IEEE P1003.2 Draft 11.2 – September 1991

Copyright © 1991 by the
Institute of Electrical and Electronics Engineers, Inc.
345 East 47th Street
New York, NY 10017, USA
All rights reserved as an unpublished work.

This is an unapproved and unpublished IEEE Standards Draft, subject to change. The publication, distribution, or copying of this draft, as well as all derivative works based on this draft, is expressly prohibited except as set forth below.

Permission is hereby granted for IEEE Standards Committee participants to reproduce this document for purposes of IEEE standardization activities only, and subject to the restrictions contained herein.

Permission is hereby also granted for member bodies and technical committees of ISO and IEC to reproduce this document for purposes of developing a national position, subject to the restrictions contained herein.

Permission is hereby also granted to the preceding entities to make limited copies of this document in an electronic form only for the stated activities.

The following restrictions apply to reproducing or transmitting the document in any form: 1) all copies or portions thereof must identify the document's IEEE project number and draft number, and must be accompanied by this entire notice in a prominent location; 2) no portion of this document may be redistributed in any modified or abridged form without the prior approval of the IEEE Standards Department.

Other entities seeking permission to reproduce this document, or any portion thereof, for standardization or other activities, must contact the IEEE Standards Department for the appropriate license.

Use of information contained in this unapproved draft is at your own risk.

IEEE Standards Department
Copyright and Permissions
445 Hoes Lane, P.O. Box 1331
Piscataway, NJ 08855-1331, USA
+1 (908) 562-3800
+1 (908) 562-1571 [FAX]

11822 ### 4.57.9 Consequences of Errors

11823 Default.

11824 ### 4.57.10 Rationale. *(This subclause is not a part of P1003.2)*

11825 **Examples, Usage**

11826 The exit status is allowed to be zero when `sleep` is interrupted by the SIGALRM
11827 signal, because most implementations of this utility rely on the arrival of that sig-
11828 nal to notify them that the requested finishing time has been successfully
11829 attained. Such implementations thus do not distinguish this situation from the
11830 successful completion case. Other implementations are allowed to catch the sig-
11831 nal and go back to sleep until the requested time expires or provide the normal
11832 signal termination procedures.

11833 **History of Decisions Made**

11834 As with all other utilities that take integral operands and do not specify
11835 subranges of allowed values, `sleep` is required by this standard to deal with *time*
11836 requests of up to 2 147 483 647 seconds. This may mean that some implementa-
11837 tions will have to make multiple calls to the underlying operating system's delay
11838 mechanism if its argument range is less than this.

11839 ## 4.58 `sort` — Sort, merge, or sequence check text files

11840 ### 4.58.1 Synopsis

11841 `sort [-m][-o` *output* `][-bdfinru][-t` *char* `][-k` *keydef* `] ... [`*file* `...]`

11842 `sort -c[-bdfinru][-t` *char* `][-k` *keydef* `] ... [`*file* `]`

11843 *Obsolescent Versions* :

11844 `sort [-mu][-o` *output* `][-bdfinr][-t` *char* `][+`*pos1* `[-`*pos2* `]] ... [`*file* `...]`

11845 `sort -c[-u][-bdfinr][-t` *char* `][+`*pos1* `[-`*pos2* `]] ... [`*file* `]`

11846 ### 4.58.2 Description

11847 The `sort` utility shall perform one of the following functions:

11848 (1) Sort lines of all the named files together and write the result to the
11849 specified output.

11850 (2) Merge lines of all the named (presorted) files together and write the
11851 result to the specified output.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

(3) Check that a single input file is correctly presorted.

Comparisons shall be based on one or more sort keys extracted from each line of input (or the entire line if no sort keys are specified), and shall be performed using the collating sequence of the current locale.

4.58.3 Options

The `sort` utility shall conform to the utility argument syntax guidelines described in 2.10.2, except that the notation *+pos1 −pos2* uses a nonstandard prefix and multidigit option names in the obsolescent versions, the `−o` *output* option shall be recognized after a *file* operand as an obsolescent feature in both versions where the `−c` option is not specified, and the `−k` *keydef* option should follow the `−b`, `−d`, `−f`, `−i`, `−n`, and `−r` options.

The following options shall be supported by the implementation:

`−c` Check that the single input file is ordered as specified by the arguments and the collating sequence of the current locale. No output shall be produced; only the exit code shall be affected.

`−m` Merge only; the input files shall be assumed to be already sorted.

`−o` *output* Specify the name of an output file to be used instead of the standard output. This file can be the same as one of the input *file*s.

`−u` Unique: suppress all but one in each set of lines having equal keys. If used with the `−c` option, check that there are no lines with duplicate keys, in addition to checking that the input file is sorted.

The following options shall override the default ordering rules. When ordering options appear independent of any key field specifications, the requested field ordering rules shall be applied globally to all sort keys. When attached to a specific key (see `−k`), the specified ordering options shall override all global ordering options for that key. In the obsolescent forms, if one or more of these options follows a *+pos1* option, it shall affect only the key field specified by that preceding option.

`−d` Specify that only `<blank>`s and alphanumeric characters, according to the current setting of LC_CTYPE, shall be significant in comparisons. The behavior is undefined for a sort key to which `−i` or `−n` also applies.

`−f` Consider all lowercase characters that have uppercase equivalents, according to the current setting of LC_CTYPE, to be the uppercase equivalent for the purposes of comparison.

`−i` Ignore all characters that are nonprintable, according to the current setting of LC_CTYPE.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

11890 −n Restrict the sort key to an initial numeric string, consisting of
11891 optional <blank>s, optional minus sign, and zero or more digits
11892 with an optional radix character and thousands separators (as
11893 defined in the current locale), which shall be sorted by arithmetic
11894 value. An empty digit string shall be treated as zero. Leading
11895 zeros and signs on zeros shall not affect ordering.

11896 −r Reverse the sense of comparisons.

11897 The treatment of field separators can be altered using the options:

11898 −b Ignore leading <blank>s when determining the starting and end-
11899 ing positions of a restricted sort key. If the −b option is specified
11900 before the first −k option, it shall be applied to all −k options.
11901 Otherwise, the −b option can be attached independently to each
11902 −k *field_start* or *field_end* option-argument (see below).

11903 −t *char* Use *char* as the field separator character; *char* shall not be con-
11904 sidered to be part of a field (although it can be included in a sort
11905 key). Each occurrence of *char* shall be significant (for example,
11906 <*char*><*char*> shall delimit an empty field). If −t is not specified,
11907 <blank> characters shall be used as default field separators;
11908 each maximal nonempty sequence of <blank> characters that fol-
11909 lows a non-<blank> character shall be a field separator.

11910 Sort keys can be specified using the options:

11911 −k *keydef* The *keydef* argument is a restricted sort key field definition. The
11912 format of this definition is

11913 *field_start* [*type*][,*field_end* [*type*]]

11914 where *field_start* and *field_end* define a key field restricted to a
11915 portion of the line (see 4.58.7), and *type* is a modifier from the list
11916 of characters b, d, f, i, n, r. The b modifier shall behave like the
11917 −b option, but applies only to the *field_start* or *field_end* to which
11918 it is attached. The other modifiers shall behave like the
11919 corresponding options, but shall apply only to the key field to
11920 which they are attached; they shall have this effect if specified
11921 with *field_start, field_end,* or both. Modifiers attached to a
11922 *field_start* or *field_end* shall override any specifications made by
11923 the options. Implementations shall support at least nine
11924 occurrences of the −k option, which shall be significant in com-
11925 mand line order. If no −k option is specified, a default sort key of
11926 the entire line shall be used.

11927 When there are multiple key fields, later keys shall be compared
11928 only after all earlier keys compare equal. Except when the −u
11929 option is specified, lines that otherwise compare equal shall be
11930 ordered as if none of the options −d, −f, −i, −n, or −k were present
11931 (but with −r still in effect, if it was specified) and with all bytes in
11932 the lines significant to the comparison. The order in which lines
11933 that still compare equal are written is unspecified.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

11934 *+pos1* (Obsolescent.) Specify the start position of a key field. See 4.58.7.

11935 *−pos2* (Obsolescent.) Specify the end position of a key field. See 4.58.7.

11936 ### 4.58.4 Operands

11937 The following operand shall be supported by the implementation:

11938 *file* A pathname of a file to be sorted, merged, or checked. If no *file*
11939 operands are specified, or if a *file* operand is −, the standard input
11940 shall be used.

11941 ### 4.58.5 External Influences

11942 #### 4.58.5.1 Standard Input

11943 The standard input shall be used only if no *file* operands are specified, or if a *file*
11944 operand is −. See Input Files.

11945 #### 4.58.5.2 Input Files

11946 The input files shall be text files, except that the `sort` utility shall add a `<new-`
11947 `line>` to the end of a file ending with an incomplete last line.

11948 #### 4.58.5.3 Environment Variables

11949 The following environment variables shall affect the execution of `sort`:

11950 **LANG** This variable shall determine the locale to use for the
11951 locale categories when both **LC_ALL** and the correspond-
11952 ing environment variable (beginning with **LC_**) do not
11953 specify a locale. See 2.6.

11954 **LC_ALL** This variable shall determine the locale to be used to over-
11955 ride any values for locale categories specified by the set-
11956 tings of **LANG** or any environment variables beginning
11957 with **LC_**.

11958 **LC_COLLATE** This variable shall determine the locale for ordering rules.

11959 **LC_CTYPE** This variable shall determine the locale for the interpreta-
11960 tion of sequences of bytes of text data as characters (e.g.,
11961 single- versus multibyte characters in arguments and
11962 input files) and the behavior of character classification for
11963 the `−b`, `−d`, `−f`, `−i`, and `−n` options.

11964 **LC_MESSAGES** This variable shall determine the language in which mes-
11965 sages should be written.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

11966 **LC_NUMERIC** This variable shall determine the locale for the definition
11967 of the radix character and thousands separator for the −n
11968 option.

11969 #### 4.58.5.4 Asynchronous Events

11970 Default.

11971 ### 4.58.6 External Effects

11972 #### 4.58.6.1 Standard Output

11973 Unless the −o or −c options are in effect, the standard output shall contain the
11974 sorted input.

11975 #### 4.58.6.2 Standard Error

11976 Used only for diagnostic messages. A warning message about correcting an 2
11977 incomplete last line of an input file may be generated, but need not affect the final 2
11978 exit status. 2

11979 #### 4.58.6.3 Output Files

11980 If the −o option is in effect, the sorted input shall be placed in the file *output* .

11981 ### 4.58.7 Extended Description

11982 The notation

11983 −k *field_start* [*type*][,*field_end* [*type*]]

11984 shall define a key field that begins at *field_start* and ends at *field_end* inclusive,
11985 unless *field_start* falls beyond the end of the line or after *field_end* , in which case
11986 the key field shall be empty. A missing *field_end* shall mean the last character of
11987 the line.

11988 A field comprises a maximal sequence of nonseparating characters and, in the 1
11989 absence of option −t, any preceding field separator. 1

11990 The *field_start* portion of the *keydef* option argument shall have the form:

11991 *field_number* [.*first_character*]

11992 Fields and characters within fields shall be numbered starting with 1. The
11993 *field_number* and *first_character* pieces, interpreted as positive decimal integers,
11994 shall specify the first character to be used as part of a sort key. If .*first_character*
11995 is omitted, it shall refer to the first character of the field.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

The *field_end* portion of the *keydef* option argument shall have the form:

field_number **[**.*last_character* **]**

The *field_number* shall be as described above for *field_start.* The *last_character* piece, interpreted as a nonnegative decimal integer, shall specify the last character to be used as part of the sort key. If *last_character* evaluates to zero or .*last_character* is omitted, it shall refer to the last character of the field specified by *field_number.*

If the −b option or b type modifier is in effect, characters within a field shall be counted from the first non-<blank> in the field. (This shall apply separately to *first_character* and *last_character* .)

The obsolescent **[** +*pos1* **[**−*pos2* **]** **]** options provide functionality equivalent to the −k *keydef* option. For comparison, the full formats of these options shall be:

+*field0_number* **[**.*first0_character* **]**[*type*] **[**−*field0_number* **[**.*first0_character* **]**[*type*]]
-k *field_number* **[**.*first_character* **]**[*type*]**[**,*field_number* **[**.*last_character* **]**[*type*]]

In the obsolescent form, fields (specified by *field0_number*) and characters within fields (specified by *first0_character*) shall be numbered from zero instead of one. The −*pos2* option shall specify the first character after the sort field instead of the last character in the sort field. (Therefore, *field0_number* and *first0_character* shall be interpreted as nonnegative, instead of positive, decimal integers and there is no need for a specification of a *last_character* -like form.) The optional type modifiers shall be the same in both forms. If .*first0_character* is omitted or *first0_character* evaluates to zero, it shall refer to the first character of the field.

Thus, a the fully specified +*pos1* −*pos2* form:

+*w*.*x* -*y*.*z*

shall be equivalent to:

```
-k w+1.x+1,y.0     (if z == 0)
-k w+1.x+1,y+1.z   (if z > 0)
```

As with the nonobsolescent forms, implementations shall support at least nine occurrences of the +*pos1* option, which shall be significant in command line order.

4.58.8 Exit Status

The sort utility shall exit with one of the following values:

0 All input files were output successfully, or −c was specified and the input file was correctly sorted.

1 Under the −c option, the file was not ordered as specified, or if the −c and −u options were both specified, two input lines were found with equal keys. This exit status shall not be returned if the −c option is not used.

>1 An error occurred.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

12034 ### 4.58.9 Consequences of Errors

12035 Default.

12036 ### 4.58.10 Rationale. *(This subclause is not a part of P1003.2)*

12037 **Examples, Usage**

12038 In the following examples, nonobsolescent and obsolescent ways of specifying sort
12039 keys are given as an aid to understanding the relationship between the two forms.

12040 Either of the following commands sorts the contents of `infile` with the second
12041 field as the sort key:

```
12042     sort -k 2,2 infile
12043     sort +1 -2 infile
```

12044 Either of the following commands sorts, in reverse order, the contents of `infile1`
12045 and `infile2`, placing the output in `outfile` and using the second character of
12046 the second field as the sort key (assuming that the first character of the second
12047 field is the field separator):

```
12048     sort -r -o outfile -k 2.2,2.2 infile1 infile2                    1
12049     sort -r -o outfile +1.1 -1.2 infile1 infile2
```

12050 Either of the following commands sorts the contents of `infile1` and `infile2`
12051 using the second non-`<blank>` character of the second field as the sort key:

```
12052     sort -k 2.2b,2.2b infile1 infile2
12053     sort +1.1b -1.2b  infile1 infile2
```

12054 Either of the following commands prints the System V password file (user data-
12055 base) sorted by the numeric user ID (the third colon-separated field):

```
12056     sort -t : -k 3,3n /etc/passwd
12057     sort -t : +2 -3n /etc/passwd
```

12058 Either of the following commands prints the lines of the already sorted file
12059 `infile`, suppressing all but one occurrence of lines having the same third field:

```
12060     sort -um -k 3.1,3.0 infile
12061     sort -um +2.0 -3.0 infile
```

12062 Examples in some historical documentation state that options `-um` with one input 1
12063 file keep the first in each set of lines with equal keys. This behavior was deemed 2
12064 to be an implementation artifact and was not made standard. 1

12065 The default value for `-t`, `<blank>`, has different properties than, for example, `-t`
12066 `"<space>"`. If a line contains:

```
12067     <space><space>foo
```

12068 the following treatment would occur with default separation versus specifically
12069 selecting a `<space>`:

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

Field	Default	-t "<space>"
1	<space><space>foo	*empty*
2	*empty*	*empty*
3	*empty*	foo

The leading field separator itself is included in a field when −t is not used. For example, this command returns an exit status of zero, meaning the input was already sorted:

```
sort -c -k 2 <<eof
y<tab>b
x<space>a
eof
```

(assuming that <tab> precedes <space> in the current collating sequence). The field separator is not included in a field when it is explicitly set via −t. This is historical practice and allows usage such as

```
sort -t "|" -k 2n <<eof
Atlanta|425022|Georgia
Birmingham|284413|Alabama
Columbia|100385|South Carolina
eof
```

where the second field can be correctly sorted numerically without regard to the nonnumeric field separator.

History of Decisions Made

The −z option was removed; it is not standard practice on most systems, and is inconsistent with using sort to individually sort several files and then merging them together. The previous language appeared to require implementations to determine the proper buffer length during the sort phase of operation, but not during the merge.

The −y option was removed because of nonportability. The −M option, present in System V, was removed because of nonportability in international usage.

An undocumented −T option exists in some implementations. It is used to specify a directory for intermediate files. Implementations are encouraged to support the use of the **TMPDIR** environment variable instead of adding an option to support this functionality.

The −k option was added to satisfy two complaints. First, the zero-based counting used by sort is not consistent with other utility conventions. Second, it did not meet syntax guideline requirements. The one-based counting in this standard was developed from the input provided by several ballot comments, ballot objections, and discussions with users.

The wording in Draft 10 also clarifies that the −b, −d, −f, −i, −n, and −r options have to come before the first sort key specified if they are intended to apply to all specified keys. The way it is described in this standard matches historical practice, not historical documentation. In the nonobsolescent versions, the results are

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

unspecified if these options are specified after a –k option. This will allow implementations to make the options independent of each other when the obsolescent forms are finally dropped (if that ever happens).

Historical documentation indicates that ‘‘setting –n implies –b.’’ The description of –n already states that optional leading <blank>s are tolerated in doing the comparison. If –b is enabled, rather than implied, by –n, this has unusual side effects. When a character offset is used into a column of numbers (e.g., to sort mod 100), that offset will be measured relative to the most significant digit, not to the column. Based upon a recommendation of the author of the original sort utility, the –b implication has been omitted from POSIX.2 and an application wishing to achieve the previously mentioned side effects will have to manually code the –b flag.

4.59 stty — Set the options for a terminal

4.59.1 Synopsis

```
stty [ –a| –g ]
stty operands
```

4.59.2 Description

The stty utility shall set or report on terminal I/O characteristics for the device that is its standard input. Without options or operands specified, it shall report the settings of certain characteristics, usually those that differ from implementation-defined defaults. Otherwise, it shall modify the terminal state according to the specified operands. Detailed information about the modes listed in the first five groups below are described in POSIX.1 {8} Section 7. Operands in the Combination Modes group (see 4.59.4.6) shall be implemented using operands in the previous groups. Some combinations of operands are mutually exclusive on some terminal types; the results of using such combinations are unspecified.

Typical implementations of this utility require a communications line configured to use a POSIX.1 {8} *termios* interface. On systems where none of these lines are available, and on lines not currently configured to support the POSIX.1 {8} termios interface, some of the operands need not affect terminal characteristics.

4.59.3 Options

The stty utility shall conform to the utility argument syntax guidelines described in 2.10.2.

The following options shall be supported by the implementation:

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.