

TO: Distribution  
FROM: Ross E. Klinger  
DATE: 10/25/73  
SUBJECT: New command, sort, to replace the current sort\_file  
command, and a new subroutine, sort\_.

Attached are the MPM write-ups for the proposed command sort, and its associated subroutine, sort\_. The new procedures will provide the following capabilities which were not implemented by sort\_file:

- 1) Specification of a major, and multiple minor, sort fields. sort\_file sorts on the entire line.
- 2) The ability to sort blocks of lines, and specification of sort fields relative to position within a block.
- 3) Specification of the line delimiter as any valid ASCII character string. sort\_file assumes the new line character.
- 4) Either ascending or descending sort order.

Provisions have been made so that the eventual implementation of a multi-segment file capability should require no changes to the user interface.

I would appreciate your comments and suggestions. Send written comments to:

Ross Klinger  
MIT Room 464, Bldg. 39

or mail comments to:

Klinger.PDO

---

Multics Project Internal working documentation. Not to be reproduced or distributed outside the Multics Project.

Name: sort, so

The sort command sorts the lines of an ASCII file according to the ASCII collating sequence. The reordered lines replace the original contents of the file.

Usage sort pathname -control\_args- -field\_specs-

- 1) pathname specifies the pathname of the file to be sorted. pathname may be either an entry name or an absolute pathname. The sorted lines replace the original contents of the file.
- 2) control\_args may be chosen from the following list of control arguments:
  - delimiter xxx specifies the character sequence used to delimit a line. xxx is any sequence of ASCII characters. The default is the new line character.
  - dm xxx
  - block n specifies the sort unit to be a block of n lines. The default is n = 1.
  - bk n
  - reverse specifies the sort to be in descending order. The default is ascending order.
  - rv
- 3) field\_specs may be one or more pairs of field specifications. A field specification defines the starting and ending positions of a sort field within a sort unit. The first pair of field specifications is taken to define the major sort field; the second pair, the primary minor; and so forth. Only the sort fields of a sort unit are considered when comparing one unit with another. (See Notes.)
  - from n specifies the starting position of a sort field. n must be a positive integer. The default is n = 1.
  - fm n

-to m specifies the ending position of a sort field. m must be a positive integer greater than or equal to n. The default is m equal to the last position of the sort unit.

### Notes

If the blocking factor is 1, field specifications refer to positions within a line. If the blocking factor is greater than 1, field specifications refer to positions within a block. In any case, the delimiter is ignored when computing field positions. (See Examples.)

Units of unequal length are compared by assuming the shorter unit to be padded on the right (after the unit's delimiter) with blanks. (See Examples.)

If characters are found in the file after the final unit's delimiter, they will be ignored, but will appear in the sorted file following the last sorted unit.

The file is sorted using temporary segments in the process directory. The file itself is not modified until the last moment.

### Examples

Given the characters:

ABCDEFGHIJABCDEFGHIJABCDEFGHIJABCDEFGHIJ

<u>-block 1</u> <u>-delimiter GH</u>	will result in these sort units	with these sort fields
	ABCDEF IJABCDEF IJABCDEF IJABCDEF	ABCDEFØØ IJABCDEF IJABCDEF IJABCDEF
<u>-block 1</u> <u>-delimiter GH</u> <u>-from 4</u> <u>-to 7</u>	ABCDEF IJABCDEF IJABCDEF IJABCDEF	DEFØ BCDE BCDE BCDE

sort

```
-block 2      ABCDEFIJABCDEF      ABCDEFIJABCDEFØØ
-delimiter GH IJABCDEFIJABCDEF.  IJABCDEFIJABCDEF

-block 2      ABCDEFIJABCDEF      DEFI
-delimiter GH IJABCDEFIJABCDEF      BCDE
-from 4
-to 7
```

Name: sort\_

The sort\_ subroutine orders the lines of an ASCII file according to the ASCII collating sequence. The ordered lines replace the original contents of the file.

Usage

```
dcl sort_ entry (char (*), char (*), char (*), bit (1),
                fixed bin, ptr, fixed bin (35));
```

```
call sort_ (dirname, ename, delim, ad, block, fptr, code);
```

- |            |  |
|------------|--|
| 1) dirname | is the directory name of the segment to be sorted. (Input)   |
| 2) ename   | is the entry name of the segment to be sorted. (Input)   |
| 3) delim   | is the ASCII character string which delimits a line. (Input)   |
| 4) ad      | is the ascending/descending order bit:<br>= "0"b for descending order<br>= "1"b for ascending order. |
| 5) block   | is the number of lines per sort unit. (Input)  |
| 6) fptr    | is a pointer to the user-declared field specifications structure. See <u>Notes</u> . (Input)         |
| 7) code    | is a non-standard error code. See <u>Notes</u> . (Output)  |

Notes

See the MPM write-up of the sort command for a detailed explanation of sort units and field specifications.

sort_
-------

Page 2

The field specifications structure is declared as follows:

```

dc1 1 field_specs aligned based (fptr),
    2 n fixed bin,
    2 fields (0 refer (field_specs.n)),
    3 start fixed bin,
    3 stop fixed bin;

```

- 1) n is the number of field specifications.
- 2) fields(n).start is the starting position of sort field n.
- 3) fields(n).stop is the stopping position of sort field n.

fields(1) is the major sort field, fields(2) the primary minor, fields(3) the secondary minor, and so forth.

The following error codes may be returned:

- 0 - normal return
- 1 - segment to be sorted could not be initiated
- 2 - a temporary segment could not be made in the process directory.
- 3 - the length of the delimiter is greater than the length of the segment to be sorted
- 4 - no delimiter was found in the segment to be sorted
- 5 - characters were found after the final delimiter and were placed at the end of the ordered lines

If the error code is 0 or 5, the ordered lines have replaced the original contents of the file. Otherwise, the file remains in its original state.