NAME

grep, **egrep**, **fgrep**, **rgrep** - file pattern searcher

SYNOPSIS

grep [**-abcdDEFGHhIiLlmnOopqRSsUVvwxz**] [**-A** *num*] [**-B** *num*] [**-C** *num*] [**-e** *pattern*] [**-f** *file*]
[**--binary-files**=*value*] [**--color**[=*when*]] [**--colour**[=*when*]] [**--context**=*num*] [**--label**]
[**--line-buffered**] [**--null**] [*pattern*] [*file ...*]

DESCRIPTION

The **grep** utility searches any given input files, selecting lines that match one or more patterns. By default, a pattern matches an input line if the regular expression (RE) in the pattern matches the input line without its trailing newline. An empty expression matches every line. Each input line that matches at least one of the patterns is written to the standard output.

grep is used for simple patterns and basic regular expressions (BREs); **egrep** can handle extended regular expressions (EREs). See re_format(7) for more information on regular expressions. **fgrep** is quicker than both **grep** and **egrep**, but can only handle fixed patterns (i.e., it does not interpret regular expressions). Patterns may consist of one or more lines, allowing any of the pattern lines to match a portion of the input.

The following options are available:

-A *num*, **--after-context**=*num*
> Print *num* lines of trailing context after each match. See also the **-B** and **-C** options.

-a, **--text**
> Treat all files as ASCII text. Normally **grep** will simply print "Binary file ... matches" if files contain binary characters. Use of this option forces **grep** to output lines matching the specified pattern.

-B *num*, **--before-context**=*num*
> Print *num* lines of leading context before each match. See also the **-A** and **-C** options.

-b, **--byte-offset**
> The offset in bytes of a matched pattern is displayed in front of the respective matched line.

-C *num*, **--context**=*num*
> Print *num* lines of leading and trailing context surrounding each match. See also the **-A** and **-B** options.

-c, **--count**

> Only a count of selected lines is written to standard output.

--colour=[*when*], **--color**=[*when*]

> Mark up the matching text with the expression stored in the GREP_COLOR environment variable. The possible values of *when* are "**never**", "**always**" and "**auto**".

-D *action*, **--devices**=*action*

> Specify the demanded *action* for devices, FIFOs and sockets. The default *action* is "**read**", which means, that they are read as if they were normal files. If the *action* is set to "**skip**", devices are silently skipped.

-d *action*, **--directories**=*action*

> Specify the demanded *action* for directories. It is "**read**" by default, which means that the directories are read in the same manner as normal files. Other possible values are "**skip**" to silently ignore the directories, and "**recurse**" to read them recursively, which has the same effect as the **-R** and **-r** option.

-E, **--extended-regexp**

> Interpret *pattern* as an extended regular expression (i.e., force **grep** to behave as **egrep**).

-e *pattern*, **--regexp**=*pattern*

> Specify a *pattern* used during the search of the input: an input line is selected if it matches any of the specified patterns. This option is most useful when multiple **-e** options are used to specify multiple patterns, or when a *pattern* begins with a dash ('-').

--exclude *pattern*

> If specified, it excludes files matching the given filename *pattern* from the search. Note that **--exclude** and **--include** patterns are processed in the order given. If a name matches multiple patterns, the latest matching rule wins. If no **--include** pattern is specified, all files are searched that are not excluded. Patterns are matched to the full path specified, not only to the filename component.

--exclude-dir *pattern*

> If **-R** is specified, it excludes directories matching the given filename *pattern* from the search. Note that **--exclude-dir** and **--include-dir** patterns are processed in the order given. If a name matches multiple patterns, the latest matching rule wins. If no **--include-dir** pattern is specified, all directories are searched that are not excluded.

-F, **--fixed-strings**

Interpret *pattern* as a set of fixed strings (i.e., force **grep** to behave as **fgrep**).

-f *file*, **--file=***file*

Read one or more newline separated patterns from *file*. Empty pattern lines match every input line. Newlines are not considered part of a pattern. If *file* is empty, nothing is matched.

-G, **--basic-regexp**

Interpret *pattern* as a basic regular expression (i.e., force **grep** to behave as traditional **grep**).

-H   Always print filename headers with output lines.

-h, **--no-filename**

Never print filename headers (i.e., filenames) with output lines.

--help Print a brief help message.

-I   Ignore binary files. This option is equivalent to the "**--binary-files=without-match**" option.

-i, **--ignore-case**

Perform case insensitive matching. By default, **grep** is case sensitive.

--include *pattern*

If specified, only files matching the given filename *pattern* are searched. Note that **--include** and **--exclude** patterns are processed in the order given. If a name matches multiple patterns, the latest matching rule wins. Patterns are matched to the full path specified, not only to the filename component.

--include-dir *pattern*

If **-R** is specified, only directories matching the given filename *pattern* are searched. Note that **--include-dir** and **--exclude-dir** patterns are processed in the order given. If a name matches multiple patterns, the latest matching rule wins.

-L, **--files-without-match**

Only the names of files not containing selected lines are written to standard output. Pathnames are listed once per file searched. If the standard input is searched, the string "(standard input)" is written unless a **--label** is specified.

-l, **--files-with-matches**

Only the names of files containing selected lines are written to standard output. **grep** will only search a file until a match has been found, making searches potentially less expensive.

Pathnames are listed once per file searched. If the standard input is searched, the string "(standard input)" is written unless a **--label** is specified.

--label

Label to use in place of "(standard input)" for a file name where a file name would normally be printed. This option applies to **-H**, **-L**, and **-l**.

--mmap

Use mmap(2) instead of read(2) to read input, which can result in better performance under some circumstances but can cause undefined behaviour.

-m *num*, **--max-count**=*num*

Stop reading the file after *num* matches.

-n, **--line-number**

Each output line is preceded by its relative line number in the file, starting at line 1. The line number counter is reset for each file processed. This option is ignored if **-c**, **-L**, **-l**, or **-q** is specified.

--null   Prints a zero-byte after the file name.

-O   If **-R** is specified, follow symbolic links only if they were explicitly listed on the command line. The default is not to follow symbolic links.

-o, **--only-matching**

Prints only the matching part of the lines.

-p   If **-R** is specified, no symbolic links are followed. This is the default.

-q, **--quiet**, **--silent**

Quiet mode: suppress normal output. **grep** will only search a file until a match has been found, making searches potentially less expensive.

-R, **-r**, **--recursive**

Recursively search subdirectories listed. (i.e., force **grep** to behave as **rgrep**).

-S   If **-R** is specified, all symbolic links are followed. The default is not to follow symbolic links.

-s, **--no-messages**

Silent mode. Nonexistent and unreadable files are ignored (i.e., their error messages are

suppressed).

-U, **--binary**
> Search binary files, but do not attempt to print them.

-u   This option has no effect and is provided only for compatibility with GNU grep.

-V, **--version**
> Display version information and exit.

-v, **--invert-match**
> Selected lines are those *not* matching any of the specified patterns.

-w, **--word-regexp**
> The expression is searched for as a word (as if surrounded by '[[:<:]]' and '[[:>:]]'; see
> re_format(7)). This option has no effect if **-x** is also specified.

-x, **--line-regexp**
> Only input lines selected against an entire fixed string or regular expression are considered to be
> matching lines.

-y   Equivalent to **-i**. Obsoleted.

-z, **--null-data**
> Treat input and output data as sequences of lines terminated by a zero-byte instead of a newline.

--binary-files=*value*
> Controls searching and printing of binary files. Options are:
> **binary** (default) Search binary files but do not print them.
> **without-match**   Do not search binary files.
> **text**   Treat all files as text.

--line-buffered
> Force output to be line buffered. By default, output is line buffered when standard output is a
> terminal and block buffered otherwise.

If no file arguments are specified, the standard input is used. Additionally, "**-**" may be used in place of a
file name, anywhere that a file name is accepted, to read from standard input. This includes both **-f** and
file arguments.

ENVIRONMENT

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

GREP_COLOR   This variable specifies the color used to highlight matched (non-empty) text.

GREP_OPTIONS This variable specifies default options to be placed in front of any explicit options. It may cause problems with portable scripts.

TERM   This variable specifies the type name of the terminal, console or display-device type to be used. See term(7).

EXIT STATUS

The **grep** utility exits with one of the following values:

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

EXAMPLES

- Find all occurrences of the pattern 'patricia' in a file:

      $ grep 'patricia' myfile

- Same as above but looking only for complete words:

      $ grep -w 'patricia' myfile

- Count occurrences of the exact pattern 'FOO' :

      $ grep -c FOO myfile

- Same as above but ignoring case:

      $ grep -c -i FOO myfile

- Find all occurrences of the pattern '.Pp' at the beginning of a line:

      $ grep '^\.Pp' myfile

The apostrophes ensure the entire expression is evaluated by **grep** instead of by the user's shell. The

caret '^' matches the null string at the beginning of a line, and the '\' escapes the '.', which would otherwise match any character.

-   Find all lines in a file which do not contain the words 'foo' or 'bar':

        $ grep -v -e 'foo' -e 'bar' myfile

-   Peruse the file 'calendar' looking for either 19, 20, or 25 using extended regular expressions:

        $ egrep '19|20|25' calendar

-   Show matching lines and the name of the '*.h' files which contain the pattern 'FIXME'. Do the search recursively from the */usr/src/sys/arm* directory

        $ grep -H -R FIXME --include="*.h" /usr/src/sys/arm/

-   Same as above but show only the name of the matching file:

        $ grep -l -R FIXME --include="*.h" /usr/src/sys/arm/

-   Show lines containing the text 'foo'. The matching part of the output is colored and every line is prefixed with the line number and the offset in the file for those lines that matched.

        $ grep -b --colour -n foo myfile

-   Show lines that match the extended regular expression patterns read from the standard input:

        $ echo -e 'Free\nBSD\nAll.*reserved' | grep -E -f - myfile

-   Show lines from the output of the pciconf(8) command matching the specified extended regular expression along with three lines of leading context and one line of trailing context:

        $ pciconf -lv | grep -B3 -A1 -E 'class.*=.*storage'

-   Suppress any output and use the exit status to show an appropriate message:

        $ grep -q foo myfile && echo File matches

SEE ALSO
ed(1), ex(1), sed(1), zgrep(1), re_format(7)

STANDARDS

The **grep** utility is compliant with the IEEE Std 1003.1-2008 ("POSIX.1") specification.

The flags [**-AaBbCDdGHhILmopRSUVw**] are extensions to that specification, and the behaviour of the **-f** flag when used with an empty pattern file is left undefined.

All long options are provided for compatibility with GNU versions of this utility.

Historic versions of the **grep** utility also supported the flags [**-ruy**]. This implementation supports those options; however, their use is strongly discouraged.

HISTORY

The **grep** command first appeared in Version 6 AT&T UNIX.