NAME

grep – search a file for a pattern

SYNOPSIS

/usr/bin/grep [**–bchilnsvw**] *limited-regular-expression* [*filename* . . .]

/usr/xpg4/bin/grep [**–E** | **–F**] [**–c** | **–l** | **–q**] [**–bhinsvwx**] **–e** *pattern_list* . . .
[**–f** *pattern_file*] . . . [*file* . . .]
/usr/xpg4/bin/grep [**–E** | **–F**] [**–c** | **–l** | **–q**] [**–bhinsvwx**] [**–e** *pattern_list* . . .]
–f *pattern_file* . . . [*file* . . .]
/usr/xpg4/bin/grep [**–E** | **–F**] [**–c** | **–l** | **–q**] [**–bhinsvwx**] *pattern* [*file* . . .]

AVAILABILITY

/usr/bin/grep

SUNWcsu

/usr/xpg4/bin/grep

SUNWxcu4

DESCRIPTION

The **grep** command searches files for a pattern and prints all lines that contain that pattern. It uses a compact non-deterministic algorithm.

Be careful using the characters **$**, ∗, **[**, **^**, **|** , **(**, **)**, and **** in the *pattern_list* because they are also meaningful to the shell. It is safest to enclose the entire *pattern_list* in single quotes ′ . . . ′.

If no files are specified, **grep** assumes standard input. Normally, each line found is copied to standard output. The file name is printed before each line found if there is more than one input file.

/usr/bin/grep

grep uses limited regular expressions like those described on the **regexp**(5) manual page to match the patterns.

/usr/xpg4/bin/grep

The options –**E** and –**F** affect the way **grep** interprets *pattern_list*. If –**E** is specified, **grep** interprets *pattern_list* as a full regular expression (see –**E** for description). If –**F** is specified, **grep** interprets *pattern_list* as a fixed string. If neither are specified, **grep** interprets *pattern_list* as a basic regular expression as described on **regex**(5) manual page.

OPTIONS

The following options are supported:

–b Precede each line by the block number on which it was found. This can be useful in locating block numbers by context (first block is 0).

–c Print only a count of the lines that contain the pattern.

–h Prevents the name of the file containing the matching line from being appended to that line. Used when searching multiple files.

–i Ignore upper/lower case distinction during comparisons.

–l Print only the names of files with matching lines, separated by NEWLINE characters. Does not repeat the names of files when the pattern is found more than once.

–n Precede each line by its line number in the file (first line is 1).

−s Suppress error messages about nonexistent or unreadable files

−v Print all lines except those that contain the pattern.

−w Search for the expression as a word as if surrounded by **\<** and **\>**.

/usr/xpg4/bin/grep

The following options are supported by **/usr/xpg4/bin/grep** only:

−e *pattern_list* Specify one or more patterns to be used during the search for input. Patterns in *pattern_list* must be separated by a NEWLINE character. A null pattern can be specified by two adjacent newline characters in *pattern_list*. Unless the −**E** or −**F** option is also specified, each pattern will be treated as a basic regular expression. Multiple −**e** and −**f** options are accepted by **grep**. All of the specified patterns are used when matching lines, but the order of evaluation is unspecified.

−E Match using full regular expressions. Treat each pattern specified as an full regular expression. If any entire full regular expression pattern matches an input line, the line will be matched. A null full regular expression matches every line.

Each pattern will be interpreted as a full regular expression as described on the **regex**(5) manual page, except for **\(** and **\)**, and including:

1. A full regular expression followed by **+** that matches one or more occurrences of the full regular expression.
2. A full regular expression followed by **?** that matches 0 or 1 occurrences of the full regular expression.
3. Full regular expressions separated by | or by a new-line that match strings that are matched by any of the expressions.
4. A full regular expression that may be enclosed in parentheses **( )** for grouping.

The order of precedence of operators is **[ ]**, then ∗ **? +**, then concatenation, then | and new-line.

−f *pattern_file* Read one or more patterns from the file named by the path name *pattern_file*. Patterns in *pattern_file* are terminated by a NEWLINE character. A null pattern can be specified by an empty line in *pattern_file*. Unless the −**E** or −**F** option is also specified, each pattern will be treated as a basic regular expression.

−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 will be matched. A null string matches every line. See **fgrep**(1) for more information.

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

−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.

OPERANDS

The following operands are supported:

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

/usr/bin/grep

pattern Specify a pattern to be used during the search for input.

/usr/xpg4/bin/grep

pattern Specify one or more patterns to be used during the search for input. This operand is treated as if it were specified as **−e** *pattern_list*.

USAGE

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** will use 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 will exit zero if it finds a match even if **grep** detected an access or read error on earlier file operands).

EXAMPLES

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

example% /usr/bin/grep -i -n posix text.mm

To find all empty lines in the standard input:

example% /usr/bin/grep ^$

or

example% /usr/bin/grep −v .

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

example% /usr/xpg4/bin/grep -E 'abc
def'

example% /usr/xpg4/bin/grep -F 'abc
def'

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

example% /usr/xpg4/bin/grep -E '^abc$
^def$'

example% /usr/xpg4/bin/grep -F -x 'abc
def'

ENVIRONMENT

See **environ**(5) for descriptions of the following environment variables that affect the execution of **grep**: **LC_COLLATE**, **LC_CTYPE**, **LC_MESSAGES**, and **NLSPATH**.

EXIT STATUS

The following exit values are returned:

0 one or more matches were found

1 no matches were found

2 syntax errors or inaccessible files (even if matches were found).

SEE ALSO

egrep(1), **fgrep**(1), **sed**(1), **sh**(1), **environ**(5), **regex**(5), **regexp**(5)

NOTES

Lines are limited to **BUFSIZ** characters; longer lines are truncated. **BUFSIZ** is defined in **/usr/include/stdio.h**. If there is a line with embedded nulls, **grep** will only match up to the first null; if it matches, it will print the entire line.