-- --

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All rights reserved as an unpublished work.

This is an unapproved and unpublished IEEE Standards Draft, subject to change.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]

uniform between `getopts` and *getopt* (). Exact duplication of the messages might not be possible, particularly if a utility is built on another system that has a different *getopt* () function, but the messages must have specific information included so that the program name, invalid option character, and type of error can be distinguished by a user.

Only a rare application program will intercept a `getopts` standard error message and want to parse it. Therefore, implementations are free to choose the most usable messages they can devise. The following formats are used by many historical implementations:

> `"%s: illegal option -- %c\n"`, *<program name>* ,
> *<option character>*

> `"%s: option requires an argument -- %c\n"`,
> *<program name>* , *<option character>*

Historical shells with built-in versions of *getopt* () or `getopts` have used different formats, frequently not even indicating the option character found in error.

4.28 `grep` — File pattern searcher

4.28.1 Synopsis

`grep` [`−E` | `−F`] [`−c` | `−l` | `−q`] [`−insvx`] `−e` *pattern_list* . . . [`−f` *pattern_file*] . . . [*file* . . .]

`grep` [`−E` | `−F`] [`−c` | `−l` | `−q`] [`−insvx`] [`−e` *pattern_list*] . . . `−f` *pattern_file* . . . [*file* . . .]

`grep` [`−E` | `−F`] [`−c` | `−l` | `−q`] [`−insvx`] *pattern_list* [*file* . . .]

Obsolescent Versions :

`egrep` [`−c` | `−l`] [`−inv`] `−e` *pattern_list* [*file* . . .]

`egrep` [`−c` | `−l`] [`−inv`] `−f` *pattern_file* [*file* . . .]

`egrep` [`−c` | `−l`] [`−inv`] *pattern_list* [*file* . . .]

`fgrep` [`−c` | `−l`] [`−invx`] `−e` *pattern_list* [*file* . . .]

`fgrep` [`−c` | `−l`] [`−invx`] `−f` *pattern_file* [*file* . . .]

`fgrep` [`−c` | `−l`] [`−invx`] *pattern_list* [*file* . . .]

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

4.28.2 Description

The `grep` utility shall search the input files, selecting lines matching one or more patterns; the types of patterns shall be controlled by the options specified. The patterns are specified by the `–e` option, `–f` option, or the *pattern_list* operand. The *pattern_list* 's value shall consist of one or more patterns separated by `<newline>`s; the *pattern_file* 's contents shall consist of one or more patterns terminated by `<newline>`s. By default, an input line shall be selected if any pattern, treated as an entire basic regular expression (BRE) as described in 2.8.3, matches any part of the line; a null BRE shall match every line. By default, each selected input line shall be written to the standard output.

Regular expression matching shall be based on text lines. Since `<newline>` separates or terminates patterns (see the `–e` and `–f` options below), regular expressions cannot contain a `<newline>` character. Similarly, since patterns are matched against individual lines of the input, there is no way for a pattern to match a `<newline>` found in the input.

A command invoking the (obsolescent) `egrep` utility with the `–e` option specified shall be equivalent to the command:

`grep –E [ –c | –l ] [–inv] –e` *pattern_list* [*file* . . .]

A command invoking the `egrep` utility with the `–f` option specified shall be equivalent to the command:

`grep –E [ –c | –l ] [–inv] –f` *pattern_file* [*file* . . .]

A command invoking the `egrep` utility with the *pattern_list* specified shall be equivalent to the command:

`grep –E [ –c | –l ] [–inv]` *pattern_list* [*file* . . .]

A command invoking the (obsolescent) `fgrep` utility with the `–e` option specified shall be equivalent to the command:

`grep –F [ –c | –l ] [–invx] –e` *pattern_list* [*file* . . .]

A command invoking the `fgrep` utility with the `–f` option specified shall be equivalent to the command:

`grep –F [ –c | –l ] [–invx] –f` *pattern_file* [*file* . . .]

A command invoking the `fgrep` utility with the *pattern_list* operand specified shall be equivalent to the command:

`grep –F [ –c | –l ] [–invx]` *pattern_list* [*file* . . .]

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

4.28.3 Options

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

The following options shall be supported by the implementation:

−E Match using extended regular expressions. Treat each pattern specified as an ERE, as described in 2.8.4. If any entire ERE pattern matches an input line, the line shall be matched. A null ERE shall match every line.

−F Match using fixed strings. Treat each pattern specified as a string instead of a regular expression. If an input line contains any of the patterns as a contiguous sequence of bytes, the line shall be matched. A null string shall match every line.

−c Write only a count of selected lines to standard output.

−e *pattern_list*
  Specify one or more patterns to be used during the search for input. Patterns in *pattern_list* shall be separated by a <newline>. A null pattern can be specified by two adjacent <newline>s in *pattern_list* ; in the obsolescent forms, adjacent <newline>s in *pattern_list* produce undefined results. Unless the −E or −F option is also specified, each pattern shall be treated as a BRE, as described in 2.8.3. In the nonobsolescent forms, multiple −e and −f options shall be accepted by the grep utility. All of the specified patterns shall be used when matching lines, but the order of evaluation is unspecified.

−f *pattern_file*
  Read one or more patterns from the file named by the pathname *pattern_file.* Patterns in *pattern_file* shall be terminated by a <newline>. A null pattern can be specified by an empty line in *pattern_file.* Unless the −E or −F option is also specified, each pattern shall be treated as a BRE, as described in 2.8.3.

−i Perform pattern matching in searches without regard to case. See 2.8.2.

−l (The letter ell.) Write only the names of files containing selected lines to standard output. Pathnames shall be written once per file searched. If the standard input is searched, a pathname of "(standard input)" shall be written, in the POSIX Locale. In other locales, standard input may be replaced by something more appropriate in those locales.

−n Precede each output line by its relative line number in the file, each file starting at line 1. The line number counter shall be reset for each file processed.

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

`−q` Quiet. Do not write anything to the standard output, regardless of matching lines. Exit with zero status if an input line is selected.

`−s` Suppress the error messages ordinarily written for nonexistent or unreadable files. Other error messages shall not be suppressed.

`−v` Select lines not matching any of the specified patterns. If the `−v` option is not specified, selected lines shall be those that match any of the specified patterns.

`−x` Consider only input lines that use all characters in the line to match an entire fixed string or regular expression to be matching lines.

4.28.4 Operands

The following operands shall be supported by the implementation:

pattern Specify one or more patterns to be used during the search for input. This operand shall be treated as if it were specified as `−e` *pattern_list* (see 4.28.3).

file A pathname of a file to be searched for the pattern(s). If no *file* operands are specified, the standard input shall be used.

4.28.5 External Influences

4.28.5.1 Standard Input

The standard input shall be used only if no *file* operands are specified. See Input Files.

4.28.5.2 Input Files

The input files shall be text files.

4.28.5.3 Environment Variables

The following environment variables shall affect the execution of `grep`:

LANG This variable shall determine the locale to use for the locale categories when both **LC_ALL** and the corresponding environment variable (beginning with **LC_**) do not specify a locale. See 2.6.

LC_ALL This variable shall determine the locale to be used to override any values for locale categories specified by the settings of **LANG** or any environment variables beginning with **LC_**.

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

LC_COLLATE This variable shall determine the locale for the behavior of ranges, equivalence classes, and multicharacter collating elements within regular expressions.

LC_CTYPE This variable shall determine the locale for the interpretation of sequences of bytes of text data as characters (e.g., single- versus multibyte characters in arguments) and the behavior of character classes within regular expressions.

LC_MESSAGES This variable shall determine the language in which messages should be written.

4.28.5.4 Asynchronous Events

Default.

4.28.6 External Effects

4.28.6.1 Standard Output

If the `−l` option is in effect, and the `−q` option is not, a single output line shall be written for each file containing at least one selected input line:

`"%s\n"`, *file*

Otherwise, if more than one *file* argument appears, and `−q` is not specified, the `grep` utility shall prefix each output line by:

`"%s:"`, *file*

The remainder of each output line shall depend on the other options specified:

— If the `−c` option is in effect, the remainder of each output line shall contain:

`"%d\n"`, *<count>*

— Otherwise, if `−c` is not in effect and the `−n` option is in effect, the following shall be written to standard output:

`"%d:"`, *<line number>*

— Finally, the following shall be written to standard output:

`"%s"`, *<selected-line contents>*

4.28.6.2 Standard Error

Used only for diagnostic messages.

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

4.28.6.3 Output Files

None.

4.28.7 Extended Description

None.

4.28.8 Exit Status

The `grep` utility shall exit with one of the following values:

- 0 One or more lines were selected.
- 1 No lines were selected.
- >1 An error occurred.

4.28.9 Consequences of Errors

If the `-q` option is specified, the exit status shall be zero if an input line is selected, even if an error was detected. Otherwise, default actions shall be performed.

4.28.10 Rationale. *(This subclause is not a part of P1003.2)*

Examples, Usage

This `grep` has been enhanced in an upward-compatible way to provide the exact functionality of the historical `egrep` and `fgrep` commands as well. It was the clear intention of the working group to consolidate the three `grep`s into a single command.

The old `egrep` and `fgrep` commands are likely to be supported for many years to 1
come as implementation extensions, allowing existing applications to operate unmodified.

To find all uses of the word `Posix` (in any case) in the file `text.mm`, and write with line numbers:

```
grep -i -n posix text.mm
```

To find all empty lines in the standard input: 2

```
grep ^$
   or
grep -v .
```

Both of the following commands print all lines containing strings `abc` or `def` or both:

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

```
grep -E 'abc
def'
```

```
grep -F 'abc
def'
```

Both of the following commands print all lines matching exactly `abc` or `def`:

```
grep -E '^abc$
^def$'
```

```
grep -F -x 'abc
def'
```

History of Decisions Made

The `−e` *pattern_list* option has the same effect as the *pattern_list* operand, but is useful when *pattern_list* begins with the hyphen delimiter. It is also useful when it is more convenient to provide multiple patterns as separate arguments.

Earlier drafts did not show that the `−c`, `−l`, and `−q` options were mutually exclusive. This has been fixed to more closely align with historical practice and documentation.

Historical implementations usually silently ignored all but one of multiply specified `−e` and `−f` options, but were not consistent as to which specification was actually used.

POSIX.2 requires that the nonobsolescent forms accept multiple `−e` and `−f` options and use all of the patterns specified while matching input text lines. [Note that the order of evaluation is not specified. If an implementation finds a null string as a pattern, it is allowed to use that pattern first (matching every line) and effectively ignore any other patterns.]

The `−b` option was removed from the Options subclause, since block numbers are implementation dependent.

The System V restriction on using `−` to mean standard input was lifted.

A definition of action taken when given a null RE or ERE is specified. This is an error condition in some historical implementations.

The `−l` option previously indicated that its use was undefined when no files were explicitly named. This behavior was historical and placed an unnecessary restriction on future implementations. It has been removed.

The `−q` option was added at the suggestion of members of the balloting group as a means of easily determining whether or not a pattern (or string) exists in a group of files. When searching several files, it provides a performance improvement (because it can quit as soon as it finds the first match) and requires less care by the user in choosing the set of files to supply as arguments (because it will exit zero if it finds a match even if `grep` detected an access or read error on earlier file operands).

The historical BSD `grep −s` option practice is easily duplicated by redirecting standard output to `/dev/null`. The `−s` option required here is from System V.

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

The `−x` option, historically available only with `fgrep`, is available here for all of the nonobsolescent versions.

4.29 head — Copy the first part of files

4.29.1 Synopsis

`head [−n` *number* `] [`*file* `... ]`

Obsolescent version:

`head [−`*number* `] [`*file* `... ]`

4.29.2 Description

The `head` utility shall copy its input files to the standard output, ending the output for each file at a designated point.

Copying shall end at the point in each input file indicated by the `−n` *number* option (or the obsolescent version's −*number* argument). The option-argument *number* shall be counted in units of lines.

4.29.3 Options

The `head` utility shall conform to the utility argument syntax guidelines described in standard described in 2.10.2, except that the obsolescent version accepts multicharacter numeric options.

The following option shall be supported by the implementation in the nonobsolescent version:

`−n` *number* The first *number* lines of each input file shall be copied to standard output. The *number* option argument shall be a positive decimal integer.

If no options are specified, `head` shall act as if `-n 10` had been specified.

In the obsolescent version, the following option shall be supported by the implementation:

−*number* The *number* argument is a positive decimal integer with the same effect as the `−n`*number* option in the nonobsolescent version.

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