

A Grip on Grep

- [Swiss Army Knife considered harmful](#)
- [Filters](#)

Swiss Army Knife considered harmful

grep

Filters

Overview

Filters read input line by line and match each line against a set of pattern.

Filters serve two purposes:

1. They detect if a line matches the pattern set
2. They provide information about matched lines on output

Filters exit with code 0 (success) if any line matched any pattern. If no line matched the exit code is 1 (failure), on error a code >1 is returned. The caller can request to invert the match logic: finding a match is a failure, not finding one as success.

Types of information provided by filters:

- Default: the list of matched/unmatched lines
- The number of matched/unmatched lines
- The number of matched strings
- The line number (1 based) of each matched/unmatched line
- The byte offset of each match inside the file
- The set of matched strings in the line.
- The position of each match on its line
- The timestamp of arrival of each match/line

Note: already implemented features are marked in bold.

Filter micro languages

- POSIX
 - **bre** .. basic regular expressions
 - **ere** .. extended regular expressions
 - **fixed** .. fixed string
 - **pat** .. glob pattern matching (fnmatch)
-

GNU

- pcre .. Perl regular expressions
- Other
 - **mull** .. multilog/svlogd
 - pattern .. Loki "<_>" syntax

Match Information Formats

By default, filters output all matching lines unaltered.

Another simple case is the number of matching lines: if requested, the filter only outputs this information as decimal number.

Any other information is output in the following way:

- unmatched lines are prefixed with a '-' character (minus).
- matched lines are prefixed with a '+' character
- Any other information is output as on lines which are prefixed by a '#' character

For matched/unmatched lines, the prefix is followed immediately by the 1 based line number in decimal notation, followed by a colon (':'), followed by the zero based byte offset of the line in decimal notation, followed by a colon.

If match string information is requested, each matched line is followed by the list of matched strings in order of detection, each match on a successive line with a '#' prefix.

The format of the match string report is:

```
#line: id: string: index: len: offset: num: type: pattern
```

Where:

- *line* .. 1 based line number since start of the input
- *id* .. the identifier of the pattern, if any, which matched the string
- *string* .. the matched string
- *index* .. byte offset of the match within the line
- *len* .. byte count of the matched string
- *offset* .. byte offset of the match since start of the input
- *num* .. the number of the pattern which matched the string
- *type* .. the pattern type which matched the string
- *pattern* .. the pattern which matched the string

If *string* or *pattern* contain a colon or a backslash, they are escaped with a backslash.

`|id|` can be the sequence number of the match in decimal notation.

All fields after `|string|` are optional.

Command Line Interface

```
filter [output_type] [behavior] [output_options] [filter_options] [pattern ..]
```

```
filter [-c|-q] [-v][-m n][-0] [-n][-N][-p prefix][-z] [-i][-w][-x] [pattern ..]
```

output type:

- no option at all: output selected lines
- `-c` .. only output count
- `-q` .. no output.

behavior:

- `-v` .. invert selection: unmatched lines are selected, counted and with no match exit is successful
- `-m n` ... stop after *n* matches.
- `-0` .. Split input by `|NUL|` bytes instead of newlines.

output options:

- `-p prefix` .. prepend all printed lines with *prefix*, followed by a `:` if printed. With `-q` print prefix if a match is found, with `-c` prefix the the count with prefix and a `:`.
- `-n` .. print the line number followed by a `:` before the line (and after the prefix)
- `-N` .. print a `+` for selected lines, a `-` for not selected lines, then, if `-p` is given the prefix with a `:`, then the line number with a `:` and finally the line. All lines are printed, further processing must be done on output (e.g. to print context lines around the matches).
- `-z` separate output lines with `|NUL|` bytes instead of newline.

filter options:

A filter does not need to implement any option. The following option names are reserved for the respective functionality

- `-i` .. case insensitive match
- `-w` .. only match whole words
- `-x` .. only match whole lines

All patterns are specified as arguments on the command line. If no pattern is given, nothing is selected. The filter implementation must specify clearly the meaning of an empty pattern. Some filters don't match any string, others match everything in this case.