnnnnn — Print line nnnnn on the user's Teletype; nnnnn can be specified as either a line sequence number or a point. Typing ALTMODE following a typeout will cause the next sequential line to be printed.

2.6 Printing Multiple Lines

* Pmmmmm,nnnnn — Print lines mmmmm through nnnnn of the currently open file; mmmmm must be less than nnnnn. Either mmmmm or nnnnn may be specified as a point as long as mmmmm is less than nnnnn.

2.7 Closing the Current File

E) — Closes the current file and returns to LINED command mode. At this point, the user may either open another file or type ↑C to return to Monitor level to assemble, list, and/or save his file on a permanent storage device (e.g., DECtape).

2.8 Examples of Command Sequence

Example 1

```
.CREATE FILEA                RUN LINED AND OPEN FILE FILEA
*I10                         BEGIN INSERTING LINES AT LINE NUMBER
00010 THE PROGRAM            10 INCREMENTING BY 10.
00020 IS INSERTED
00030 HERE
   .
   .
   .
00350 ($)                    RETURN CONTROL TO LINED COMMAND
*E                           MODE BY TYPING ($) . CLOSE FILE FILEA
*↑C                          BY TYPING AN E. TYPING A ↑C RETURNS
.                            TO THE MONITOR COMMAND LEVEL.
```

Example 2

```
.EDIT FILEA                    RUN LINED AND OPEN EXISTING FILE
                               FILEA
*P10,30                        PRINT LINES 10 THROUGH 30
00010 THE PROGRAM              PRINTOUT
00020 IS INSERTED
00030 HERE
*I20                           INSERT LINE 20
00020 IS PLACED
00030 ($)
*D30                           DELETE LINE 30
*P 10,30                       PRINT LINES 10 THROUGH 30
00010 THE PROGRAM              PRINTOUT
00020 IS PLACED
* E                            TYPE E TO CLOSE FILE FILEA
*↑C                            TYPING A ↑C RETURNS JOB TO MONITOR
.                              CONTROL LEVEL.
```

3.0 AUXILIARY COMMANDS

These Auxiliary Commands provide an alternate method of calling LINED and opening files. In most cases, auxiliary commands can be replaced by the monitor instructions CREATE and EDIT (Section 1).

3.1 R LINED

LINED can be called in from the system device by typing

```
.R LINED )
*
```

LINED responds with an asterisk to indicate its readiness to receive a command.

3.2 Initializing a File for Processing

S filename.ext)	Select an existing disk file, filename.ext, for editing.
S filename.ext ($)	Select (create) a new disk file for editing, calling it filename.ext.

4.0 LINED CONVENTIONS AND RESTRICTIONS

The following conventions and restrictions should be noted.

a. Files are written with the installation standard protection. See the DECsystem-10 Operating System Commands manual for explanation of protected files.

b. When in insert mode, typing ALTMODE following the printout of the next insertion line sequence number causes a returned to LINED command level. Typing ALTMODE to terminate a line of text to be inserted causes the text line to be ignored.

```
00010    LINE OF TEXT
00020    ($)                    Returns to LINED command level
*

00010    LINE OF TEXT ($) Line is ignored
*
```

c. LINED assumes that all blocks in a disk file have an integral number of lines (i.e., each block begins with a sequence number and no line is split between blocks). This will always be the case with files which have been created and edited only with LINED; however, if sequence numbers have been removed, say by TECO, they may be restored by using PIP switch /S.

d. LINED files can be resequenced using PIP switch /S.

e. Line number 0 is illegal and cannot be used.

f. Lines can be edited in any order; however, editing lines by ascending line numbers reduces file access time.

5.0 ERROR HANDLING

When an error is detected, LINED types a message and returns the user to LINED command level (indicated by the output of an * on the Teletype). Some errors are fatal and cause control to return to the monitor. Error messages for LINED are given in Table 1.

Table 1
LINED Error Messages

Message	Meaning
?FILE NAME ALREADY IN USE	The filename specified in a CREATE or S command already exists on disk. Type the S command with a correct filename, followed by ($).
?FILE NOT SPECIFIED	The user attempted to execute an editing command without first naming the file to be edited. Using an S command, name the file to be edited.
?ILLEGAL COMMAND	The user attempted to use a letter that is not a command. Type the correct command letter.
?INPUT FILE NOT FOUND	The file named in an EDIT or S command cannot be found on disk. Either place the file on disk, or create the file with the S command followed by ($).

(continued on next page)

Table 1 (Cont)
LINED Error Messages

Message	Meaning
?LINE REFERENCED DOES NOT EXIST	A line referenced in a P or D command does not exist in the file. Either retype the command with the correct line number, or insert the line.
SYSTEM ERROR READING COMMAND FILE	A system error occurred while LINED was trying to read the CCL command file generated by a CREATE or EDIT command. Try to create or select the file using the appropriate form of the S command.
NOTE The following are internal system errors and cause control to return to the monitor.	
?CANNOT ACCESS DISK	LINED cannot access the disk. This message can only occur at the beginning of operations. Notify the system manager.
?CANNOT INIT TTY	LINED cannot initialize the user's terminal. This message can only occur at program initialization. Notify the system manager.
?ERROR IN RENAME PROCESS INPUT FILE CLOSED WITH NAME filnam.ext OUTPUT FILE CLOSED WITH NAME ###LIN.TMP	An error occurred while LINED was renaming the output file. The input file should be renamed filnam.BAK and the output file should be renamed filnam.ext.
?ERROR IN RENAME PROCESS INPUT FILE CLOSED WITH NAME ###TMP.TMP OUTPUT FILE CLOSED WITH NAME ###LIN.TMP	An error occurred while LINED was renaming the files. The input file should be renamed filnam.BAK and the output file should be renamed filnam.ext.
?ERROR IN RENAME PROCESS INPUT FILE CLOSED WITH NAME ###TMP.TMP OUTPUT FILE CLOSED WITH NAME filnam.ext	An error occurred while LINED was renaming the files. The input file should be renamed filnam.BAK. In the three messages above, ### is the user's job number and filnam.ext is the name of the file that he was editing.
?INPUT ERROR. INCOMPLETE OUTPUT FILE CLOSED WITH NAME ###LIN.TMP	A system error occurred on input. The output file is incomplete; thus, the user must start editing again with the backup file.
?NO CORE AVAILABLE FOR DATA SEGMENT	There is no core available for LINED to do editing on the user's file. This message can occur only during program initialization. Notify the system manager.
?OUTPUT ERROR. INCOMPLETE OUTPUT FILE CLOSED WITH NAME ###LIN.TMP	A system error occurred on output. The output file is incomplete; thus the user must start editing again with the backup file.

6.0 IMPLEMENTATION

The following explanation is intended to help the user to understand how LINED works so that he may use it more effectively.

Lines of text are stored in a 1000-word working buffer. Each line has a 1-word header containing two items. The left half contains the sequence number of the line, and the right half contains the number of words (including the word containing the line header) needed to store the line of text. Thus, to find the beginning of the next line of text, it is necessary to simply take the address of the current line header and add the word count of the current line.

Several pointer words are used to keep track of the lines in the working buffer. WRTLST contains the sequence number of the highest line in the buffer. SN contains the sequence number of the line currently being handled in a command.

When LINED discovers that SN is greater than WRTLST, it knows that the line being sought has already passed through the working buffer. This line is not directly accessible, because there is no way to read a disk file backwards. Consequently, it is necessary for LINED to close the file and then reopen it. This process of going from the current position of the file to the end of the file, from there to the beginning of the file, and finally to the line being sought is accomplished as follows:

a. To close the file, all remaining text must be passed through the working buffer to the temporary output file (called ###LIN.TMP). This is done by giving the subroutine FNDLIN (which finds a line whose sequence number is SN) the highest possible sequence number - 99999.

b. Next, the original file is renamed to ###TMP.TMP, the temporary output file is renamed to the original filename and the original file (###TMP.TMP) is renamed to name.BAK (same name as original with an extension of BAK).

c. FNDLIN is then given the sequence number being sought, and LINED continues with the original command.

7.0 STANDARD FOR DECsystem-10 LINE SEQUENCE NUMBERS

ASCII data files containing line sequence numbers conform to the following rules.

a. Each line must begin at a word boundary. Lines are padded at the end with nulls to fill an integral number of words.

b. Every line must have a line sequence number.

c. The line sequence number consists of five ASCII characters contained in the first word of the line.

d. Bit 35 of the line sequence number word is set to 1.

e. The line sequence number can contain only decimal digits. The characters preceding the first non-zero digit should be ASCII zeros. However, on input, leading spaces as well as leading zeros are accepted for compatibility with those data files that have leading spaces.

f. The first character after the line sequence number is always a tab except in files created by BASIC. All compilers except BASIC ignore the character after the line sequence number. The utility programs (editors and PIP) automatically cause a tab to follow the line sequence number when they are creating new line sequence numbers. However, for compatibility with BASIC, the utility programs do not force a tab after the line sequence number when they are merely transferring existing line sequence numbers from an input file to an output file.

g. Line blocking is optional.