

PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

NAME

`grep` – search a file for a pattern

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...]
```

DESCRIPTION

The *grep* utility shall search the input files, selecting lines matching one or more patterns; the types of patterns are 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>. By default, an input line shall be selected if any pattern, treated as an entire basic regular expression (BRE) as described in the Base Definitions volume of IEEE Std 1003.1-2001, Section 9.3, Basic Regular Expressions, matches any part of the line excluding the terminating <newline>; 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 a <newline> separates or terminates patterns (see the **-e** and **-f** options below), regular expressions cannot contain a <newline>. Similarly, since patterns are matched against individual lines (excluding the terminating <newline>s) of the input, there is no way for a pattern to match a <newline> found in the input.

OPTIONS

The *grep* utility shall conform to the Base Definitions volume of IEEE Std 1003.1-2001, Section 12.2, Utility Syntax Guidelines.

The following options shall be supported:

- E** Match using extended regular expressions. Treat each pattern specified as an ERE, as described in the Base Definitions volume of IEEE Std 1003.1-2001, Section 9.4, Extended Regular Expressions. If any entire ERE pattern matches some part of an input line excluding the terminating <newline>, 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. The application shall ensure that patterns in *pattern_list* are separated by a <newline>. A null pattern can be specified by two adjacent <newline>s in *pattern_list*. Unless the **-E** or **-F** option is also specified, each pattern shall be treated as a BRE, as described in the Base Definitions volume of IEEE Std 1003.1-2001, Section 9.3, Basic Regular Expressions. 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 the Base Definitions volume of IEEE Std 1003.1-2001, Section 9.3, Basic Regular Expressions.

- i** Perform pattern matching in searches without regard to case; see the Base Definitions volume of IEEE Std 1003.1-2001, Section 9.2, Regular Expression General Requirements.
- 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.
- q** Quiet. Nothing shall be written 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 excluding the terminating <new-line> to match an entire fixed string or regular expression to be matching lines.

OPERANDS

The following operands shall be supported:

pattern_list

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*.

file

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

STDIN

The standard input shall be used only if no *file* operands are specified. See the INPUT FILES section.

INPUT FILES

The input files shall be text files.

ENVIRONMENT VARIABLES

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

LANG Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)

LC_ALL

If set to a non-empty string value, override the values of all the other internationalization variables.

LC_COLLATE

Determine the locale for the behavior of ranges, equivalence classes, and multi-character collating elements within regular expressions.

LC_CTYPE

Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments and input files) and the behavior of character classes within regular expressions.

LC_MESSAGES

Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.

NLSPATH

Determine the location of message catalogs for the processing of *LC_MESSAGES*.

ASYNCHRONOUS EVENTS

Default.

STDOUT

If the **-l** option is in effect, and the **-q** option is not, the following 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>`

STDERR

The standard error shall be used only for diagnostic messages.

OUTPUT FILES

None.

EXTENDED DESCRIPTION

None.

EXIT STATUS

The following exit values shall be returned:

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

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.

The following sections are informative.

APPLICATION USAGE

Care should be taken when using characters in *pattern_list* that may also be meaningful to the command interpreter. It is safest to enclose the entire *pattern_list* argument in single quotes:

'...'

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.

Multiple **-e** and **-f** options are accepted and *grep* uses all of the patterns it is given 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 **-q** option provides 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 exits zero if it finds a match even if *grep* detected an access or read error on earlier *file* operands).

EXAMPLES

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

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

2. To find all empty lines in the standard input:

```
grep ^$
```

or:

```
grep -v .
```

3. Both of the following commands print all lines containing strings "abc" or "def" or both:

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

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

4. Both of the following commands print all lines matching exactly "abc" or "def" :

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

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

RATIONALE

This *grep* has been enhanced in an upwards-compatible way to provide the exact functionality of the historical *egrep* and *fgrep* commands as well. It was the clear intention of the standard developers to consolidate the three *greps* into a single command.

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

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.

The **-b** option was omitted from the OPTIONS section because block numbers are implementation-defined.

The System V restriction on using `-` to mean standard input was omitted.

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

The `-I` 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 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.

The `-x` option, historically available only with `fgrep`, is available here for all of the non-obsolete versions.

FUTURE DIRECTIONS

None.

SEE ALSO

sed

COPYRIGHT

Portions of this text are reprinted and reproduced in electronic form from IEEE Std 1003.1, 2003 Edition, Standard for Information Technology -- Portable Operating System Interface (POSIX), The Open Group Base Specifications Issue 6, Copyright (C) 2001-2003 by the Institute of Electrical and Electronics Engineers, Inc and The Open Group. In the event of any discrepancy between this version and the original IEEE and The Open Group Standard, the original IEEE and The Open Group Standard is the referee document. The original Standard can be obtained online at <http://www.open-group.org/unix/online.html>.