

NAME ln - make links between files

SYNOPSIS

ln [OPTION]... [-T] TARGET LINK_NAME
ln [OPTION]... TARGET
ln [OPTION]... TARGET... DIRECTORY
ln [OPTION]... -t DIRECTORY TARGET...

DESCRIPTION

In the 1st form, create a link to TARGET with the name LINK_NAME. In the 2nd form, create a link to TARGET in the current directory. In the 3rd and 4th forms, create links to each TARGET in

DIRECTORY. Create hard links by default, symbolic links with **--symbolic**. By default, each destination (name of new link) should not already exist. When creating hard links, each TARGET must exist. Symbolic links can hold arbitrary text; if later resolved, a relative link is interpreted in relation to its parent directory.

Mandatory arguments to long options are mandatory for short options too.

--backup[=CONTROL] - make a backup of each existing destination file

-b like **--backup** but does not accept an argument

-d, -F, --directory - allow the superuser to attempt to hard link directories (note: will probably fail due to system restrictions, even for the superuser)

-f, --force - remove existing destination files

-i, --interactive - prompt whether to remove destinations

-L, --logical - dereference TARGETs that are symbolic links

-n, --no-dereference - treat LINK_NAME as a normal file if it is a symbolic link to a directory

-P, --physical - make hard links directly to symbolic links

-r, --relative - with **-s**, create links relative to link location

-s, --symbolic - make symbolic links instead of hard links

-S, --suffix=SUFFIX - override the usual backup suffix

-t, --target-directory=DIRECTORY - specify the DIRECTORY in which to create the links

-T, --no-target-directory - treat LINK_NAME as a normal file always

-v, --verbose - print name of each linked file

--help display this help and exit

--version - output version information and exit

The backup suffix is '~', unless set with **--suffix** or SIMPLE_BACKUP_SUFFIX. The version control method may be selected via the **--backup** option or through the VERSION_CONTROL environment variable. Here are the values:

none, off: never make backups (even if **--backup** is given)

numbered, t: make numbered backups

existing, nil: numbered if numbered backups exist, simple otherwise

simple, never: always make simple backups

Using **-s** ignores **-L** and **-P**. Otherwise, the last option specified controls behavior when a TARGET is a symbolic link, defaulting to **-P**.

NAME sed - stream editor for filtering and transforming text

SYNOPSIS

sed [**OPTION**]... {**script-only-if-no-other-script**} [**input-file**]...

DESCRIPTION

Sed is a stream editor. A stream editor is used to perform basic text transformations on an input stream (a file or input from a pipeline). While in some ways similar to an editor which permits scripted edits (such as **ed**), **sed** works by making only one pass over the input(s), and is consequently more efficient. But it is **sed**'s ability to filter text in a pipeline which particularly distinguishes it from other types of editors.

-n, **--quiet**, **--silent** - suppress automatic printing of pattern space

-e script, **--expression=script** - add the script to the commands to be executed **-f script-file**, **--file=script-file** - add the contents of script-file to the commands to be executed **--follow-symlinks** - follow symlinks when processing in place **-i[SUFFIX]**, **--in-place[=SUFFIX]** - edit files in place (makes backup if SUFFIX supplied) **-l N**, **--line-length=N** - specify the desired line-wrap length for the 'l' command **--posix** disable all GNU extensions. **-E**, **-r**, **--regexp-extended** use extended regular expressions in the script (for portability use POSIX **-E**).

-s, **--separate** consider files as separate rather than as a single, continuous long stream.

--sandbox operate in sandbox mode (disable e/r/w commands).

-u, **--unbuffered** load minimal amounts of data from the input files and flush the output buffers more often

-Z, **--null-data** separate lines by NUL characters **--help** display this help and exit

--version output version information and exit

If no **-e**, **--expression**, **-f**, or **--file** option is given, then the first non-option argument is taken as the **sed** script to interpret. All remaining arguments are names of input files; if no input files are specified, then the standard input is read.

COMMAND SYNOPSIS

This is just a brief synopsis of **sed** commands to serve as a reminder to those who already know **sed**; other documentation (such as the texinfo document) must be consulted for fuller descriptions.

Zero-address ``commands``

:label Label for **b** and **t** commands.

#comment The comment extends until the next newline (or the end of a **-e** script fragment).

} The closing bracket of a { } block.

Zero- or One- address commands

= Print the current line number.

**a **

text Append **text**, which has each embedded newline preceded by a backslash.

**i **

text Insert **text**, which has each embedded newline preceded by a backslash.

q [exit-code]

Immediately quit the **sed** script without processing any more input, except that if auto-print is not disabled the current pattern space will be printed. The exit code argument is a GNU

extension.

Q [exit-code] Immediately quit the **sed** script without processing any more input. This is a GNU extension.

r filename Append text read from **filename**.

R filename Append a line read from **filename**. Each invocation of the command reads a line from the file. This is a GNU extension.

Commands which accept address ranges

{ Begin a block of commands (end with a **}**).

b label Branch to **label**; if **label** is omitted, branch to end of script.

**c **

text Replace the selected lines with **text**, which has each embedded newline preceded by a backslash.

d Delete pattern space. Start next cycle.

D If pattern space contains no newline, start a normal new cycle as if the **d** command was issued. Otherwise, delete text in the pattern space up to the first newline, and restart cycle with the resultant pattern space, without reading a new line of input.

h H Copy/append pattern space to hold space. **g G** Copy/append hold space to pattern space.

l List out the current line in a ``visually unambiguous`` form.

l width List out the current line in a ``visually unambiguous`` form, breaking it at **width** characters. This is a GNU extension.

n N Read/append the next line of input into the pattern space.

p Print the current pattern space ; **P** Print up to the first embedded newline of the current pattern space.

s/regexp/replacement/

Attempt to match **regexp** against the pattern space. If successful, replace that portion matched with **replacement**. The **replacement** may contain the special character **&** to refer to that

portion of the pattern space which matched, and the special escapes **\1** through **\9** to refer to the corresponding matching sub-expressions in the **regexp**.

t label If a **s///** has done a successful substitution since the last input line was read and since the last **t** or **T** command, then branch to **label**; if **label** is omitted, branch to end of script.

T label If no **s///** has done a successful substitution since the last input line was read and since the last **t** or **T** command, then branch to **label**; if **label** is omitted, branch to end of script.

This is a GNU extension.

w filename Write the current pattern space to **filename**.

W filename Write the first line of the current pattern space to **filename**.

This is a GNU extension.

x Exchange the contents of the hold and pattern spaces.

y/source/dest/

Transliterate the characters in the pattern space which appear in **source** to the corresponding character in **dest**.

Addresses

Sed commands can be given with no addresses, in which case the command will be executed for all input lines; with one address, in which case the command will only be executed for input lines

which match that address; or with two addresses, in which case the command will be executed for all input lines which match the inclusive range of lines starting from the first address and continuing to the second address. Three things to note about address ranges: the syntax is **addr1,addr2** (i.e., the addresses are separated by a comma); the line which **addr1** matched will always be accepted, even if **addr2** selects an earlier line; and if **addr2** is a **regexp**, it will not be tested against the line that **addr1** matched.

After the address (or address-range), and before the command, a **!** may be inserted, which specifies that the command shall only be executed if the address (or address-range) does **not** match.

The following address types are supported:

number Match only the specified line **number** (which increments cumulatively across files, unless the **-s** option is specified on the command line).

first~step

Match every **step**'th line starting with line **first**. For example, ``sed -n 1~2p`` will print all the odd-numbered lines in the input stream, and the address 2~5 will match every fifth

line, starting with the second. **first** can be zero; in this case, **sed** operates as if it were equal to **step**. (This is an extension.)

\$ Match the last line.

/regexp/ Match lines matching the regular expression **regexp**. Matching is performed on the current pattern space, which can be modified with commands such as ``s///``.

\cregexpc Match lines matching the regular expression **regexp**. The **c** may be any character.

GNU **sed** also supports some special 2-address forms:

0,addr2 Start out in "matched first address" state, until **addr2** is found. This is similar to **1,addr2**, except that if **addr2** matches the very first line of input the **0,addr2** form will be at the end of its range, whereas the **1,addr2** form will still be at the beginning of its range. This works only when **addr2** is a regular expression.

addr1,+N Will match **addr1** and the **N** lines following **addr1**.

addr1,~N Will match **addr1** and the lines following **addr1** until the next line whose input line number is a multiple of **N**.

REGULAR EXPRESSIONS

POSIX.2 BREs **should** be supported, but they aren't completely because of performance problems. The **\n** sequence in a regular expression matches the newline character, and similarly for **\a**, **\t**,

and other sequences. The **-E** option switches to using extended regular expressions instead; the **-E** option has been supported for years by GNU **sed**, and is now included in POSIX.

NAME **which** - shows the full path of (shell) commands.

SYNOPSIS

which [options] [--] programname [...]

DESCRIPTION

Which takes one or more arguments. For each of its arguments it prints to stdout the full path of the executables that would have been executed when this argument had been entered at the

shell prompt. It does this by searching for an executable or script in the directories listed in the environment variable **PATH** using the same algorithm as **bash(1)**.

This man page is generated from the file [which.texinfo](#).

OPTIONS

--all, -a

Print all matching executables in **PATH**, not just the first.

--read-alias, -i

Read aliases from stdin, reporting matching ones on stdout. This is useful in combination with using an alias for which itself. For example

alias which='alias | which -i'.

--skip-alias

Ignore option **--read-alias**, if any. This is useful to explicitly search for normal binaries, while using the **--read-alias** option in an alias or function for which.

--read-functions

Read shell function definitions from stdin, reporting matching ones on stdout. This is useful in combination with using a shell function for which itself. For example:

which() { declare -f | which --read-functions \$@ }

export -f which

--skip-functions

Ignore option **--read-functions**, if any. This is useful to explicitly search for normal binaries, while using the **--read-functions** option in an alias or function for which.

--skip-dot

Skip directories in **PATH** that start with a dot.

--skip-tilde

Skip directories in **PATH** that start with a tilde and executables which reside in the **HOME** directory.

--show-dot

If a directory in **PATH** starts with a dot and a matching executable was found for that path, then print **"/programname"** rather than the full path.

--show-tilde

Output a tilde when a directory matches the **HOME** directory. This option is ignored when which is invoked as root.

--tty-only

Stop processing options on the right if not on tty.

--version, -v, -V

Print version information on standard output then exit successfully.

--help

Print usage information on standard output then exit successfully.

RETURN VALUE

Which returns the number of failed arguments, or -1 when no **programname** was given.

EXAMPLE

The recommended way to use this utility is by adding an alias (C shell) or shell function (Bourne shell) for **which** like the following:

```
[ba]sh:
which ()
{
    (alias; declare -f) | /usr/bin/which --tty-only --read-alias --read-functions --show-tilde --show-dot $@
}
export -f which
[t]csh:
```

```
alias which 'alias | /usr/bin/which --tty-only --read-alias --show-dot --show-tilde'
```

This will print the readable **~/** and **./** when starting which from your prompt, while still printing the full path when used from a script:

```
> which q2
~/bin/q2
> echo `which q2`
/home/carlo/bin/q2
```

BUGS

The **HOME** directory is determined by looking for the **HOME** environment variable, which aborts when this variable doesn't exist. **Which** will consider two equivalent directories to be different when one of them contains a path with a symbolic link.

NAME seq - print a sequence of numbers

SYNOPSIS

```
seq [OPTION]... LAST
seq [OPTION]... FIRST LAST
seq [OPTION]... FIRST INCREMENT LAST
```

DESCRIPTION

Print numbers from FIRST to LAST, in steps of INCREMENT.

Mandatory arguments to long options are mandatory for short options too.

-f, --format=FORMAT
use printf style floating-point FORMAT

-s, --separator=STRING
use STRING to separate numbers
(default: \n)

-w, --equal-width
equalize width by padding with leading zeroes

--help display this help and exit

--version
output version information and exit

If FIRST or INCREMENT is omitted, it defaults to 1. That is, an omitted INCREMENT defaults to 1 even when LAST is smaller than FIRST. The sequence of numbers ends when the sum of the current number and INCREMENT would become greater than LAST. FIRST, INCREMENT, and LAST are interpreted as floating point values. INCREMENT is usually positive if FIRST is smaller than LAST, and INCREMENT is usually negative if FIRST is greater than LAST. INCREMENT must not be 0; none of FIRST, INCREMENT and LAST may be NaN. FORMAT must be suitable for printing one argument of type 'double'; it defaults to %.PRECf if FIRST, INCREMENT, and LAST are all fixed point decimal numbers with maximum precision PREC, and to %g otherwise.

AUTHOR

Written by Ulrich Drepper.

REPORTING BUGS

GNU coreutils online help:
<<https://www.gnu.org/software/coreutils/>>
Report any translation bugs to
<<https://translationproject.org/team/>>

NAME: cp - copy files and directories

SYNOPSIS

```
cp [OPTION]... [-T] SOURCE DEST
cp [OPTION]... SOURCE... DIRECTORY
cp [OPTION]... -t DIRECTORY SOURCE...
```

DESCRIPTION

Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY.

Mandatory arguments to long options are mandatory for short options too.

-a, --archive - same as **-dR --preserve=all**

--attributes-only - don't copy the file data, just the attributes

--backup[=CONTROL] - make a backup of each existing destination file

-b like **--backup** but does not accept an argument

--copy-contents - copy contents of special files when recursive

-d same as **--no-dereference --preserve=links**

-f, --force - if an existing destination file cannot be opened, remove it and try again (this option is ignored when the **-n** option is also used)

-i, --interactive - prompt before overwrite (overrides a previous **-n** option)

-H follow command-line symbolic links in SOURCE

-l, --link - hard link files instead of copying

-L, --dereference - always follow symbolic links in SOURCE

-n, --no-clobber - do not overwrite an existing file (overrides a previous **-i** option)

-P, --no-dereference - never follow symbolic links in SOURCE

-p same as **--preserve=mode,ownership,timestamps**

--preserve[=ATTR_LIST] - preserve the specified attributes (default: mode,ownership,timestamps), if possible additional attributes: context, links, xattr, all

--no-preserve=ATTR_LIST - don't preserve the specified attributes

--parents - use full source file name under DIRECTORY

-R, -r, --recursive - copy directories recursively

--reflink[=WHEN] - control clone/CoW copies. See below

--remove-destination - remove each existing destination file before attempting to open it (contrast with **--force**)

--sparse=WHEN - control creation of sparse files. See below

--strip-trailing-slashes - remove any trailing slashes from each SOURCE argument

-s, --symbolic-link - make symbolic links instead of copying

-S, --suffix=SUFFIX - override the usual backup suffix

-t, --target-directory=DIRECTORY - copy all SOURCE arguments into DIRECTORY

-T, --no-target-directory - treat DEST as a normal file

-u, --update - copy only when the SOURCE file is newer than the destination file or when the destination file is missing

-v, --verbose - explain what is being done

-x, --one-file-system - stay on this file system

-Z set SELinux security context of destination file to default type

--context[=CTX]

like **-Z**, or if CTX is specified then set the SELinux or SMACK security context to CTX

--help display this help and exit

--version - output version information and exit

By default, sparse SOURCE files are detected by a crude heuristic and the corresponding DEST file is made sparse as well. That is the behavior selected by **--sparse=auto**. Specify

--sparse=always to create a sparse DEST file whenever the SOURCE file contains a long enough sequence of zero bytes. Use **--sparse=never** to inhibit creation of sparse files.

When **--reflink[=always]** is specified, perform a lightweight copy, where the data blocks are copied only when modified. If this is not possible the copy fails, or if **--reflink=auto** is specified, fall back to a standard copy. Use **--reflink=never** to ensure a standard copy is performed.

The backup suffix is '~', unless set with **--suffix** or SIMPLE_BACKUP_SUFFIX. The version control method may be selected via the **--backup** option or through the VERSION_CONTROL environment variable. Here are the values:

none, off: never make backups (even if **--backup** is given)

numbered, t: make numbered backups

existing, nil: numbered if numbered backups exist, simple otherwise

simple, never: always make simple backups

As a special case, cp makes a backup of SOURCE when the force and backup options are given and SOURCE and DEST are the same name for an existing, regular file.

NAME ls - list directory contents

SYNOPSIS

ls [OPTION]... [FILE]...

DESCRIPTION

List information about the FILEs (the current directory by default). Sort entries alphabetically if none of **-cftuvSUX** nor **--sort** is specified.

Mandatory arguments to long options are mandatory for short options too.

- a, --all** - do not ignore entries starting with **.**
 - A, --almost-all** - do not list implied **.** and **..**
 - author** - with **-l**, print the author of each file
 - b, --escape** - print C-style escapes for nongraphic characters
 - block-size=SIZE** - with **-l**, scale sizes by SIZE when printing them; e.g., **--block-size=M**; see SIZE format below
 - B, --ignore-backups** - do not list implied entries ending with **~**
 - C** with **-lt**: sort by, and show, ctime (time of last modification of file status information); with **-l**: show ctime and sort by name; otherwise: sort by ctime, newest first
 - C** list entries by columns
 - color[=WHEN]** - color the output WHEN; more info below
 - d, --directory** - list directories themselves, not their contents
 - D, --dired** - generate output designed for Emacs' dired mode
 - f** - list all entries in directory order
 - F, --classify[=WHEN]** - append indicator (one of ***/=>@|**) to entries
- WHEN
- file-type** - likewise, except do not append ******
 - format=WORD** - across **-x**, commas **-m**, horizontal **-x**, long **-l**, single-column **-1**, verbose **-l**, vertical **-C**
 - full-time** - like **-l** **--time-style=full-iso**
 - g** like **-l**, but do not list owner
 - group-directories-first** - group directories before files; can be augmented with a **--sort** option, but any use of **--sort=none** (**-U**) disables grouping
 - G, --no-group** - in a long listing, don't print group names
 - h, --human-readable** - with **-l** and **-s**, print sizes like 1K 234M 2G etc.
 - si** likewise, but use powers of 1000 not 1024
 - H, --dereference-command-line** - follow symbolic links listed on the command line
 - dereference-command-line-symlink-to-dir** - follow each command line symbolic link that points to a directory
 - hide=PATTERN** - do not list implied entries matching shell PATTERN (overridden by **-a** or **-A**)
 - hyperlink[=WHEN]** - hyperlink file names WHEN
 - indicator-style=WORD** - append indicator with style WORD to entry names: none (default), slash (**-p**), file-type (**--file-type**), classify (**-F**)
 - i, --inode** - print the index number of each file
 - l, --ignore=PATTERN** - do not list implied entries matching shell PATTERN
 - k, --kibibytes** - default to 1024-byte blocks for file system usage; used only with **-s** and per directory totals
 - l** use a long listing format
 - L, --dereference** - when showing file information for a symbolic link, show information for the file the link references rather than for the link itself

- m** fill width with a comma separated list of entries
- n, --numeric-uid-gid** - like **-l**, but list numeric user and group IDs
- N, --literal** - print entry names without quoting
- o** like **-l**, but do not list group information
- p, --indicator-style=slash** - append **/** indicator to directories
- q, --hide-control-chars** - print **?** instead of nongraphic characters
- show-control-chars** - show nongraphic characters as-is (the default, unless program is 'ls' and output is a terminal)
- Q, --quote-name** - enclose entry names in double quotes
- quoting-style=WORD** - use quoting style WORD for entry names: literal, locale, shell, shell-always, shell-escape, shell-escape-always, c, escape (overrides QUOTING_STYLE environment variable)
- r, --reverse** - reverse order while sorting
- R, --recursive** - list subdirectories recursively
- s, --size** - print the allocated size of each file, in blocks
- S** sort by file size, largest first
- sort=WORD** - sort by WORD instead of name: none (**-U**), size (**-S**), time (**-t**), version (**-v**), extension (**-X**), width
- time=WORD** 0 change the default of using modification times; access time (**-u**): atime, access, use; change time (**-c**): ctime, status; birth time: birth, creation;
- with **-l**, WORD determines which time to show; with **--sort=time**, sort by WORD (newest first)
- time-style=TIME_STYLE** - time/date format with **-l**; see TIME_STYLE below
- t** sort by time, newest first; see **--time**
- T, --tabsize=COLS** - assume tab stops at each COLS instead of 8
- u** with **-lt**: sort by, and show, access time; with **-l**: show access time and sort by name; otherwise: sort by access time, newest first
- U** do not sort; list entries in directory order
- v** natural sort of (version) numbers within text
- w, --width=COLS** set output width to COLS. 0 means no limit
- x** list entries by lines instead of by columns
- X** sort alphabetically by entry extension
- Z, --context** print any security context of each file
- zero** end each output line with NUL, not newline
- 1** list one file per line
- help** display this help and exit
- version** output version information and exit

The SIZE argument is an integer and optional unit (example: 10K is 10*1024). Units are K,M,G,T,P,E,Z,Y (powers of 1024) or KB,MB,... (powers of 1000). Binary prefixes can be used, too: KiB=K, MiB=M, and so on.

The TIME_STYLE argument can be full-iso, long-iso, iso, locale, or +FORMAT. FORMAT is interpreted like in **date(1)**. If FORMAT is FORMAT1<newline>FORMAT2, then FORMAT1 applies to non-recent files and FORMAT2 to recent files. TIME_STYLE prefixed with 'posix-' takes effect only outside the POSIX locale. Also the TIME_STYLE environment variable sets the default style to use.

The WHEN argument defaults to 'always' and can also be 'auto' or 'never'.

Using **color** to distinguish file types is disabled both by default and with **--color=never**. With **--color=auto**, ls emits color codes only when standard output is connected to a terminal. The LS_COLORS environment variable can change the settings. Use the **dircolors(1)** command to set it.

Exit status:

- 0 if OK,
- 1 if minor problems (e.g., cannot access subdirectory),
- 2 if serious trouble (e.g., cannot access command-line argument).

NAME mv - move (rename) files

SYNOPSIS

```
mv [OPTION]... [-T] SOURCE DEST
mv [OPTION]... SOURCE... DIRECTORY
mv [OPTION]... -t DIRECTORY SOURCE...
```

DESCRIPTION

Rename SOURCE to DEST, or move SOURCE(s) to DIRECTORY.

Mandatory arguments to long options are mandatory for short options too.

--backup[=CONTROL] make a backup of each existing destination file

-b like **--backup** but does not accept an argument

-f, --force - do not prompt before overwriting

-i, --interactive - prompt before overwrite

-n, --no-clobber - do not overwrite an existing file

If you specify more than one of **-i**, **-f**, **-n**, only the final one takes effect.

--strip-trailing-slashes - remove any trailing slashes from each SOURCE argument

-S, --suffix=SUFFIX - override the usual backup suffix

-t, --target-directory=DIRECTORY - move all SOURCE arguments into DIRECTORY

-T, --no-target-directory - treat DEST as a normal file

-u, --update - move only when the SOURCE file is newer than the destination file or when the destination file is missing

-v, --verbose - explain what is being done

-Z, --context - set SELinux security context of destination file to default type

--help display this help and exit

--version - output version information and exit

The backup suffix is '~', unless set with **--suffix** or **SIMPLE_BACKUP_SUFFIX**. The version control method may be selected via the **--backup** option or through the **VERSION_CONTROL** environment variable. Here are the values:

- none, off: never make backups (even if **--backup** is given)
- numbered, t: make numbered backups
- existing, nil: numbered if numbered backups exist, simple otherwise

simple, never: always make simple backups

NAME

shred - overwrite a file to hide its contents, and optionally delete it

SYNOPSIS

```
shred [OPTION]... FILE...
```

DESCRIPTION

Overwrite the specified FILE(s) repeatedly, in order to make it harder for even very expensive hardware probing to recover the data.

If FILE is -, shred standard output.

Mandatory arguments to long options are mandatory for short options too.

-f, --force
change permissions to allow writing if necessary

-n, --iterations=N
overwrite N times instead of the default (3)

--random-source=FILE
get random bytes from FILE

-s, --size=N
shred this many bytes (suffixes like K, M, G accepted)

-u deallocate and remove file after overwriting

--remove[=HOW]
like **-u** but give control on HOW to delete; See below

-v, --verbose
show progress

-x, --exact
do not round file sizes up to the next full block;

this is the default for non-regular files

-z, --zero
add a final overwrite with zeros to hide shredding

--help display this help and exit

--version
output version information and exit

Delete FILE(s) if **--remove (-u)** is specified. The default is not to remove the files because it is common to operate on device files like /dev/hda, and those files usually should not be removed. The optional HOW parameter indicates how to remove a directory entry: 'unlink' => use a standard unlink call. 'wipe' => also first obfuscate bytes in the name. 'wipesync' => also sync each obfuscated byte to the device. The default mode is 'wipesync', but note it can be expensive.

CAUTION: shred assumes the file system and hardware overwrite data in place. Although this is common, many platforms operate otherwise. Also, backups and mirrors may contain unremovable copies that will let a shredded file be recovered later. See the GNU coreutils manual for details.

NAME **rm** - remove files or directories

SYNOPSIS

rm [OPTION]... [FILE]...

DESCRIPTION

This manual page documents the GNU version of **rm**. **rm** removes each specified file. By default, it does not remove directories.

If the **-I** or **--interactive=once** option is given, and there are more than three files or the **-r**, **-R**, or **--recursive** are given, then **rm** prompts the user for whether to proceed with the entire

operation. If the response is not affirmative, the entire command is aborted.

Otherwise, if a file is unwritable, standard input is a terminal, and the **-f** or **--force** option is not given, or the **-i** or **--interactive=always** option is given, **rm** prompts the user for whether to remove the file. If the response is not affirmative, the file is skipped.

OPTIONS

Remove (unlink) the FILE(s).

-f, --force - ignore nonexistent files and arguments, never prompt

-i prompt before every removal

-I prompt once before removing more than three files, or when removing recursively; less intrusive than **-i**, while still giving protection against most mistakes

--interactive[=WHEN] - prompt according to WHEN: never, once (**-I**), or always (**-i**); without WHEN, prompt always

--one-file-system - when removing a hierarchy recursively, skip any directory that is on a file system different from that of the corresponding command line argument

--no-preserve-root - do not treat '/' specially

--preserve-root[=all] - do not remove '/' (default); with 'all', reject any command line argument on a separate device from its parent

-r, -R, --recursive - remove directories and their contents recursively

-d, --dir - remove empty directories

-v, --verbose - explain what is being done

--help display this help and exit

--version - output version information and exit

By default, **rm** does not remove directories. Use the **--recursive** (**-r** or **-R**) option to remove each listed directory, too, along with all of its contents.

To remove a file whose name starts with a '-', for example '-foo', use one of these commands:

```
rm -- -foo
```

```
rm ./-foo
```

Note that if you use **rm** to remove a file, it might be possible to recover some of its contents, given sufficient expertise and/or time. For greater assurance that the contents are truly unrecoverable, consider using **shred**(1).

NAME **cat** - concatenate files and print on the standard output

SYNOPSIS

cat [OPTION]... [FILE]...

DESCRIPTION

Concatenate FILE(s) to standard output.

With no FILE, or when FILE is -, read standard input.

-A, --show-all
equivalent to **-vET**

-b, --number-nonblank
number nonempty output lines, overrides **-n**

-e equivalent to **-vE**

-E, --show-ends
display \$ at end of each line

-n, --number
number all output lines

-s, --squeeze-blank
suppress repeated empty output lines

-t equivalent to **-vT**

-T, --show-tabs
display TAB characters as ^I

-u (ignored)

-v, --show-nonprinting
use ^ and M- notation, except for LFD and TAB

--help display this help and exit

--version
output version information and exit

EXAMPLES

```
cat f - g
```

Output f's contents, then standard input, then g's contents.

```
cat
```

Copy standard input to standard output.

NAME chmod - change file mode bits

SYNOPSIS

```
chmod [OPTION]... MODE[MODE]... FILE...  
chmod [OPTION]... OCTAL-MODE FILE...  
chmod [OPTION]... --reference=RFILE FILE...
```

DESCRIPTION

This manual page documents the GNU version of **chmod**. **chmod** changes the file mode bits of each given file according to mode, which can be either a symbolic representation of changes to make, or an octal number representing the bit pattern for the new mode bits.

The format of a symbolic mode is [**ugoa**...][[-+=][**perms**...][...], where **perms** is either zero or more letters from the set **rwXst**, or a single letter from the set **ugo**. Multiple symbolic modes can be given, separated by commas.

A combination of the letters **ugoa** controls which users' access to the file will be changed: (**u**): the user who owns it, (**g**): other users in the file's group, (**o**): other users not in the file's group, or (**a**): all users. If none of these are given, the effect is as if (**a**) were given, but bits that are set in the umask are not affected.

The operator **+** causes the selected file mode bits to be added to the existing file mode bits of each file; **-** causes them to be removed; and **=** causes them to be added and causes unmentioned bits to be removed except that a directory's unmentioned set user and group ID bits are not affected.

The letters **rwXst** select file mode bits for the affected users: read (**r**), write (**w**), execute (or search for directories) (**x**), execute/search only if the file is a directory or already has execute permission for some user (**X**), set user or group ID on execution (**s**), restricted deletion flag or sticky bit (**t**). Instead of one or more of these letters, you can specify exactly one of the letters **ugo**: the permissions granted to the user who owns the file (**u**), the permissions granted to other users who are members of the file's group (**g**), and the permissions granted to users that are in neither of the two preceding categories (**o**).

A numeric mode is from one to four octal digits (0-7), derived by adding up the bits with values 4, 2, and 1. Omitted digits are assumed to be leading zeros. The first digit selects the set user ID (4) and set group ID (2) and restricted deletion or sticky (1) attributes. The second digit selects permissions for the user who owns the file: read (4), write (2), and execute (1); the third selects permissions for other users in the file's group, with the same values; and the fourth for other users not in the file's group, with the same values.

chmod never changes the permissions of symbolic links; the **chmod** system call cannot change their permissions. This is not a problem since the permissions of symbolic links are never used. However, for each symbolic link listed on the command line, **chmod** changes the permissions of the pointed-to file. In contrast, **chmod** ignores symbolic links encountered during recursive directory traversals.

SETUID AND SETGID BITS

chmod clears the set-group-ID bit of a regular file if the file's group ID does not match the user's effective group ID or one of the user's supplementary group IDs, unless the user has appropriate privileges. Additional restrictions may cause the set-user-ID and set-group-ID bits of MODE or RFILE to be ignored. This behavior depends on the policy and functionality of the underlying **chmod** system call. When in doubt, check the underlying system behavior.

For directories **chmod** preserves set-user-ID and set-group-ID bits unless you explicitly specify otherwise. You can set or clear the bits with symbolic modes like **u+s** and **g-s**. To clear these bits for directories with a numeric mode requires an additional leading zero like **00755**, leading minus like **-6000**, or leading equals like **=755**.

RESTRICTED DELETION FLAG OR STICKY BIT

The restricted deletion flag or sticky bit is a single bit, whose interpretation depends on the file type. For directories, it prevents unprivileged users from removing or renaming a file in the directory unless they own the file or the directory; this is called the restricted deletion flag for the directory, and is commonly found on world-writable directories like **/tmp**. For regular files on some older systems, the bit saves the program's text image on the swap device so it will load more quickly when run; this is called the sticky bit.

OPTIONS

Change the mode of each FILE to MODE. With **--reference**, change the mode of each FILE to that of RFILE.

-c, **--changes** - like verbose but report only when a change is made
-f, **--silent**, **--quiet** - suppress most error messages
-v, **--verbose** - output a diagnostic for every file processed
--no-preserve-root - do not treat '/' specially (the default)
--preserve-root - fail to operate recursively on '/'
--reference=RFILE - use RFILE's mode instead of MODE values
-R, **--recursive** - change files and directories recursively
--help display this help and exit
--version - output version information and exit

Each MODE is of the form '[**ugoa**]*([-+]=)[(**rwXst**)*][**ugo**]]+([-+])[0-7]+'.

NAME find - search for files in a directory hierarchy (10 pages)

SYNOPSIS **find** [-H] [-L] [-P] [-D debugopts] [-Olevel] [starting-point...] [expression]

DESCRIPTION This manual page documents the GNU version of **find**. GNU **find** searches the directory tree rooted at each given starting-point by evaluating the given expression from left to right, according to the rules of precedence (see section OPERATORS), until the outcome is known (the left hand side is false for and operations, true for or), at which point **find** moves on to the next file name. If no starting-point is specified, '.' is assumed.

If you are using **find** in an environment where security is important (for example if you are using it to search directories that are writable by other users), you should read the "Security Considerations" chapter of the findutils documentation, which is called **Finding Files** and comes with findutils. That document also includes a lot more detail and discussion than this manual page, so you may find it a more useful source of information.

OPTIONS

The **-H**, **-L** and **-P** options control the treatment of symbolic links. Command-line arguments following these are taken to be names of files or directories to be examined, up to the first argument that begins with '-', or the argument '(' or '!'. That argument and any following arguments are taken to be the expression describing what is to be searched for. If no paths are given, the current directory is used. If no expression is given, the expression **-print** is used (but you should probably consider using **-print0** instead, anyway).

This manual page talks about 'options' within the expression list. These options control the behaviour of **find** but are specified immediately after the last path name. The five 'real' options **-H**, **-L**, **-P**, **-D** and **-O** must appear before the first path name, if at all. A double dash -- can also be used to signal that any remaining arguments are not options (though ensuring that all start points begin with either './' or '/' is generally safer if you use wildcards in the list of start points).

-P Never follow symbolic links. This is the default behaviour. When **find** examines or prints information a file, and the file is a symbolic link, the information used shall be taken from the properties of the symbolic link itself.

-L Follow symbolic links. When **find** examines or prints information about files, the information used shall be taken from the properties of the file to which the link points, not from the link itself (unless it is a broken symbolic link or **find** is unable to examine the file to which the link points). Use of this option implies **-noleaf**. If you later use the **-P** option, **-noleaf** will still be in effect. If **-L** is in effect and **find** discovers a symbolic link to a subdirectory during its search, the subdirectory pointed to by the symbolic link will be searched.

When the **-L** option is in effect, the **-type** predicate will always match against the type of the file that a symbolic link points to rather than the link itself (unless the symbolic link is broken). Actions that can cause symbolic links to become broken while **find** is executing (for example **-delete**) can give rise to confusing behaviour. Using **-L** causes the **-lname** and **-ilname** predicates always to return false.

-H Do not follow symbolic links, except while processing the command line arguments. When **find** examines or prints information about files, the information used shall be taken from the properties of the symbolic link itself. The only exception to this behaviour is when a file specified on the command line is a symbolic link, and the link can be resolved. For that situation, the information used is taken from whatever the link points to (that is, the link is followed). The information about the link itself is used as a fallback if the file pointed to by the symbolic link cannot be examined. If **-H** is in effect and one of the paths specified on the command line is a symbolic link to a directory, the contents of that directory will be examined (though of course **-maxdepth 0** would prevent this).

If more than one of **-H**, **-L** and **-P** is specified, each overrides the others; the last one appearing on the command line takes effect. Since it is the default, the **-P** option should be considered to be in effect unless either **-H** or **-L** is specified.

GNU **find** frequently stats files during the processing of the command line itself, before any searching has begun. These options also affect how those arguments are processed. Specifically, there are a number of tests that compare files listed on the command line against a file we are currently considering. In each case, the file specified on the command line will have been examined and some of its properties will have been saved. If the named file is in fact a symbolic link, and the **-P** option is in effect (or if neither **-H** nor **-L** were specified), the information used for the comparison will be taken from the properties of the symbolic link. Otherwise, it will be taken from the properties of the file the link points to. If **find** cannot follow the link (for example because it has insufficient privileges or the link points to a nonexistent file) the properties of the link itself will be used.

When the **-H** or **-L** options are in effect, any symbolic links listed as the argument of **-newer** will be dereferenced, and the timestamp will be taken from the file to which the symbolic link points. The same consideration applies to **-newerXY**, **-anewer** and **-cnewer**.

The **-follow** option has a similar effect to **-L**, though it takes effect at the point where it appears (that is, if **-L** is not used but **-follow** is, any symbolic links appearing after **-follow** on the command line will be dereferenced, and those before it will not).

-D debugoptions - Print diagnostic information; this can be helpful to diagnose problems with why **find** is not doing what you want. The list of debug options should be comma separated. Compatibility of the debug options is not guaranteed between releases of findutils. For a complete list of valid debug options, see the output of **find -D help**. Valid debug options include

help Explain the debugging options

tree Show the expression tree in its original and optimised form.

stat Print messages as files are examined with the **stat** and **lstat** system calls. The **find** program tries to minimise such calls.

opt Prints diagnostic information relating to the optimisation of the expression tree; see the **-O** option.

rates Prints a summary indicating how often each predicate succeeded or failed.

-Olevel Enables query optimisation. The **find** program reorders tests to speed up execution while preserving the overall effect; that is, predicates with side effects are not reordered relative to each other. The optimisations performed at each optimisation level are as follows.

- 0 Equivalent to optimisation level 1.
- 1 This is the default optimisation level and corresponds to the traditional behaviour. Expressions are reordered so that tests based only on the names of files (for example **-name** and **-regex**) are performed first.
- 2 Any **-type** or **-xtype** tests are performed after any tests based only on the names of files, but before any tests that require information from the inode. On many modern versions of Unix, file types are returned by **readdir()** and so these predicates are faster to evaluate than predicates which need to stat the file first. If you use the **-fstype FOO** predicate and specify a filesystem type **FOO** which is not known (that is, present in `/etc/mstab`) at the time **find** starts, that predicate is equivalent to **-false**.
- 3 At this optimisation level, the full cost-based query optimiser is enabled. The order of tests is modified so that cheap (i.e. fast) tests are performed first and more expensive ones are performed later, if necessary. Within each cost band, predicates are evaluated earlier or later according to whether they are likely to succeed or not. For **-o**, predicates which are likely to succeed are evaluated earlier, and for **-a**, predicates which are likely to fail are evaluated earlier.

The cost-based optimiser has a fixed idea of how likely any given test is to succeed. In some cases the probability takes account of the specific nature of the test (for example, **-type f** is assumed to be more likely to succeed than **-type c**). The cost-based optimiser is currently being evaluated. If it does not actually improve the performance of **find**, it will be removed again. Conversely, optimisations that prove to be reliable, robust and effective may be enabled at lower optimisation levels over time. However, the default behaviour (i.e. optimisation level 1) will not be changed in the 4.3.x release series. The **findutils** test suite runs all the tests on **find** at each optimisation level and ensures that the result is the same.

EXPRESSION

The part of the command line after the list of starting points is the expression. This is a kind of query specification describing how we match files and what we do with the files that were matched. An expression is composed of a sequence of things:

Tests - Tests return a true or false value, usually on the basis of some property of a file we are considering. The **-empty** test for example is true only when the current file is empty.

Actions - Actions have side effects (such as printing something on the standard output) and return either true or false, usually based on whether or not they are successful. The **-print** action for example prints the name of the current file on the standard output.

Global options - Global options affect the operation of tests and actions specified on any part of the command line. Global options always return true. The **-depth** option for example makes **find** traverse the file system in a depth-first order.

Positional options - Positional options affect only tests or actions which follow them. Positional options always return true. The **-regextype** option for example is positional, specifying the regular expression dialect for regular expressions occurring later on the command line.

Operators - Operators join together the other items within the expression. They include for example **-o** (meaning logical OR) and **-a** (meaning logical AND). Where an operator is missing, **-a** is assumed.

If the whole expression contains no actions other than **-prune** or **-print**, **-print** is performed on all files for which the whole expression is true. The **-delete** action also acts like an option (since it implies **-depth**).

POSITIONAL OPTIONS

Positional options always return true. They affect only tests occurring later on the command line.

-daystart - Measure times (for **-amin**, **-atime**, **-cmin**, **-ctime**, **-mmin**, and **-mtime**) from the beginning of today rather than from 24 hours ago. This option only affects tests which appear later on the command line.

-follow - Deprecated; use the **-L** option instead. Dereference symbolic links. Implies **-noleaf**. The **-follow** option affects only those tests which appear after it on the command line. Unless the **-H** or **-L** option has been specified, the position of the **-follow** option changes the behaviour of the **-newer** predicate; any files listed as the argument of **-newer** will be dereferenced if they are symbolic links. The same consideration applies to **-newerXY**, **-anewer** and **-cnewer**. Similarly, the **-type** predicate will always match against the type of the file that a symbolic link points to rather than the link itself. Using **-follow** causes the **-lname** and **-ilname** predicates always to return false.

-regextype type - Changes the regular expression syntax understood by **-regex** and **-iregex** tests which occur later on the command line. To see which regular expression types are known, use **-regextype help**. The Texinfo documentation (see **SEE ALSO**) explains the meaning of and differences between the various types of regular expression.

-warn, **-nowarn** - Turn warning messages on or off. These warnings apply only to the command line usage, not to any conditions that **find** might encounter when it searches directories. The default behaviour corresponds to **-warn** if standard input is a tty, and to **-nowarn** otherwise. If a warning message relating to command-line usage is produced, the exit status of **find** is not affected. If the **POSIXLY_CORRECT** environment variable is set, and **-warn** is also used, it is not specified which, if any, warnings will be active.

GLOBAL OPTIONS

Global options always return true. Global options take effect even for tests which occur earlier on the command line. To prevent confusion, global options should be specified on the command-line after the list of start points, just before the first test, positional option or action. If you specify a global option in some other place, **find** will issue a warning message explaining that this can be confusing.

The global options occur after the list of start points, and so are not the same kind of option as **-L**, for example.

-d A synonym for **-depth**, for compatibility with FreeBSD, NetBSD, MacOS X and OpenBSD.

-depth Process each directory's contents before the directory itself. The **-delete** action also implies **-depth**.

-help, --help Print a summary of the command-line usage of **find** and exit.

-ignore_readdir_race - Normally, **find** will emit an error message when it fails to stat a file. If you give this option and a file is deleted between the time **find** reads the name of the file from the directory and the time it tries to stat the file, no error message will be issued. This also applies to files or directories whose names are given on the command line. This option takes effect at the time the command line is read, which means that you cannot search one part of the filesystem with this option on and part of it with this option off (if you need to do that, you will need to issue two **find** commands instead, one with the option and one without it).

-maxdepth levels - Descend at most levels (a non-negative integer) levels of directories below the starting-points. **-maxdepth 0** means only apply the tests and actions to the starting-points themselves.

-mindepth levels - Do not apply any tests or actions at levels less than levels (a non-negative integer). **-mindepth 1** means process all files except the starting-points.

-mount - Don't descend directories on other filesystems. An alternate name for **-xdev**, for compatibility with some other versions of **find**.

-noignore_readdir_race - Turns off the effect of **-ignore_readdir_race**.

-noleaf - Do not optimize by assuming that directories contain 2 fewer subdirectories than their hard link count. This option is needed when searching filesystems that do not follow the Unix directory-link convention, such as CD-ROM or MS-DOS filesystems or AFS volume mount points. Each directory on a normal Unix filesystem has at least 2 hard links: its name and its ``.`` entry. Additionally, its subdirectories (if any) each have a ``.`` entry linked to that directory. When **find** is examining a directory, after it has stat'd 2 fewer subdirectories than the directory's link count, it knows that the rest of the entries in the directory are non-directories ('leaf' files in the directory tree). If only the files' names need to be examined, there is no need to stat them; this gives a significant increase in search speed.

-version, --version - Print the **find** version number and exit.

-xautofs - Don't descend directories on autofs filesystems.

-xdev - Don't descend directories on other filesystems.

TESTS

Some tests, for example **-newerXY** and **-samefile**, allow comparison between the file currently being examined and some reference file specified on the command line. When these tests are used, the interpretation of the reference file is determined by the options **-H**, **-L** and **-P** and any previous **-follow**, but the reference file is only examined once, at the time the command line is parsed. If the reference file cannot be examined (for example, the **stat(2)** system call fails for it), an error message is issued, and **find** exits with a nonzero status.

Numeric arguments can be specified as

+n for greater than n.

-n for less than n.

n for exactly n.

-amin n File was last accessed n minutes ago.

-anewer file File was last accessed more recently than file was modified. If file is a symbolic link and the **-H** option or the **-L** option is in effect, the access time of the file it points to is always used.

-atime n File was last accessed n*24 hours ago. When **find** figures out how many 24-hour periods ago the file was last accessed, any fractional part is ignored, so to match **-atime +1**, a file has to have been accessed at least two days ago.

-cmin n File's status was last changed n minutes ago.

-cnewer file File's status was last changed more recently than file was modified. If file is a symbolic link and the **-H** option or the **-L** option is in effect, the status-change time of the file it points to is always used.

-ctime n File's status was last changed n*24 hours ago. See the comments for **-atime** to understand how rounding affects the interpretation of file status change times.

-empty File is empty and is either a regular file or a directory.

-executable Matches files which are executable and directories which are searchable (in a file name resolution sense). This takes into account access control lists and other permissions artefacts which the **-perm** test ignores. This test makes use of the **access(2)** system call, and so can be fooled by NFS servers which do UID mapping (or root-squashing), since many systems implement **access(2)** in the client's kernel and so cannot make use of the UID mapping information held on the server. Because this test is based only on the result of the **access(2)** system call, there is no guarantee that a file for which this test succeeds can actually be executed.

-false Always false.

-fstype type

File is on a filesystem of type type. The valid filesystem types vary among different versions of Unix; an incomplete list of filesystem types that are accepted on some version of Unix or another is: ufs, 4.2, 4.3, nfs, tmp, mfs, S51K, S52K. You can use **-printf** with the %F directive to see the types of your filesystems.

-gid n File's numeric group ID is n.

-group gname File belongs to group gname (numeric group ID allowed).

-lname pattern Like **-lname**, but the match is case insensitive. If the **-L** option or the **-follow** option is in effect, this test returns false unless the symbolic link is broken.

-iname pattern Like **-name**, but the match is case insensitive. For example, the patterns 'fo*' and 'F??' match the file names 'Foo', 'FOO', 'foo', 'fOo', etc. The pattern '*foo*' will also match a file called 'foobar'.

-inum n File has inode number n. It is normally easier to use the **-samefile** test instead.

-ipath pattern Like **-path**, but the match is case insensitive.

-iregex pattern Like **-regex**, but the match is case insensitive.

-iwholename pattern See **-ipath**. This alternative is less portable than **-ipath**.

-links n File has n links.

-lname pattern - File is a symbolic link whose contents match shell pattern pattern. The metacharacters do not treat '/' or '.' specially. If the **-L** option or the **-follow** option is in effect, this test returns false unless the symbolic link is broken.

-mmin n - File's data was last modified n minutes ago.

-mtime n - File's data was last modified n*24 hours ago. See the comments for **-atime** to understand how rounding affects the interpretation of file modification times.

-name pattern - Base of file name (the path with the leading directories removed) matches shell pattern pattern. Because the leading directories are removed, the file names considered for a match with **-name** will never include a slash, so '-name a/b' will never match anything (you probably need to use **-path** instead). A warning is issued if you try to do this, unless the environment variable POSIXLY_CORRECT is set. The metacharacters ('*', '?', and '[') match a '.' at the start of the base name (this is a change in findutils-4.2.2; see section STANDARDS CONFORMANCE below). To ignore a directory and the files under it, use **-prune**; see an example in the description of **-path**. Braces are not recognised as being special, despite the fact

that some shells including Bash imbue braces with a special meaning in shell patterns. The filename matching is performed with the use of the **fnmatch(3)** library function. Don't forget to enclose the pattern in quotes in order to protect it from expansion by the shell.

-newer file - File was modified more recently than file. If file is a symbolic link and the **-H** option or the **-L** option is in effect, the modification time of the file it points to is always used.

-newerXY reference Succeeds if timestamp X of the file being considered is newer than timestamp Y of the file reference. The letters X and Y can be any of the following letters:

a The access time of the file reference

B The birth time of the file reference

c The inode status change time of reference

m The modification time of the file reference

t reference is interpreted directly as a time

Some combinations are invalid; for example, it is invalid for X to be t. Some combinations are not implemented on all systems; for example B is not supported on all systems. If an invalid or unsupported combination of XY is specified, a fatal error results. Time specifications are interpreted as for the argument to the **-d** option of GNU **date**. If you try to use the birth time of a reference file, and the birth time cannot be determined, a fatal error message results. If you specify a test which refers to the birth time of files being examined, this test will fail for any files where the birth time is unknown.

-nogroup - No group corresponds to file's numeric group ID.

-nouser - No user corresponds to file's numeric user ID.

-path pattern - File name matches shell pattern pattern. The metacharacters do not treat '/' or '.' specially; so, for example, find . -path "/sr*sc" will print an entry for a directory called './src/misc' (if one exists). To ignore a whole directory tree, use **-prune** rather than checking every file in the tree. For example, to skip the directory 'src/emacs' and all files and directories under it, and print the names of the other files found, do something like this: find . -path ./src/emacs -prune -o -print Note that the pattern match test applies to the whole file name, starting from one of the start points named on the command line. It would only make sense to use an absolute path name here if the relevant start point is also an absolute path. This means that this command will never match anything: find bar -path /foo/bar/myfile -print.

Find compares the **-path** argument with the concatenation of a directory name and the base name of the file it's examining. Since the concatenation will never end with a slash, **-path** arguments ending in a slash will match nothing (except perhaps a start point specified on the command line). The predicate **-path** is also supported by HP-UX **find** and will be in a forthcoming version of the POSIX standard.

-perm mode - File's permission bits are exactly mode (octal or symbolic). Since an exact match is required, if you want to use this form for symbolic modes, you may have to specify a rather complex mode string. For example '-perm g=w' will only match files which have mode 0020 (that is, ones for which group write permission is the only permission set). It is more likely that you will want to use the '/' or '-' forms, for example '-perm -g=w', which matches any file with group write permission. See the **EXAMPLES** section for some illustrative examples.

-perm -mode - All of the permission bits mode are set for the file. Symbolic modes are accepted in this form, and this is usually the way in which you would want to use them. You must specify 'u', 'g' or 'o' if you use a symbolic mode. See the **EXAMPLES** section for some illustrative examples.

-perm /mode - Any of the permission bits mode are set for the file. Symbolic modes are accepted in this form. You must specify 'u', 'g' or 'o' if you use a symbolic mode. See the **EXAMPLES** section for some illustrative examples. If no permission bits in mode are set, this test matches any file (the idea here is to be consistent with the behaviour of **-perm -000**).

-perm +mode - This is no longer supported (and has been deprecated since 2005). Use **-perm /mode** instead.

-readable - Matches files which are readable. This takes into account access control lists and other permissions artefacts which the **-perm** test ignores. This test makes use of the **access(2)** system call, and so can be fooled by NFS servers which do UID mapping (or root-squashing), since many systems implement **access(2)** in the client's kernel and so cannot make use of the UID mapping information held on the server.

-regex pattern - File name matches regular expression **pattern**. This is a match on the whole path, not a search. For example, to match a file named `./fubar3`, you can use the regular expression `.*bar.` or `.*b.*3`, but not `.*f.*r3`. The regular expressions understood by **find** are by default Emacs Regular Expressions, but this can be changed with the **-regextype** option.

-samefile name - File refers to the same inode as **name**. When **-L** is in effect, this can include symbolic links.

-size n[cwbkMG]

File uses **n** units of space, rounding up. The following suffixes can be used:

'b' for 512-byte blocks (this is the default if no suffix is used), **'c'** for bytes, **'w'** for two-byte words, **'k'** for Kilobytes (units of 1024 bytes), **'M'** for Megabytes (units of 1048576 bytes), **'G'** for Gigabytes (units of 1073741824 bytes)

The size does not count indirect blocks, but it does count blocks in sparse files that are not actually allocated. Bear in mind that the `'%k'` and `'%b'` format specifiers of **-printf** handle sparse files differently. The `'b'` suffix always denotes 512-byte blocks and never 1 Kilobyte blocks, which is different to the behaviour of **-ls**. The **+** and **-** prefixes signify greater than and less than, as usual, but bear in mind that the size is rounded up to the next unit (so a 1-byte file is not matched by **-size -1M**).

-true Always true.

-type c

File is of type **c**:

b block (buffered) special, **c** character (unbuffered) special

d directory **p** named pipe (FIFO)

f regular file **l** symbolic link; this is never true if the **-L** option or the **-follow** option is in effect, unless the symbolic link is broken. If you want to search for symbolic links when **-L** is in effect, use **-xtype**.

s socket **D** door (Solaris)

-uid n File's numeric user ID is **n**.

-used n - File was last accessed **n** days after its status was last changed.

-user uname - File is owned by user **uname** (numeric user ID allowed).

-wholename pattern - See **-path**. This alternative is less portable than **-path**.

-writable - Matches files which are writable. This takes into account access control lists and other permissions artefacts which the **-perm** test ignores. This test makes use of the **access(2)** system call, and so can be fooled by NFS servers which do UID mapping (or root-squashing), since many systems implement **access(2)** in the client's kernel and so cannot make use of the UID mapping information held on the server.

-xtype c - The same as **-type** unless the file is a symbolic link. For symbolic links: if the **-H** or **-P** option was specified, true if the file is a link to a file of type **c**; if the **-L** option has been given, true if **c** is `'l'`. In other words, for symbolic links, **-xtype** checks the type of the file that **-type** does not check.

-context pattern - (SELinux only) Security context of the file matches glob **pattern**.

ACTIONS

-delete - Delete files; true if removal succeeded. If the removal failed, an error message is issued. If **-delete** fails, **find**'s exit status will be nonzero (when it eventually exits). Use of

-delete automatically turns on the `'-depth'` option.

Warnings: Don't forget that the **find** command line is evaluated as an expression, so putting **-delete** first will make **find** try to delete everything below the starting points you specified. When testing a **find** command line that you later intend to use with **-delete**, you should explicitly specify **-depth** in order to avoid later surprises. Because **-delete** implies **-depth**, you cannot usefully use **-prune** and **-delete** together.

-exec command ;

Execute **command**; true if 0 status is returned. All following arguments to **find** are taken to be arguments to the command until an argument consisting of `;` is encountered. The string `{}` is replaced by the current file name being processed everywhere it occurs in the arguments to the command, not just in arguments where it is alone, as in some versions of **find**. Both of these constructions might need to be escaped (with a `'\'`) or quoted to protect them from expansion by the shell. See the **EXAMPLES** section for examples of the use of the **-exec** option. The specified command is run once for each matched file. The command is executed in the starting directory. There are unavoidable security problems surrounding use of the **-exec** action; you should use the **-execdir** option instead.

-exec command {} + This variant of the **-exec** action runs the specified command on the selected files, but the command line is built by appending each selected file name at the end; the total number of invocations of the command will be much less than the number of matched files. The command line is built in much the same way that **xargs** builds its command lines. Only one instance of `{}` is allowed within the command. The command is executed in the starting directory. If any invocation returns a non-zero value as exit status, then **find** returns a non-zero exit status. If **find** encounters an error, this can sometimes cause an immediate exit, so some pending commands may not be run at all. This variant of **-exec** always returns true.

-execdir command ;

-execdir command {} + Like **-exec**, but the specified command is run from the subdirectory containing the matched file, which is not normally the directory in which you started **find**. This is a much more secure method for invoking commands, as it avoids race conditions during resolution of the paths to the matched files. As with the **-exec** action, the `+` form of **-execdir** will build a command line to process more than one matched file, but any given invocation of **command** will only list files that exist in the same subdirectory. If you use this option, you must ensure that your **\$PATH** environment variable does not reference `.`; otherwise, an attacker can run any commands they like by leaving an appropriately-named file in a directory in which you will run **-execdir**. The same applies to having entries in **\$PATH** which are empty or which are not absolute directory names. If any invocation returns a non-zero value as exit status, then **find** returns a non-zero exit status. If **find** encounters an error, this can sometimes cause an immediate exit, so some pending commands may not be run at all. The result of the action depends on whether the `+` or the `;` variant is being used; **-execdir command {} +** always returns true, while **-execdir command {} ;** returns true only if **command** returns 0.

-fls file - True; like **-ls** but write to file like **-fprint**. The output file is always created, even if the predicate is never matched. See the **UNUSUAL FILENAMES** section for information about how unusual characters in filenames are handled.

-fprint file True; print the full file name into file file. If file does not exist when **find** is run, it is created; if it does exist, it is truncated. The file names `/dev/stdout` and `/dev/stderr` are handled specially; they refer to the standard output and standard error output, respectively. The output file is always created, even if the predicate is never matched. See the **UNUSUAL FILENAMES** section for information about how unusual characters in filenames are handled.

-fprint0 file
True; like **-print0** but write to file like **-fprint**. The output file is always created, even if the predicate is never matched. See the **UNUSUAL FILENAMES** section for information about how unusual characters in filenames are handled.

-fprintf file format - True; like **-printf** but write to file like **-fprint**. The output file is always created, even if the predicate is never matched. See the **UNUSUAL FILENAMES** section for information about how unusual characters in filenames are handled.

-ls True; list current file in **ls -dils** format on standard output. The block counts are of 1K blocks, unless the environment variable `POSIXLY_CORRECT` is set, in which case 512-byte blocks are used. See the **UNUSUAL FILENAMES** section for information about how unusual characters in filenames are handled.

-ok command ; - Like **-exec** but ask the user first. If the user agrees, run the command. Otherwise just return false. If the command is run, its standard input is redirected from `/dev/null`.

The response to the prompt is matched against a pair of regular expressions to determine if it is an affirmative or negative response. This regular expression is obtained from the system if the `POSIXLY_CORRECT` environment variable is set, or otherwise from **find**'s message translations. If the system has no suitable definition, **find**'s own definition will be used. In either case, the interpretation of the regular expression itself will be affected by the environment variables `LC_CTYPE` (character classes) and `LC_COLLATE` (character ranges and equivalence classes).

-okdir command ; Like **-execdir** but ask the user first in the same way as for **-ok**. If the user does not agree, just return false. If the command is run, its standard input is redirected from `/dev/null`.

-print True; print the full file name on the standard output, followed by a newline. If you are piping the output of **find** into another program and there is the faintest possibility that the files which you are searching for might contain a newline, then you should seriously consider using the **-print0** option instead of **-print**. See the **UNUSUAL FILENAMES** section for information about how unusual characters in filenames are handled.

-print0 - True; print the full file name on the standard output, followed by a null character (instead of the newline character that **-print** uses). This allows file names that contain newlines or other types of white space to be correctly interpreted by programs that process the **find** output. This option corresponds to the **-0** option of **xargs**.

-printf format True; print format on the standard output, interpreting `\` escapes and `%` directives. Field widths and precisions can be specified as with the `printf` C function. Please note that

many of the fields are printed as `%s` rather than `%d`, and this may mean that flags don't work as you might expect. This also means that the `-'` flag does work (it forces fields to be

left-aligned). Unlike **-print**, **-printf** does not add a newline at the end of the string. The escapes and directives are:

`\a` Alarm bell. `\b` Backspace. `\c` Stop printing from this format immediately and flush the output. `\f` Form feed.

`\n` Newline. `\r` Carriage return. `\t` Horizontal tab. `\v` Vertical tab. `\0` ASCII NUL.

`\\` A literal backslash (`\"`). `\NNN` The character whose ASCII code is NNN (octal).

A `\"` character followed by any other character is treated as an ordinary character, so they both are printed.

`%%` A literal percent sign. `%a` File's last access time in the format returned by the C `ctime` function.

`%Ak` File's last access time in the format specified by k, which is either `@` or a directive for the C `strftime` function. The possible values for k are listed below; some of them might not be available on all systems, due to differences in `strftime` between systems.

`@` seconds since Jan. 1, 1970, 00:00 GMT, with fractional part.

Time fields: **H** hour (00..23) **I** hour (01..12) **k** hour (0..23)

l hour (1..12) **M** minute (00..59) **p** locale's AM or PM

r time, 12-hour (hh:mm:ss [AP]M) **S** Second (00.00 .. 61.00). There is a fractional part.

T time, 24-hour (hh:mm:ss)

+ Date and time, separated by `+`, for example `'2004-04-28+22:22:05.0'`. This is a GNU extension. The time is given in the current timezone (which may be affected by setting the `TZ` environment variable). The seconds field includes a fractional part.

X locale's time representation (H:M:S) **Z** time zone (e.g., EDT), or nothing if no time zone is determinable

Date fields:

a locale's abbreviated weekday name (Sun..Sat) **A** locale's full weekday name, variable length (Sunday..Saturday)

b locale's abbreviated month name (Jan..Dec) **B** locale's full month name, variable length (January..December)

c locale's date and time (Sat Nov 04 12:02:33 EST 1989). The format is the same as for **ctime**(3) and so to preserve compatibility

with that format, there is no fractional part in the seconds field.

d day of month (01..31) **D** date (mm/dd/yy) **h** same as **b** **j** day of year (001..366)

m month (01..12) **U** week number of year with Sunday as first day of week (00..53) **w** day of week (0..6)

W week number of year with Monday as first day of week (00..53) **x** locale's date representation (mm/dd/yy)

y last two digits of year (00..99) **Y** year (1970...)

%b The amount of disk space used for this file in 512-byte blocks. Since disk space is allocated in multiples of the filesystem block size this is usually greater than %s/512, but it can also be smaller if the file is a sparse file.

%c File's last status change time in the format returned by the C ``ctime'` function.

%Ck File's last status change time in the format specified by **k**, which is the same as for **%A**.

%d File's depth in the directory tree; 0 means the file is a starting-point.

%D The device number on which the file exists (the `st_dev` field of struct `stat`), in decimal.

%f File's name with any leading directories removed (only the last element).

%F Type of the filesystem the file is on; this value can be used for `-fstype`.

%g File's group name, or numeric group ID if the group has no name.

%G File's numeric group ID.

%h Leading directories of file's name (all but the last element). If the file name contains no slashes (since it is in the current directory) the **%h** specifier expands to `."`.

%H Starting-point under which file was found.

%i File's inode number (in decimal).

%k The amount of disk space used for this file in 1K blocks. Since disk space is allocated in multiples of the filesystem block size this is usually greater than %s/1024, but it can also be smaller if the file is a sparse file.

%l Object of symbolic link (empty string if file is not a symbolic link).

%m File's permission bits (in octal). This option uses the 'traditional' numbers which most Unix implementations use, but if your particular implementation uses an unusual ordering of octal permissions bits, you will see a difference between the actual value of the file's mode and the output of **%m**. Normally you will want to have a leading zero on this number, and to do this, you should use the **#** flag (as in, for example, `'%#m'`).

%M File's permissions (in symbolic form, as for **ls**). This directive is supported in `findutils` 4.2.5 and later.

%n Number of hard links to file. **%p** File's name.

%P File's name with the name of the starting-point under which it was found removed.

%s File's size in bytes. **%S** File's sparseness. This is calculated as $(\text{BLOCKSIZE} * \text{st_blocks} / \text{st_size})$. The exact value you will get for an ordinary file of a certain length is system-dependent. However, normally sparse files will have values less than 1.0, and files which use indirect blocks may have a value which is greater than 1.0. The value used for `BLOCKSIZE` is system-dependent, but is usually 512 bytes. If the file size is zero, the value printed is undefined. On systems which lack support for `st_blocks`, a file's sparseness is assumed to be 1.0.

%t File's last modification time in the format returned by the C ``ctime'` function.

%Tk File's last modification time in the format specified by **k**, which is the same as for **%A**.

%u File's user name, or numeric user ID if the user has no name.

%U File's numeric user ID.

%y File's type (like in **ls -l**), U=unknown type (shouldn't happen)

%Y File's type (like **%y**), plus follow symlinks: L=loop, N=nonexistent

%Z (SELinux only) file's security context.

%{ %[(Reserved for future use.

A `'%'` character followed by any other character is discarded, but the other character is printed (don't rely on this, as further format characters may be introduced). A `'%'` at the end of the format argument causes undefined behaviour since there is no following character. In some locales, it may hide your door keys, while in others it may remove the final page from the novel you are reading.

The **%m** and **%d** directives support the **#**, **0** and **+** flags, but the other directives do not, even if they print numbers. Numeric directives that do not support these flags include **G**, **U**, **b**, **D**, **k** and **n**. The `'-'` format flag is supported and changes the alignment of a field from right-justified (which is the default) to left-justified.

See the **UNUSUAL FILENAMES** section for information about how unusual characters in filenames are handled.

-prune True; if the file is a directory, do not descend into it. If **-depth** is given, false; no effect. Because **-delete** implies **-depth**, you cannot usefully use **-prune** and **-delete** together.

-quit Exit immediately. No child processes will be left running, but no more paths specified on the command line will be processed. For example, `find /tmp/foo /tmp/bar -print -quit` will print only `/tmp/foo`. Any command lines which have been built up with **-execdir ... {} +** will be invoked before **find** exits. The exit status may or may not be zero, depending on whether an error has already occurred.

OPERATORS

Listed in order of decreasing precedence: (**expr**)

Force precedence. Since parentheses are special to the shell, you will normally need to quote them. Many of the examples in this manual page use backslashes for this purpose: `'\(...\)'` instead of `'(...)'`.

! expr True if **expr** is false. This character will also usually need protection from interpretation by the shell.

-not expr Same as **! expr**, but not POSIX compliant.

expr1 expr2 - Two expressions in a row are taken to be joined with an implied "and"; **expr2** is not evaluated if **expr1** is false.

expr1 -a expr2 - Same as **expr1 expr2**.

expr1 -and expr2 - Same as **expr1 expr2**, but not POSIX compliant.

expr1 -o expr2 - Or; **expr2** is not evaluated if **expr1** is true.

expr1 -or expr2 - Same as **expr1 -o expr2**, but not POSIX compliant.

expr1 , expr2 - List; both **expr1** and **expr2** are always evaluated. The value of **expr1** is discarded; the value of the list is the value of **expr2**. The comma operator can be useful for searching for several different types of thing, but traversing the filesystem hierarchy only once. The **-fprint** action can be used to list the various matched items into several different output files.

UNUSUAL FILENAMES

Many of the actions of **find** result in the printing of data which is under the control of other users. This includes file names, sizes, modification times and so forth. File names are a potential problem since they can contain any character except `\0` and `\/`. Unusual characters in file names can do unexpected and often undesirable things to your terminal (for example, changing the settings of your function keys on some terminals). Unusual characters are handled differently by various actions, as described below.

-print0, -fprint0 - Always print the exact filename, unchanged, even if the output is going to a terminal.

-ls, -fls - Unusual characters are always escaped. White space, backslash, and double quote characters are printed using C-style escaping (for example `\f`, `\"`). Other unusual characters are printed using an octal escape. Other printable characters (for **-ls** and **-fls** these are the characters between octal 041 and 0176) are printed as-is.

-printf, -fprintf - If the output is not going to a terminal, it is printed as-is. Otherwise, the result depends on which directive is in use. The directives `%D`, `%F`, `%g`, `%G`, `%H`, `%Y`, and `%y` expand to values which are not under control of files' owners, and so are printed as-is. The directives `%a`, `%b`, `%c`, `%d`, `%i`, `%k`, `%m`, `%M`, `%n`, `%s`, `%t`, `%u` and `%U` have values which are under the control of files' owners but which cannot be used to send arbitrary data to the terminal, and so these are printed as-is. The directives `%f`, `%h`, `%l`, `%p` and `%P` are quoted. This quoting is performed in the same way as for GNU **ls**. This is not the same quoting mechanism as the one used for **-ls** and **-fls**. If you are able to decide what format to use for the output of **find** then it is normally better to use `\0` as a terminator than to use newline, as file names can contain white space and newline characters. The setting of the `'LC_CTYPE'` environment variable is used to determine which characters need to be quoted.

-print, -fprint - Quoting is handled in the same way as for **-printf** and **-fprintf**. If you are using **find** in a script or in a situation where the matched files might have arbitrary names, you should consider using **-print0** instead of **-print**.

The **-ok** and **-okdir** actions print the current filename as-is. This may change in a future release.

STANDARDS CONFORMANCE

For closest compliance to the POSIX standard, you should set the `POSIXLY_CORRECT` environment variable. The following options are specified in the POSIX standard (IEEE Std 1003.1, 2003 Edition):

-H This option is supported. **-L** This option is supported.

-name This option is supported, but POSIX conformance depends on the POSIX conformance of the system's **fnmatch(3)** library function. As of findutils-4.2.2, shell metacharacters (`*`, `?` or `[]` for example) will match a leading `.`, because IEEE PASC interpretation 126 requires this. This is a change from previous versions of findutils.

-type Supported. POSIX specifies `'b'`, `'c'`, `'d'`, `'l'`, `'p'`, `'f'` and `'s'`. GNU find also supports `'D'`, representing a Door, where the OS provides these.

-ok Supported. Interpretation of the response is according to the "yes" and "no" patterns selected by setting the `'LC_MESSAGES'` environment variable. When the `'POSIXLY_CORRECT'` environment variable is set, these patterns are taken system's definition of a positive (yes) or negative (no) response. See the system's documentation for **nl_langinfo(3)**, in particular

YESEXPR and NOEXPR. When `'POSIXLY_CORRECT'` is not set, the patterns are instead taken from **find**'s own message catalogue.

-newer Supported. If the file specified is a symbolic link, it is always dereferenced. This is a change from previous behaviour, which used to take the relevant time from the symbolic link; see the HISTORY section below.

-perm Supported. If the `POSIXLY_CORRECT` environment variable is not set, some mode arguments (for example `+a+x`) which are not valid in POSIX are supported for backward-compatibility.

Other predicates The predicates **-atime**, **-ctime**, **-depth**, **-group**, **-links**, **-mtime**, **-nogroup**, **-nouser**, **-print**, **-prune**, **-size**, **-user** and **-xdev** `'-atime'`, `'-ctime'`, `'-depth'`, `'-group'`, `'-links'`, `'-mtime'`, `'-nogroup'`, `'-nouser'`, `'-perm'`, `'-print'`, `'-prune'`, `'-size'`, `'-user'` and `'-xdev'`, are all supported. The POSIX standard specifies parentheses `'('`, `')'`, negation `'!'` and the `'and'` and `'or'` operators (**-a**, **-o**). All other options, predicates, expressions and so forth are extensions beyond the POSIX standard. Many of these extensions are not unique to GNU find, however.

The POSIX standard requires that **find** detects loops: The **find** utility shall detect infinite loops; that is, entering a previously visited directory that is an ancestor of the last file encountered. When it detects an infinite loop, find shall write a diagnostic message to standard error and shall either recover its position in the hierarchy or terminate.

GNU **find** complies with these requirements. The link count of directories which contain entries which are hard links to an ancestor will often be lower than they otherwise should be. This can mean that GNU find will sometimes optimise away the visiting of a subdirectory which is actually a link to an ancestor. Since **find** does not actually enter such a subdirectory, it is allowed to avoid emitting a diagnostic message. Although this behaviour may be somewhat confusing, it is unlikely that anybody actually depends on this behaviour. If the leaf optimisation

has been turned off with **-noleaf**, the directory entry will always be examined and the diagnostic message will be issued where it is appropriate. Symbolic links cannot be used to create filesystem cycles as such, but if the **-L** option or the **-follow** option is in use, a diagnostic message is issued when **find** encounters a loop of symbolic links. As with loops containing hard links, the leaf optimisation will often mean that **find** knows that it doesn't need to call **stat()** or **chdir()** on the symbolic link, so this diagnostic is frequently not necessary. The **-d** option is supported for compatibility with various BSD systems, but you should use the POSIX-compliant option **-depth** instead. The `POSIXLY_CORRECT` environment variable does not affect the behaviour of the **-regex** or **-iregex** tests because those tests aren't specified in the POSIX standard.

ENVIRONMENT VARIABLES

LANG - Provides a default value for the internationalization variables that are unset or null.

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

LC_COLLATE - The POSIX standard specifies that this variable affects the pattern matching to be used for the **-name** option. GNU find uses the **fnmatch(3)** library function, and so support for **'LC_COLLATE'** depends on the system library. This variable also affects the interpretation of the response to **-ok**; while the **'LC_MESSAGES'** variable selects the actual pattern used to interpret the response to **-ok**, the interpretation of any bracket expressions in the pattern will be affected by **'LC_COLLATE'**.

LC_CTYPE This variable affects the treatment of character classes used in regular expressions and also with the **-name** test, if the system's **fnmatch(3)** library function supports this. This variable also affects the interpretation of any character classes in the regular expressions used to interpret the response to the prompt issued by **-ok**. The **'LC_CTYPE'** environment variable will also affect which characters are considered to be unprintable when filenames are printed; see the section UNUSUAL FILENAMES.

LC_MESSAGES - Determines the locale to be used for internationalised messages. If the **'POSIXLY_CORRECT'** environment variable is set, this also determines the interpretation of the response to the prompt made by the **-ok** action.

NLSPATH - Determines the location of the internationalisation message catalogues.

PATH - Affects the directories which are searched to find the executables invoked by **-exec**, **-execdir**, **-ok** and **-okdir**.

POSIXLY_CORRECT - Determines the block size used by **-ls** and **-fls**. If **POSIXLY_CORRECT** is set, blocks are units of 512 bytes. Otherwise they are units of 1024 bytes. Setting this variable also turns off warning messages (that is, implies **-nowarn**) by default, because POSIX requires that apart from the output for **-ok**, all messages printed on stderr are diagnostics and must result in a non-zero exit status.

When **POSIXLY_CORRECT** is not set, **-perm +zzz** is treated just like **-perm /zzz** if **+zzz** is not a valid symbolic mode. When **POSIXLY_CORRECT** is set, such constructs are treated as an error. When **POSIXLY_CORRECT** is set, the response to the prompt made by the **-ok** action is interpreted according to the system's message catalogue, as opposed to according to **find**'s own message translations.

TZ Affects the time zone used for some of the time-related format directives of **-printf** and **-fprintf**.

EXAMPLES

find /tmp -name core -type f -print | xargs /bin/rm -f - Find files named **core** in or below the directory **/tmp** and delete them. Note that this will work incorrectly if there are any filenames containing newlines, single or double quotes, or spaces.

find /tmp -name core -type f -print0 | xargs -0 /bin/rm -f - Find files named **core** in or below the directory **/tmp** and delete them, processing filenames in such a way that file or directory names containing single or double quotes, spaces or newlines are correctly handled. The **-name** test comes before the **-type** test in order to avoid having to call **stat(2)** on every file.

find . -type f -exec file '{}' \; Runs **'file'** on every file in or below the current directory. Notice that the braces are enclosed in single quote marks to protect them from interpretation as shell script punctuation. The semicolon is similarly protected by the use of a backslash, though single quotes could have been used in that case also.

**find / \(-perm -4000 -fprintf /root/suid.txt '%#m %u %p\n' \) , **
\(-size +100M -fprintf /root/big.txt '%-10s %p\n' \) - Traverse the filesystem just once, listing setuid files and directories into **/root/suid.txt** and large files into **/root/big.txt**.

find \$HOME -mtime 0 - Search for files in your home directory which have been modified in the last twenty-four hours. This command works this way because the time since each file was last modified is divided by 24 hours and any remainder is discarded. That means that to match **-mtime 0**, a file will have to have a modification in the past which is less than 24 hours ago.

find /sbin /usr/sbin -executable ! -readable -print - Search for files which are executable but not readable.

find . -perm 664 - Search for files which have read and write permission for their owner, and group, but which other users can read but not write to. Files which meet these criteria but have other permissions bits set (for example if someone can execute the file) will not be matched.

find . -perm -664 - Search for files which have read and write permission for their owner and group, and which other users can read, without regard to the presence of any extra permission bits (for example the executable bit). This will match a file which has mode 0777, for example.

find . -perm /222 - Search for files which are writable by somebody (their owner, or their group, or anybody else).

find . -perm /220

find . -perm /u+w,g+w

find . -perm /u=w,g=w

All three of these commands do the same thing, but the first one uses the octal representation of the file mode, and the other two use the symbolic form. These commands all search for files which are writable by either their owner or their group. The files don't have to be writable by both the owner and group to be matched; either will do.

find . -perm -220

find . -perm -g+w,u+w

Both these commands do the same thing; search for files which are writable by both their owner and their group.

find . -perm -444 -perm /222 ! -perm /111

find . -perm -a+r -perm /a+w ! -perm /a+x

These two commands both search for files that are readable for everybody (**-perm -444** or **-perm -a+r**), have at least one write bit set (**-perm /222** or **-perm /a+w**) but are not executable for anybody (**! -perm /111** and **! -perm /a+x** respectively).

cd /source-dir

find . -name .snapshot -prune -o \(! -name *~ -print0 \)

cpio -pmd0 /dest-dir

This command copies the contents of **/source-dir** to **/dest-dir**, but omits files and directories named **.snapshot** (and anything in them). It also omits files or directories whose name ends in **~**, but not their contents. The construct **-prune -o \{ ... -print0 \}** is quite common. The idea here is that the expression before **-prune** matches things which are to be pruned. However, the **-prune** action itself returns true, so the following **-o** ensures that the right hand side is evaluated only for those directories which didn't get pruned (the contents of the pruned directories are not even visited, so their contents are irrelevant). The expression on the right hand side of the **-o** is in parentheses only for clarity. It emphasises that the **-print0** action takes place only for things that didn't have **-prune** applied to them. Because the default **`and'** condition between tests binds more tightly than **-o**, this is the default anyway, but the parentheses help to show what is going on.

```
find repo/ -exec test -d {}/.svn \; -or \
-exec test -d {}/.git \; -or -exec test -d {}/CVS \; \
-print -prune
```

Given the following directory of projects and their associated SCM administrative directories, perform an efficient search for the projects' roots:

```
repo/project1/CVS
repo/gnu/project2/.svn
repo/gnu/project3/.svn
repo/gnu/project3/src/.svn
repo/project4/.git
```

In this example, **-prune** prevents unnecessary descent into directories that have already been discovered (for example we do not search **project3/src** because we already found **project3/.svn**), but ensures sibling directories (**project2** and **project3**) are found.

EXIT STATUS

find exits with status 0 if all files are processed successfully, greater than 0 if errors occur. This is deliberately a very broad description, but if the return value is non-zero, you should not rely on the correctness of the results of **find**.

When some error occurs, **find** may stop immediately, without completing all the actions specified. For example, some starting points may not have been examined or some pending program invocations for **-exec ... {} +** or **-execdir ... {} +** may not have been performed.

SEE ALSO **locate**(1), **locatedb**(5), **updatedb**(1), **xargs**(1), **chmod**(1), **fnmatch**(3), **regex**(7), **stat**(2), **lstat**(2), **ls**(1), **printf**(3), **strptime**(3), **ctime**(3)

The full documentation for **find** is maintained as a Texinfo manual. If the **info** and **find** programs are properly installed at your site, the command **info find** should give you access to the complete manual.

HISTORY As of findutils-4.2.2, shell metacharacters (**`***, **`?'** or **`[]'** for example) used in filename patterns will match a leading **`.'**, because IEEE POSIX interpretation 126 requires this.

As of findutils-4.3.3, **-perm /000** now matches all files instead of none.

Nanosecond-resolution timestamps were implemented in findutils-4.3.3.

As of findutils-4.3.11, the **-delete** action sets **find**'s exit status to a nonzero value when it fails. However, **find** will not exit immediately. Previously, **find**'s exit status was unaffected by the failure of **-delete**.

Feature	Added in	Also occurs in
-newerXY	4.3.3	BSD
-D	4.3.1	
-O	4.3.1	
-readable	4.3.0	
-writable	4.3.0	
-executable	4.3.0	
-regextype	4.2.24	
-exec ... +	4.2.12	POSIX
-execdir	4.2.12	BSD
-okdir	4.2.12	
-samefile	4.2.11	
-H	4.2.5	POSIX
-L	4.2.5	POSIX
-P	4.2.5	BSD
-delete	4.2.3	
-quit	4.2.3	
-d	4.2.3	BSD
-wholename	4.2.0	
-iwholename	4.2.0	
-ignore_readdir_race	4.2.0	
-fls	4.0	
-lname	3.8	
-iname	3.8	
-ipath	3.8	
-iregex	3.8	

NON-BUGS

```
$ find . -name *.c -print
find: paths must precede expression
Usage: find [-H] [-L] [-P] [-Olevel] [-D help|tree|search|stat|rates|opt|exec] [path...] [expression]
```

This happens because ***_c** has been expanded by the shell resulting in **find** actually receiving a command line like this:

```
find . -name bigram.c code.c frcode.c locate.c -print
```

That command is of course not going to work. Instead of doing things this way, you should enclose the pattern in quotes or escape the wildcard:

```
$ find . -name '*.c' -print
$ find . -name \*.c -print
```

BUGS

There are security problems inherent in the behaviour that the POSIX standard specifies for **find**, which therefore cannot be fixed. For example, the **-exec** action is inherently insecure, and **-execdir** should be used instead. Please see **Finding Files** for more information.

The environment variable **LC_COLLATE** has no effect on the **-ok** action.

The syntax **-perm +MODE** was removed in findutils-4.5.12, in favour of **-perm /MODE**. The **+MODE** syntax had been deprecated since findutils-4.2.21 which was released in 2005.

NAME **ps** - report a snapshot of the current processes. (8 pages)

SYNOPSIS **ps** [[options](#)]

DESCRIPTION **ps** displays information about a selection of the active processes. If you want a repetitive update of the selection and the displayed information, use [top](#)(1) instead. This version of **ps** accepts several kinds of options:

- 1 UNIX options, which may be grouped and must be preceded by a dash.
- 2 BSD options, which may be grouped and must not be used with a dash.
- 3 GNU long options, which are preceded by two dashes.

Options of different types may be freely mixed, but conflicts can appear. There are some synonymous options, which are functionally identical, due to the many standards and **ps** implementations that this **ps** is compatible with.

Note that "**ps -aux**" is distinct from "**ps aux**". The POSIX and UNIX standards require that "**ps -aux**" print all processes owned by a user named "x", as well as printing all processes that would be selected by the **-a** option. If the user named "x" does not exist, this **ps** may interpret the command as "**ps aux**" instead and print a warning. This behavior is intended to aid in transitioning old scripts and habits. It is fragile, subject to change, and thus should not be relied upon.

By default, **ps** selects all processes with the same effective user ID (euid=EUID) as the current user and associated with the same terminal as the invoker. It displays the process ID (pid=PID), the terminal associated with the process (tname=TTY), the cumulated CPU time in [DD-]hh:mm:ss format (time=TIME), and the executable name (ucmd=CMD). Output is unsorted by default.

The use of BSD-style options will add process state (stat=STAT) to the default display and show the command args (args=COMMAND) instead of the executable name. You can override this with the **PS_FORMAT** environment variable. The use of BSD-style options will also change the process selection to include processes on other terminals (TTYs) that are owned by you; alternately, this may

be described as setting the selection to be the set of all processes filtered to exclude processes owned by other users or not on a terminal. These effects are not considered when options are described as being "identical" below, so **-M** will be considered identical to **Z** and so on

Except as described below, process selection options are additive. The default selection is discarded, and then the selected processes are added to the set of processes to be displayed. A process will thus be shown if it meets any of the given selection criteria.

EXAMPLES

To see every process on the system using standard syntax:

```
ps -e  
ps -ef  
ps -eF  
ps -ely
```

To see every process on the system using BSD syntax:

```
ps ax  
ps axu
```

To see every process with a user-defined format:

```
ps -eo pid,tid,class,rtprio,ni,pri,psr,pcpu,stat,wchan:14,comm  
ps axo stat,euid,ruid,tt,tpgid,sess,pgrp,ppid,pid,pcpu,comm  
ps -Ao pid,tt,user,fname,tmout,f,wchan
```

Print only the process IDs of syslogd:

```
ps -C syslogd -o pid=
```

Print only the name of PID 42:

```
ps -q 42 -o comm=
```

To print a process tree:

```
ps -ejH  
ps axjf
```

To get info about threads:

```
ps -eLf  
ps axms
```

To get security info:

```
ps -eo euser,ruser,suser,fuser,f,comm,label  
ps axZ  
ps -eM
```

To see every process running as root (real & effective ID) in user format:

```
ps -U root -u root u
```

SIMPLE PROCESS SELECTION

a Lift the BSD-style "only yourself" restriction, which is imposed upon the set of all processes when some BSD-style (without "-") options are used or when the **ps** personality setting is

BSD-like. The set of processes selected in this manner is in addition to the set of processes selected by other means. An alternate description is that this option causes **ps** to list

all processes with a terminal (tty), or to list all processes when used together with the **x** option.

-A Select all processes. Identical to **-e**.

-a Select all processes except both session leaders (see [getsid](#)(2)) and processes not associated with a terminal.

-d Select all processes except session leaders.

--deselect

Select all processes except those that fulfill the specified conditions (negates the selection). Identical to **-N**.

-e Select all processes. Identical to **-A**.

g Really all, even session leaders. This flag is obsolete and may be discontinued in a future release. It is normally implied by the **a** flag, and is only useful when operating in the **sunos4** personality.

-N Select all processes except those that fulfill the specified conditions (negates the selection). Identical to **--deselect**.

T Select all processes associated with this terminal. Identical to the **t** option without any argument.

r Restrict the selection to only running processes.

x Lift the BSD-style "must have a tty" restriction, which is imposed upon the set of all processes when some BSD-style (without "-") options are used or when the **ps** personality setting is BSD-like. The set of processes selected in this manner is in addition to the set of processes selected by other means. An alternate description is that this option causes **ps** to list all processes owned by you (same EUID as **ps**), or to list all processes when used together with the **a** option.

PROCESS SELECTION BY LIST

These options accept a single argument in the form of a blank-separated or comma-separated list. They can be used multiple times. For example: **ps -p "1 2" -p 3,4**

-123 Identical to **--pid 123**. **123** Identical to **--pid 123**.

-C cmdlist - Select by command name. This selects the processes whose executable name is given in **cmdlist**.

-G grplist - Select by real group ID (RGID) or name. This selects the processes whose real group name or ID is in the **grplist** list. The real group ID identifies the group of the user who created the process, see **getuid(2)**.

-g grplist - Select by session OR by effective group name. Selection by session is specified by many standards, but selection by effective group is the logical behavior that several other operating systems use. This **ps** will select by session when the list is completely numeric (as sessions are). Group ID numbers will work only when some group names are also specified. See the **-s** and **--group** options.

--Group grplist - Select by real group ID (RGID) or name. Identical to **-G**.

--group grplist - Select by effective group ID (EGID) or name. This selects the processes whose effective group name or ID is in **grplist**. The effective group ID describes the group whose file access permissions are used by the process (see **getegid(2)**). The **-g** option is often an alternative to **--group**.

p pidlist - Select by process ID. Identical to **-p** and **--pid**.

-p pidlist - Select by PID. This selects the processes whose process ID numbers appear in **pidlist**. Identical to **p** and **--pid**.

--pid pidlist - Select by process ID. Identical to **-p** and **p**.

--ppid pidlist - Select by parent process ID. This selects the processes with a parent process ID in **pidlist**. That is, it selects processes that are children of those listed in **pidlist**.

q pidlist - Select by process ID (quick mode). Identical to **-q** and **--quick-pid**.

-q pidlist - Select by PID (quick mode). This selects the processes whose process ID numbers appear in **pidlist**. With this option **ps** reads the necessary info only for the pids listed in the **pidlist** and doesn't apply additional filtering rules. The order of pids is unsorted and preserved. No additional selection options, sorting and forest type listings are allowed in this mode. Identical to **q** and **--quick-pid**.

--quick-pid pidlist - Select by process ID (quick mode). Identical to **-q** and **q**.

-s sesslist - Select by session ID. This selects the processes with a session ID specified in **sesslist**.

--sid sesslist - Select by session ID. Identical to **-s**.

t ttylist - Select by tty. Nearly identical to **-t** and **--tty**, but can also be used with an empty **ttylist** to indicate the terminal associated with **ps**. Using the **T** option is considered cleaner than using **t** with an empty **ttylist**.

-t ttylist - Select by tty. This selects the processes associated with the terminals given in **ttylist**. Terminals (ttys, or screens for text output) can be specified in several forms: **/dev/ttyS1**, **ttyS1**, **S1**. A plain "-" may be used to select processes not attached to any terminal.

--tty ttylist - Select by terminal. Identical to **-t** and **t**.

U userlist - Select by effective user ID (EUID) or name. This selects the processes whose effective user name or ID is in **userlist**. The effective user ID describes the user whose file access permissions are used by the process (see **geteuid(2)**). Identical to **-u** and **--user**.

-U userlist - Select by real user ID (RUID) or name. It selects the processes whose real user name or ID is in the **userlist** list. The real user ID identifies the user who created the process, see **getuid(2)**.

-u userlist - Select by effective user ID (EUID) or name. This selects the processes whose effective user name or ID is in **userlist**.

The effective user ID describes the user whose file access permissions are used by the process (see **geteuid(2)**). Identical to **U** and **--user**.

--User userlist - Select by real user ID (RUID) or name. Identical to **-U**.

--user userlist - Select by effective user ID (EUID) or name. Identical to **-u** and **U**.

OUTPUT FORMAT CONTROL

These options are used to choose the information displayed by **ps**. The output may differ by personality.

-c Show different scheduler information for the **-i** option.

--context - Display security context format (for SELinux).

-f Do full-format listing. This option can be combined with many other UNIX-style options to add additional columns. It also causes the command arguments to be printed. When used with **-L**, the NLWP (number of threads) and LWP (thread ID) columns will be added. See the **c** option, the format keyword **args**, and the format keyword **comm**.

-F Extra full format. See the **-f** option, which **-F** implies.

--format format - user-defined format. Identical to **-o** and **o**.

j BSD job control format. **-j** Jobs format. **l** Display BSD long format. **-l** Long format. The **-y** option is often useful with this.

-M Add a column of security data. Identical to **Z** (for SELinux).

O format is preloaded **o** (overloaded). The BSD **O** option can act like **-O** (user-defined output format with some common fields predefined) or can be used to specify sort order. Heuristics are used to determine the behavior of this option. To ensure that the desired behavior is obtained (sorting or formatting), specify the option in some other way (e.g. with **-O** or **--sort**).

When used as a formatting option, it is identical to **-O**, with the BSD personality.

-O format - Like **-o**, but preloaded with some default columns. Identical to **-o pid,format,state,tname,time,command** or **-o pid,format,tname,time,cmd**, see **-o** below.

o format - Specify user-defined format. Identical to **-o** and **--format**.

-o format User-defined format. **format** is a single argument in the form of a blank-separated or comma-separated list, which offers a way to specify individual output columns. The recognized keywords are described in the **STANDARD FORMAT SPECIFIERS** section below. Headers may be renamed (**ps -o pid,ruser=RealUser -o comm=Command**) as desired. If all column headers are empty (**ps -o pid= -o comm=**) then the header line will not be output. Column width will increase as needed for wide headers; this may be used to widen up columns such as WCHAN (**ps -o pid,wchan=WIDE-WCHAN-COLUMN -o comm**). Explicit width control (**ps opid,wchan:42,cmd**) is offered too. The behavior of **ps -o pid=X,comm=Y** varies with personality; output may be one column named "X,comm=Y" or two columns named "X" and "Y". Use multiple **-o** options when in doubt. Use the **PS_FORMAT** environment variable to specify a default as desired; DefSysV and DefBSD are macros that may be used to choose the default UNIX or BSD columns.

s Display signal format.

u Display user-oriented format.

v Display virtual memory format.

X Register format.

-y Do not show flags; show rss in place of addr. This option can only be used with **-l**.

Z Add a column of security data. Identical to **-M** (for SELinux).

OUTPUT MODIFIERS

c Show the true command name. This is derived from the name of the executable file, rather than from the argv value. Command arguments and any modifications to them are thus not shown.

This option effectively turns the **args** format keyword into the **comm** format keyword; it is useful with the **-f** format option and with the various BSD-style format options, which all

normally display the command arguments. See the **-f** option, the format keyword **args**, and the format keyword **comm**.

--cols n - Set screen width.

--columns n - Set screen width.

--cumulative - Include some dead child process data (as a sum with the parent).

e Show the environment after the command.

f ASCII art process hierarchy (forest).

--forest - ASCII art process tree.

h No header. (or, one header per screen in the BSD personality). The **h** option is problematic. Standard BSD **ps** uses this option to print a header on each page of output, but older Linux **ps** uses this option to totally disable the header. This version of **ps** follows the Linux usage of not printing the header unless the BSD personality has been selected, in which case it prints a header on each page of output. Regardless of the current personality, you can use the long options **--headers** and **--no-headers** to enable printing headers each page or disable headers entirely, respectively.

-H Show process hierarchy (forest).

--headers Repeat header lines, one per page of output.

k spec Specify sorting order. Sorting syntax is **[+|-]key[. [+|-]key[,...]]**. Choose a multi-letter key from the **STANDARD FORMAT SPECIFIERS** section. The "+" is optional since default

direction is increasing numerical or lexicographic order. Identical to **--sort**.

Examples:

ps jaxkuid,-ppid,+pid

ps axk comm o comm,args

ps kstart_time -ef

--lines n - Set screen height.

n Numeric output for WCHAN and USER (including all types of UID and GID).

--no-headers - Print no header line at all. **--no-heading** is an alias for this option.

O order - Sorting order (overloaded). The BSD **O** option can act like **-O** (user-defined output format with some common fields predefined) or can be used to specify sort order. Heuristics are used to determine the behavior of this option. To ensure that the desired behavior is obtained (sorting or formatting), specify the option in some other way (e.g. with **-O** or **--sort**).

For sorting, obsolete BSD **O** option syntax is **O[+|-]k1[. [+|-]k2[,...]]**. It orders the processes listing according to the multilevel sort specified by the sequence of one-letter short keys **k1**, **k2**, ... described in the **OBSOLETE SORT KEYS** section below. The "+" is currently optional, merely re-iterating the default direction on a key, but may help to distinguish an **O** sort from an **O** format. The "-" reverses direction only on the key it precedes.

--rows n - Set screen height.

S Sum up some information, such as CPU usage, from dead child processes into their parent. This is useful for examining a system where a parent process repeatedly forks off short-lived children to do work.

--sort spec - Specify sorting order. Sorting syntax is **[+|-]key[. [+|-]key[,...]]**. Choose a multi-letter key from the **STANDARD FORMAT SPECIFIERS** section. The "+" is optional since default direction is increasing numerical or lexicographic order. Identical to **k**. For example: **ps jax**

--sort=uid,-ppid,+pid

w Wide output. Use this option twice for unlimited width.

-w Wide output. Use this option twice for unlimited width.

--width n - Set screen width.

THREAD DISPLAY

- H** Show threads as if they were processes.
- L** Show threads, possibly with LWP and NLWP columns.
- m** Show threads after processes.
- m** Show threads after processes.
- T** Show threads, possibly with SPID column.

OTHER INFORMATION

--help section - Print a help message. The section argument can be one of simple, list, output, threads, misc or all. The argument can be shortened to one of the underlined letters as in:

s||o|t|m|a.

--info Print debugging info.

L List all format specifiers.

V Print the procps-ng version.

-V Print the procps-ng version.

--version Print the procps-ng version.

NOTES

This **ps** works by reading the virtual files in /proc. This **ps** does not need to be setuid kmem or have any privileges to run. Do not give this **ps** any special permissions.

CPU usage is currently expressed as the percentage of time spent running during the entire lifetime of a process. This is not ideal, and it does not conform to the standards that **ps**

otherwise conforms to. CPU usage is unlikely to add up to exactly 100%.

The SIZE and RSS fields don't count some parts of a process including the page tables, kernel stack, struct thread_info, and struct task_struct. This is usually at least 20 KiB of memory that is always resident. SIZE is the virtual size of the process (code+data+stack).

Processes marked <defunct> are dead processes (so-called "zombies") that remain because their parent has not destroyed them properly. These processes will be destroyed by init(8) if the parent process exits. If the length of the username is greater than the length of the display column, the username will be truncated. See the -o and -O formatting options to customize length.

Commands options such as **ps -aux** are not recommended as it is a confusion of two different standards. According to the POSIX and UNIX standards, the above command asks to display all processes with a TTY (generally the commands users are running) plus all processes owned by a user named "x". If that user doesn't exist, then **ps** will assume you really meant "**ps aux**".

PROCESS FLAGS

The sum of these values is displayed in the "F" column, which is provided by the **flags** output specifier:

- 1 forked but didn't exec
- 4 used super-user privileges

PROCESS STATE CODES

Here are the different values that the **s**, **stat** and **state** output specifiers (header "STAT" or "S") will display to describe the state of a process:

- D uninterruptible sleep (usually IO)
- I Idle kernel thread
- R running or runnable (on run queue)
- S interruptible sleep (waiting for an event to complete)
- T stopped by job control signal
- t stopped by debugger during the tracing
- W paging (not valid since the 2.6.xx kernel)
- X dead (should never be seen)
- Z defunct ("zombie") process, terminated but not reaped by its parent

For BSD formats and when the **stat** keyword is used, additional characters may be displayed:

- < high-priority (not nice to other users)
- N low-priority (nice to other users)
- L has pages locked into memory (for real-time and custom IO)
- s is a session leader
- l is multi-threaded (using CLONE_THREAD, like NPTL pthreads do)
- + is in the foreground process group

OBSOLETE SORT KEYS

These keys are used by the BSD **O** option (when it is used for sorting). The GNU **--sort** option doesn't use these keys, but the specifiers described below in the **STANDARD FORMAT SPECIFIERS** section. Note that the values used in sorting are the internal values **ps** uses and not the "cooked" values used in some of the output format fields (e.g. sorting on tty will sort into device number, not according to the terminal name displayed). Pipe **ps** output into the **sort(1)** command if you want to sort the cooked values.

KEY	LONG	DESCRIPTION
c	cmd	simple name of executable
C	pcpu	cpu utilization
f	flags	flags as in long format F field
g	pgrp	process group ID
G	tpgid	controlling tty process group ID
j	cutime	cumulative user time
J	cstime	cumulative system time
k	utime	user time
m	min_fit	number of minor page faults
M	maj_fit	number of major page faults
n	cmin_fit	cumulative minor page faults
N	cmaj_fit	cumulative major page faults
o	session	session ID
p	pid	process ID
P	ppid	parent process ID
r	rss	resident set size
R	resident	resident pages
s	size	memory size in kilobytes
S	share	amount of shared pages
t	tty	the device number of the controlling tty
T	start_time	time process was started
U	uid	user ID number
u	user	user name
v	vsize	total VM size in KiB
y	priority	kernel scheduling priority

AIX FORMAT DESCRIPTORS

This **ps** supports AIX format descriptors, which work somewhat like the formatting codes of printf(1) and printf(3). For example, the normal default output can be produced with this: **ps -eo "%p %y %x %c"**. The **NORMAL** codes are described in the next section.

CODE	NORMAL	HEADER
%C	pcpu	%CPU
%G	group	GROUP
%P	ppid	PPID
%U	user	USER
%a	args	COMMAND
%c	comm	COMMAND
%g	rgroup	RGROUP
%n	nice	NI
%p	pid	PID
%r	pgid	PGID
%t	etime	ELAPSED
%u	ruser	RUSER
%x	time	TIME
%y	tty	TTY
%z	vsz	VSZ

STANDARD FORMAT SPECIFIERS

Here are the different keywords that may be used to control the output format (e.g. with option **-o**) or to sort the selected processes with the GNU-style **--sort** option.

For example: **ps -eo pid,user,args --sort user**
This version of **ps** tries to recognize most of the keywords used in other implementations of **ps**.
The following user-defined format specifiers may contain spaces: **args**, **cmd**, **comm**, **command**, **fname**, **ucmd**, **ucomm**, **lstart**, **bsdstart**, **start**.
Some keywords may not be available for sorting.

CODE HEADER DESCRIPTION

%cpu %CPU cpu utilization of the process in "##.#" format. Currently, it is the CPU time used divided by the time the process has been running (cputime/realtime ratio), expressed as a percentage. It will not add up to 100% unless you are lucky. (alias **pcpu**).

%mem %MEM ratio of the process's resident set size to the physical memory on the machine, expressed as a percentage. (alias **pmem**).

args COMMAND command with all its arguments as a string. Modifications to the arguments may be shown. The output in this column may contain spaces. A process marked <defunct> is partly dead, waiting to be fully destroyed by its parent. Sometimes the process args will be unavailable; when this happens, **ps** will instead print the executable name in brackets. (alias **cmd**, **command**). See also the **comm** format keyword, the **-f** option, and the **c** option. When specified last, this column will extend to the edge of the display. If **ps** can not determine display width, as when output is redirected (piped) into a file or another command, the output width is undefined (it may be 80, unlimited, determined by the **TERM** variable, and so on). The **COLUMNS** environment variable or **--cols** option may be used to exactly determine the width in this case. The **w** or **-w** option may be also be used to adjust width.

blocked BLOCKED mask of the blocked signals, see [signal\(7\)](#). According to the width of the field, a 32 or 64-bit mask in hexadecimal format is displayed. (alias **sig_block**, **sigmask**).

bsdstart START time the command started. If the process was started less than 24 hours ago, the output format is " HH:MM", else it is " Mmm:SS" (where Mmm is the three letters of the month). See also **lstart**, **start**, **start_time**, and **stime**.

bsdtime TIME accumulated cpu time, user + system. The display format is usually "MMM:SS", but can be shifted to the right if the process used more than 999 minutes of cpu time.

c C processor utilization. Currently, this is the integer value of the percent usage over the lifetime of the process. (see **%cpu**).

caught CAUGHT mask of the caught signals, see [signal\(7\)](#). According to the width of the field, a 32 or 64 bits mask in hexadecimal format is displayed. (alias **sig_catch**, **sigcatch**).

cgname CGNAME display name of control groups to which the process belongs.

cgroup CGROUP display control groups to which the process belongs.

class CLS scheduling class of the process. (alias **policy**, **cls**). Field's possible values are:
 - not reported
 TS SCHED_OTHER, FF SCHED_FIFO, RR SCHED_RR, B SCHED_BATCH, ISO SCHED_ISO, IDL SCHED_IDLE, DLN SCHED_DEADLINE,
 ? unknown value

cls CLS scheduling class of the process. (alias **policy**, **cls**). Field's possible values are: - not reported,
 TS SCHED_OTHER, FF SCHED_FIFO, RR SCHED_RR, B SCHED_BATCH, ISO SCHED_ISO, IDL SCHED_IDLE, DLN
 SCHED_DEADLINE, ? unknown value

cmd CMD see **args**. (alias **args**, **command**).

comm COMMAND command name (only the executable name). Modifications to the command name will not be shown. A process marked <defunct> is partly dead, waiting to be fully destroyed by its parent. The output in this column may contain spaces. (alias **ucmd**, **ucomm**). See also the **args format keyword**, the **-f** option, and the **c** option. When specified last, this column will extend to the edge of the display. If **ps** can not determine display width, as when output is redirected (piped) into a file or another command, the output width is undefined (it may be 80, unlimited, determined by the **TERM** variable, and so on). The **COLUMNS** environment variable or **--cols** option may be used to exactly determine the width in this case. The **w** or **-w** option may be also be used to adjust width.

command COMMAND See **args**. (alias **args**, **command**).

cp CP per-mill (tenths of a percent) CPU usage. (see **%cpu**).

cputime TIME cumulative CPU time, "[DD-]hh:mm:ss" format. (alias **time**).

cputimes TIME cumulative CPU time in seconds (alias **times**).

drs DRS data resident set size, the amount of physical memory devoted to other than executable code.

egid EGID effective group ID number of the process as a decimal integer. (alias **gid**).

egroup EGROUP effective group ID of the process. This will be the textual group ID, if it can be obtained and the field width permits, or a decimal representation otherwise. (alias **group**).

eip EIP instruction pointer.

esp ESP stack pointer.

etime ELAPSED elapsed time since the process was started, in the form [[DD-]hh:]mm:ss.

etimes ELAPSED elapsed time since the process was started, in seconds.

euid EUID effective user ID (alias **uid**).

euser EUSER effective user name. This will be the textual user ID, if it can be obtained and the field width permits, or a decimal representation otherwise. The **n** option can be used to force the decimal representation. (alias **uname**, user).

f F flags associated with the process, see the **PROCESS FLAGS** section. (alias **flag**, **flags**).

fgid FGID filesystem access group ID. (alias **fsgid**).

fgroup FGROU filesystem access group ID. This will be the textual group ID, if it can be obtained and the field width permits, or a decimal representation otherwise. (alias **fsgroup**).

flag F see **f**. (alias **f**, **flags**).

flags F see **f**. (alias **f**, **flag**).

fname COMMAND first 8 bytes of the base name of the process's executable file. The output in this column may contain spaces.

fuid FUID filesystem access user ID. (alias **fsuid**).

fuser FUSER filesystem access user ID. This will be the textual user ID, if it can be obtained and the field width permits, or a decimal representation otherwise.

gid GID see **egid**. (alias **egid**).

group GROUP see **egroup**. (alias **egroup**).

ignored IGNORED mask of the ignored signals, see [signal\(7\)](#). According to the width of the field, a 32 or 64 bits mask in hexadecimal format is displayed. (alias **sig_ignore**, **sigignore**).

ipcns IPCNS Unique inode number describing the namespace the process belongs to. See namespaces(7).

label LABEL security label, most commonly used for SELinux context data. This is for the Mandatory Access Control ("MAC") found on high-security systems.

Istart STARTED time the command started. See also **bsdstart**, **start**, **start_time**, and **stime**.

lsession SESSION displays the login session identifier of a process, if systemd support has been included.

luid LUID displays Login ID associated with a process.

lwp LWP light weight process (thread) ID of the dispatchable entity (alias **spid**, **tid**). See **tid** for additional information.

lxc LXC The name of the lxc container within which a task is running. If a process is not running inside a container, a dash ('-') will be shown.

machine MACHINE displays the machine name for processes assigned to VM or container, if systemd support has been included.

majflt MAJFLT The number of major page faults that have occurred with this process.

minflt MINFLT The number of minor page faults that have occurred with this process.

mntns MNTNS Unique inode number describing the namespace the process belongs to. See namespaces(7).

netns NETNS Unique inode number describing the namespace the process belongs to. See namespaces(7).

ni NI nice value. This ranges from 19 (nicest) to -20 (not nice to others), see nice(1). (alias **nice**).

nice NI see **ni**.(alias **ni**).

nlwp NLWP number of lwps (threads) in the process. (alias **thcount**).

numa NUMA The node associated with the most recently used processor. A -1 means that NUMA information is unavailable.

nwchan WCHAN address of the kernel function where the process is sleeping (use **wchan** if you want the kernel function name).

Running tasks will display a dash ('-') in this column.

oid OWNER displays the Unix user identifier of the owner of the session of a process, if systemd support has been included.

pcpu %CPU see **%cpu**. (alias **%cpu**).

pending PENDING mask of the pending signals. See signal(7). Signals pending on the process are distinct from signals pending on individual threads. Use the **m** option or the **-m** option to see both. According to the width of the field, a 32 or 64 bits mask in hexadecimal format is displayed. (alias **sig**).

pgid PGID process group ID or, equivalently, the process ID of the process group leader. (alias **pgrp**).

pgrp PGRP see **pgid**. (alias **pgid**).

pid PID a number representing the process ID (alias **tgid**).

pidns PIDNS Unique inode number describing the namespace the process belongs to. See namespaces(7).

pmem %MEM see **%mem**. (alias **%mem**).

policy POL scheduling class of the process. (alias **class**, **cls**). Possible values are:
- not reported, TS SCHED_OTHER, FF SCHED_FIFO, RR SCHED_RR, B SCHED_BATCH, ISO SCHED_ISO, IDL SCHED_IDLE
DLN SCHED_DEADLINE, ? unknown value

ppid PPID parent process ID.

pri PRI priority of the process. Higher number means lower priority.

psr PSR processor that process is currently assigned to.

rgid RGID real group ID.

rgroup RGROUP real group name. This will be the textual group ID, if it can be obtained and the field width permits, or a decimal representation otherwise.

rss RSS resident set size, the non-swapped physical memory that a task has used (in kiloBytes). (alias **rssize**, **rsz**).

rssize RSS see **rss**. (alias **rss**, **rsz**).

rsz RSZ see **rss**. (alias **rss**, **rssize**).

rtprio RTPRIO realtime priority.

ruid RUID real user ID.

ruser RUSER real user ID. This will be the textual user ID, if it can be obtained and the field width permits, or a decimal representation otherwise.

s S minimal state display (one character). See section **PROCESS STATE CODES** for the different values. See also **stat** if you want additional information displayed. (alias **state**).

sched SCH scheduling policy of the process. The policies SCHED_OTHER (SCHED_NORMAL), SCHED_FIFO, SCHED_RR, SCHED_BATCH, SCHED_ISO, SCHED_IDLE and SCHED_DEADLINE are respectively displayed as 0, 1, 2, 3, 4, 5 and 6.

seat SEAT displays the identifier associated with all hardware devices assigned to a specific workplace, if systemd support has been included.

sess SESS session ID or, equivalently, the process ID of the session leader. (alias **session**, **sid**).

sgi_p P processor that the process is currently executing on. Displays "***" if the process is not currently running or runnable.

sgid SGID saved group ID. (alias **svgid**).

sgroup SGROUP saved group name. This will be the textual group ID, if it can be obtained and the field width permits, or a decimal representation otherwise.

sid SID see **sess**. (alias **sess**, **session**).

sig PENDING see **pending**. (alias **pending**, **sig_pending**).

sigcatch CAUGHT see **caught**. (alias **caught**, **sig_catch**).

sigignore IGNORED see **ignored**. (alias **ignored**, **sig_ignore**).

sigmask BLOCKED see **blocked**. (alias **blocked**, **sig_block**).

size SIZE approximate amount of swap space that would be required if the process were to dirty all writable pages and then be swapped out.
This number is very rough!

slice SLICE displays the slice unit which a process belongs to, if systemd support has been included.

spid SPID see **lwp**. (alias **lwp**, **tid**).

stackp STACKP address of the bottom (start) of stack for the process.

start STARTED time the command started. If the process was started less than 24 hours ago, the output format is "HH:MM:SS", else it is "Mmm dd" (where Mmm is a three-letter month

name). See also **lstart**, **bsdstart**, **start_time**, and **stime**.

start_time START starting time or date of the process. Only the year will be displayed if the process was not started the same year **ps** was invoked, or "MmmDD" if it was not started the

same day, or "HH:MM" otherwise. See also **bsdstart**, **start**, **lstart**, and **stime**.

stat STAT multi-character process state. See section **PROCESS STATE CODES** for the different values meaning. See also **s** and **state** if you just want the first character displayed.

state S see **s**. (alias **s**).

suid SUID saved user ID. (alias **svuid**).

supgid SUPGID group ids of supplementary groups, if any. See **getgroups(2)**.

supgrp SUPGRP group names of supplementary groups, if any. See **getgroups(2)**.

suser SUSER saved user name. This will be the textual user ID, if it can be obtained and the field width permits, or a decimal representation otherwise. (alias **svuser**).

svgid SVGID see **sgid**. (alias **sgid**). **svuid** SVUID see **suid**. (alias **suid**).

sz SZ size in physical pages of the core image of the process. This includes text, data, and stack space. Device mappings are currently excluded; this is subject to change. See **vsz** and **rss**.

tgid TGID a number representing the thread group to which a task belongs (alias **pid**). It is the process ID of the thread group leader.

thcount THCNT see **nlwp**. (alias **nlwp**). number of kernel threads owned by the process.

tid TID the unique number representing a dispatchable entity (alias **lwp**, **spid**). This value may also appear as: a process ID (pid); a process group ID (pgpr); a session ID for the session leader (sid); a thread group ID for the thread group leader (tgid); and a tty process group ID for the process group leader (tpgid).

time TIME cumulative CPU time, "[DD-]HH:MM:SS" format. (alias **cputime**).

times TIME cumulative CPU time in seconds (alias **cputimes**).

tname TTY controlling tty (terminal). (alias **tt**, **tty**).

tpgid TPGID ID of the foreground process group on the tty (terminal) that the process is connected to, or -1 if the process is not connected to a tty.

trs TRS text resident set size, the amount of physical memory devoted to executable code.

tt TT controlling tty (terminal). (alias **tname**, **tty**).

tty TT controlling tty (terminal). (alias **tname**, **tt**).

ucmd CMD see **comm**. (alias **comm**, **ucomm**).

ucomm COMMAND see **comm**. (alias **comm**, **ucmd**).

uid UID see **euid**. (alias **euid**).

uname USER see **euser**. (alias **euser**, **user**).

unit UNIT displays unit which a process belongs to, if systemd support has been included.

user USER see **euser**. (alias **euser**, **uname**).

usersns USERSNS Unique inode number describing the namespace the process belongs to. See **namespaces(7)**.

utsns UTSNS Unique inode number describing the namespace the process belongs to. See **namespaces(7)**.

uunit UUNIT displays user unit which a process belongs to, if systemd support has been included.

vsz VSZ see **vsz**. (alias **vsz**).

vsz VSZ virtual memory size of the process in KiB (1024-byte units). Device mappings are currently excluded; this is subject to change. (alias **vsz**).

wchan WCHAN name of the kernel function in which the process is sleeping, a "-" if the process is running, or a "*" if the process is multi-threaded and **ps** is not displaying threads.

ENVIRONMENT VARIABLES

The following environment variables could affect **ps**:

COLUMNS Override default display width.

LINES Override default display height.

PS_PERSONALITY Set to one of posix, old, linux, bsd, sun, digital... (see section **PERSONALITY** below).

CMD_ENV Set to one of posix, old, linux, bsd, sun, digital... (see section **PERSONALITY** below).

I_WANT_A_BROKEN_PS Force obsolete command line interpretation.

LC_TIME Date format.

PS_FORMAT

Default output format override. You may set this to a format string of the type used for the **-o** option. The **DefSysV** and **DefBSD** values are particularly useful.