# APPENDIX A

# Command-Line Reference

Throughout *Mac OS X Panther Unleashed*, you learned a number of different commands that can be used to interact with your system. Literally hundreds of shell commands and utilities can be used with the Mac OS X distribution, and, unfortunately, there simply isn't enough space to provide information on them all. This appendix provides an alphabetical reference to some of the commonly used commands on Mac OS X.

If you cannot find what you're looking for here, a more complete PDF reference guide is available at `http://www.macosxunleashed.com/downloads/clref.pdf` for your printing pleasure.

## IN THIS APPENDIX

## apropos

| | |
|---|---|
| `apropos` | Displays a list of manual pages by keyword lookup. |
| `apropos <keyword1> <keyword2> ...` | Looks up commands with any `<keywords>` in their description. |

## at, atq, atrm, batch

| | |
|---|---|
| `at` | Executes commands at a specified time. |
| `atq` | Lists the user's pending jobs, unless the user is super user. If the user is super user, lists all users' jobs. |
| `atrm` | Deletes jobs. |
| `batch` | Executes commands as soon as system load levels permit. This is either when the average load drops to below 1.5 or the value specified at the invocation of `atrun`. |

Using any of these commands requires the configuration of the `atrun` command in root's crontab. Read the `atrun` man page (following) for setup information.

```
at [-q <queue>] [-f <file>] [-m] <time>
atq [-q <queue>] [-v]
atrm [-q <queue>] <job> [<job2>...]
batch [-f <file>] [-m]
```

Both `at` and `batch` take input from either standard input or the file specified by `-f` option. The working directory, environment (except for variables `TERM`, `TERMCAP`, `DISPLAY`, and `_`), and `umask` are retained from the time of invocation. Any `at` or `batch` command invoked from an `su` shell retains the current user ID.

Permission to use these commands depends on the files `/var/at/at.allow` and `/var/at/at.deny`. The super user may use these commands. If `/var/at/at.allow` exits, only the users (one per line) listed in the file may use these commands. If `/var/at/at.allow` does not exist, `/var/at/at.deny` is checked. Only users listed in `/var/at/at.deny` may not use these commands. If an empty `/var/at/at.deny` exists, all users may use these commands. If neither file exists, only the super user may use these commands.

| | |
|---|---|
| `-q <queue>` | Uses the specified queue. A queue consists of a single letter. Valid queue ranges are `a` to `l`. The `a` queue is the default, and `b` is the batch queue. Queues with higher letters run with increased niceness. If `atq` is given a specific queue, it shows only the pending jobs in the specified queue. |
| `-f <file>` | Executes commands in the specified `<file>` rather than from standard input. |
| `-m` | Sends mail to the user when the job is complete, whether or not there was any output. |
| `-v` | For `atq`, shows completed, but not yet deleted, jobs in the queue. Otherwise, shows the time the job will be executed. |

*<time>*

*<time>* may be given in a variety of formats. Times may be of the form *<HHMM>* or *<HH:MM>* for a specific time of day. If the time has already passed, the next day is assumed. You may also specify midnight, noon, or teatime (teatime for 4:00 p.m.). You may also append AM, am, PM, or pm to a specific time. A time may also include a date in any of the following forms: *<month-name>* *<day>* [*<year>*] or MMDDYY or MM/DD/YY or DD.MM.YY. The date must follow the time specification. Time may also be given in increments, such as *<now>* + *<count><time_units>*, where *<time_units>* can be minutes, hours, days, or weeks. Terms today and tomorrow may also be used.

## bzip2, bunzip2, bzcat, bzip2recover

| | |
|---|---|
| `bzip2, bunzip2` | Block-sorting file compressor, v1.0.2 |
| `bzcat` | Decompresses files to stdout |
| `bzip2recover` | Recovers data from damaged bzip2 files |

```
bzip2 [-cdfkqstvzVL123456789 ] [<filename1> <filename2> .. ]

bunzip2 [-fkvsVL] [<filename1> <filename2> ...]

bzcat [-s] [<filename1> <filename2> ...]

bzip2recover <filename>
```

`bzip2`, `bunzip2` and `bzcat` are really the same program. The decision about what actions to take is done on the basis of which name is used.

`bzip2` compresses files using the Burrows-Wheeler block sorting text compression algorithm, and Huffman coding.

`bzip2` expects a list of filenames to accompany the command-line flags. Each file is replaced by a compressed version of itself, with the name `<original_name>.bz2`. Each compressed file has the same modification date, permissions, and, when possible, ownership as the corresponding original so that these properties can be correctly restored at decompression time.

If no filenames are specified, `bzip2` compresses from standard input to standard output.

`bzip2` reads arguments from the environment variables `BZIP2` and `BZIP`, in that order, and processes them before reading any arguments from the command line. Compression is always performed even if the compressed file is slightly larger than the original.

`bunzip2` (or `bzip2 -d`) decompresses files. Files not created by `bzip2` are detected and ignored, and a warning is issued. Filenames are restored as follows:

| | |
|---|---|
| `<filename>.bz2` | <filename> |
| `<filename>.bz` | <filename> |
| `<filename>.tbz2` | <filename>.tar |
| `<filename>.tbz` | <filename>.tar |
| `<anyothername>` | <anyothername>.out |

Supplying no filenames causes decompression from standard input to standard output.

`bzcat` (or `bzip2 -dc`) decompresses all specified files to standard output.

`bzip2recover` is a simple program whose purpose is to search for blocks in .bz2 files and write each block out into its own .bz2 file. You can then use bzip2 -t to test the integrity of the resulting files and decompress those that are undamaged.

`bzip2recover` takes a single argument, the name of the damaged file, and writes a number of files `rec00001file.bz2`, `rec00002file.bz2`, and so on, containing the extracted blocks. The output filenames are designed so that the use of wildcards in subsequent processing—for example, `bzip2 -dc rec*file.bz2 > recovered_data`—processes the files in the correct order.

`-h`
`--help`

Displays a help menu.

`-c`
`--stdout`

Compresses or decompresses to standard output.

`-d`
`--decompress`

Forces decompression.

`-f`
`--force`

Forces overwrite of output files. Normally, bzip2 does not overwrite existing output files. Also forces bzip2 to break hard links to files, which it otherwise doesn't do.

bzip2 normally declines to decompress files that don't have the correct magic header bytes. If forced (`-f`), however, it will pass such files through unmodified. This is how GNU gzip behaves.

`-k`
`--keep`

Keeps (doesn't delete) input files during compression or decompression.

`-q`
`--quiet`

Suppresses non-essential warning messages. Messages pertaining to I/O errors and other critical events are not suppressed.

`-s`
`--small`

Reduces memory usage, for compression, decompression and testing. If your machine is low on memory (8 megabytes or less), use `-s` for everything.

`-t`
`--test`

Checks integrity of the specified file(s), but doesn't decompress them. This really performs a trial decompression and throws away the result.

`-v`
`--verbose`

Verbose mode. Shows the compression ratio for each file processed. Further -vs increase the verbosity level.

`-z`
`--compress`

Forces compression, regardless of the invocation name.

`-V`
`--version`

Displays the software version, license terms, and conditions.

`-L`
`--license`

Displays the software version, license terms, and conditions.

-1 (or –fast) . –9 (or –best)

Sets block size to 100k . 900k. The --fast and –best aliases are primarily for GNU gzip compatibility. In particular, --fast doesn't make things significantly faster. --best merely selects the default behavior.

`--`

Treats all subsequent arguments as file names, even if they start with a dash. This is so you can handle files with names beginning with a dash—for example, `bzip2 -- -myfilename`.

# cancel

`cancel` Removes print jobs from the queue

`cancel [ -a ] [ -h <server> ] [ <id> ] [ <destination> ] [ <destination-id> ]`

`-a` Removes all jobs from the specified destination

`-h <server>` Specifies the print server hostname. The default is `localhost` or the value of the `CUPS_SERVER` environment variable.

# cat

`cat` Concatenates and prints files.

`cat [-nbsvetu] <file1> <file2> ...`

`cat [-nbsvetu] [-]`

`cat` reads files in sequential, command-line order and writes them to standard output. A single dash represents standard input.

`-n` Numbers all output lines.

`-b` Numbers all output lines, except `b` or blank lines.

`-s` Squeezes multiple adjacent empty lines, causing single-spaced output.

`-v` Displays nonprinting characters. Control characters print as `^X` for Control+X; delete (octal 0177) prints as `^?`; non-ASCII characters with the high bit set are printed as `M-` (for meta) followed by the character for the low 7 bits.

`-e` Implies `-v` option. Displays a dollar sign (`$`) at the end of each line as well.

`-t` Implies `-v` option. Displays tab characters as `^I` as well.

`-u` Guarantees unbuffered output.

# cd

`cd` Changes working directory

`cd [-p] [-l] [-n ¦ -v] [<directory>]`

`cd`

*`<directory>`* is an absolute or relative pathname. The interpretation of the relative pathname depends on the `CDPATH` environment variable.

`-p` Prints the final directory stack

`-l` Expands `~` or `~<name>` to the pathname of the user's home directory. Prints output in long form.

`-n` Wraps entries before they reach the end of the screen.

`-v` Prints one entry per line, preceded by their stack position. If both `-n` and `-v` are specified, `-v` takes precedence.

The following environment variables affect the execution of `cd`:

`HOME` If `cd` is invoked without any arguments and the `$HOME` exists, `$HOME` becomes the new working directory.

`CDPATH` If <*directory*> does not begin with `/`, `.`, or `..`, `cd` searches for the directory relative to each directory named in `CDPATH` variable, in the order listed. If the new working directory is derived from `$CDPATH`, it is printed to standard output.

## chflags

`chflags` Changes file flags.

```
chflags [-R [-H ¦ -L ¦ -P]] <flags> <file1> <file2> ...
```

`-R` Recursively descends through directory arguments to change file flags.

`-H` If `-R` is specified, symbolic links on the command line are followed. Symbolic links encountered in tree traversal are not followed.

`-L` If `-R` is specified, all symbolic links are followed.

`-P` If `-R` is specified, no symbolic links are followed.

Symbolic links do not have flags. Unless `-H` or `-L` is specified, `chflags` on a symbolic link always succeeds and has no effect. `-H`, `-L`, and `-P` options are ignored unless `-R` is specified. Furthermore, `-H`, `-L`, and `-P` override each other. The last option specified determines the action that is taken.

<*flags*> is a comma-separated list of keywords. Currently available keywords are as follows:

| | |
|---|---|
| `arch` | Sets the `archived` flag (super user only) |
| `opaque` | Sets the `opaque` flag (owner or super user only) |
| `nodump` | Sets the `nodump` flag (owner or super user only) |
| `sappnd` | Sets the system `append-only` flag (super user only) |
| `schg` | Sets the system `immutable` flag (super user only) |
| `uappnd` | Sets the user `append-only` flag (owner or super user only) |
| `uchg` | Sets the user `immutable` flag (owner or super-user only) |

Prepending the letters `no` to a flag turns the flag off.

## chgrp

`chgrp` Changes group.

```
chgrp [-R [-H ¦ -L ¦ -P]] [-fh] <group> <file1> <file2> ...
```

`-R` Recursively descends through directory arguments to change the group ID.

`-H` If `-R` is specified, symbolic links on the command line are followed. Symbolic links encountered in tree traversal are not followed.

`-L` If `-R` is specified, all symbolic links are followed.

`-P` If `-R` is specified, no symbolic links are followed.

`-f` Forces an attempt to change group ID without reporting any errors.

`-h` If the file is a symbolic link, the group ID of the link is changed.

Unless `-h`, `-H`, or `-L` is specified, `chgrp` on symbolic links always succeeds and has no effect.

The `-H`, `-L`, and `-P` options are ignored unless `-R` is specified. Because they also override each other, the last one specified determines the action that is taken.

The group may be either a numeric group ID or a group name. If a group name exists for a group ID, the associated group name is used for the group.

The user invoking `chgrp` must belong to the specified group and be the owner of the file, or be the super user.

Unless invoked by the super user, `chgrp` clears the `set-user-id` and `set-group-id` bits.

## chmod

`chmod` Changes file modes.

`chmod [-R [-H ¦ -L ¦ -P]] [-h]` <*absolute_mode*> `<file1>` <*file2*> ...

`chmod [-R [-H ¦ -L ¦ -P]] [-h] <symbolic_mode> <file1> <file2> ...`

`-R` Recursively descends through directory arguments to change file modes.

`-H` If `-R` is specified, symbolic links on the command line are followed. Symbolic links encountered in tree traversal are not followed.

`-L` If `-R` is specified, all symbolic links are followed.

`-P` If `-R` is specified, no symbolic links are followed.

Unless `-H` or `-L` is specified, `chmod` on a symbolic link always succeeds and has no effect. The `-H`, `-L`, and `-P` options are ignored unless `-R` is specified. Furthermore, `-H`, `-L`, and `-P` override each other. The last option specified determines the action that is taken.

Permissions are described by three sequences of letters in the order listed here. Each sequence describes the permissions for user, group, and other. If a certain permission has not been granted, a `-` (dash) appears in its place.

| User | Group | Other |
|---|---|---|
| `rwx` | `rwx` | `rwx` |

The permissions on a file can be viewed using `ls -l` and changed using `chmod`.

## Absolute Mode

Absolute mode is constructed by `OR`ing any of the following modes:

| | |
|---|---|
| `4000` | Sets user ID on execution—If this is a program, causes it to run as though the user who owns it is actually running it, regardless of who executes it. |
| `2000` | Sets group ID on execution—If this is a program, causes it to run with the group ID of the program, regardless of group memberships of the user who executes it. |
| `1000` | Turns on sticky bit—Has different meanings in different contexts: for directories, protects files from modification by other than owner, even if the user has permissions to write in the directory—overridden by directory ownership. |
| `0400` | Allows read by owner. |
| `0200` | Allows write by owner. |
| `0100` | Allows execute (search in a directory) by owner. |
| `0600` | Allows read, write by owner. |
| `0500` | Allows read, execute by owner. |
| `0300` | Allows write, execute by owner. |
| `0700` | Allows read, write, execute by owner. |
| `0040` | Allows read by group. |
| `0020` | Allows write by group. |
| `0010` | Allows execute (search in a directory) by group. |
| `0060` | Allows read, write by group. |
| `0050` | Allows read, execute by group. |
| `0030` | Allows write, execute by group. |
| `0070` | Allows read, write, execute by group. |
| `0004` | Allows read by other. |
| `0002` | Allows write by other. |
| `0001` | Allows execute (search in a directory) by other. |
| `0006` | Allows read, write by other. |
| `0005` | Allows read, execute by other. |
| `0003` | Allows write, execute by other. |
| `0007` | Allows read,  write, execute by other. |

## Symbolic Mode

Symbolic mode is a comma-separated list, with no intervening white space, of the form:

`[<`*`who`*`>]<`*`operator`*`>[<`*`permissions`*`>]`

<*who*> has the following form:

`< u ¦ g ¦ o ¦ a>`

| | |
|---|---|
| `u` | User’s permissions |
| `g` | Group’s permissions |
| `o` | Other’s permissions |
| `a` | All permissions (user, group, other); equivalent to `ugo` |

<*operator*> has the following form:

`< + ¦ - ¦ =>`

| | |
|---|---|
| `+` | Adds <*permissions*>. |

If <*permissions*> is not specified, no changes occur.

If <*who*> is not specified, <*who*> defaults to `a`, and <*permissions*> are added as specified, except that `chmod` does not override the file mode creation mask.

If <*who*> is specified, <*permissions*> are added as specified.

| | |
|---|---|
| `-` | Removes <*permissions*>. |

If <*permissions*> is not specified, no changes occur.

If <*who*> is not specified, <*who*> defaults to `a`, and <*permissions*> are removed as specified, except that `chmod` does not override the file mode creation mask.

If <*who*> is specified, <*permissions*> are removed as specified.

| | |
|---|---|
| `=` | Assigns the absolute <*permissions*> specified. |

If <*who*> is not specified, <*who*> defaults to `a`.

If <*permissions*> is not specified, <*permissions*> defaults to `remove`.

If <`who`> is specified and <*permissions*> is not, all permissions for <*who*> are removed.

If <*who*> is not specified and <*permissions*> is specified, <*permissions*> for all are set to <*permissions*>, except that `chmod` does not override the file creation mask.

If <*who*> is specified and <*permissions*> is specified, <*permissions*> for <*who*> are set as specified.

<*permissions*> has the following form:

`<r ¦ w ¦ x ¦ X ¦ s ¦ t ¦ u ¦ g ¦ o>`

| | |
|---|---|
| `r` | Sets read bits. |
| `w` | Sets write bits. |
| `x` | Sets execute/search bits. |
| `X` | Sets execute/search bits if the file is a directory, or if any execution/search bits are already set in the file before `X` would act upon the file. `X` is used only with `+`, and is ignored in all other cases. |
| `s` | Sets the `set-user-ID-on-execution` and `set-group-ID-on-execution` bits. A process runs as the user or group specified by `s`. |

| | |
|---|---|
| `t` | Sets the sticky bit. |
| `u` | User permission bit in the mode of the original file. |
| `g` | Group permission bits in the mode of the original file. |
| `o` | Other permission bits in the mode of the original file. |

Operations on `<who>` `o` in combination with `<permissions>` `s` or `t` are ignored.

## chown

| | |
|---|---|
| `chown` | Changes file owner and group. |

```
chown [-R [-H ¦ -L ¦ -P]] [-fh] <owner> <file1> <file2> ...
chown [-R [-H ¦ -L ¦ -P]] [-fh] :<group> <file1> <file2> ...
chown [-R [-H ¦ -L ¦ -P]] [-fh] <owner>:<group> <file1> <file2> ...
```

| | |
|---|---|
| `-R` | Recursively descends through directory arguments to change the user ID and/or group ID. |
| `-H` | If `-R` is specified, symbolic links on the command line are followed. Symbolic links encountered in tree traversal are not followed. |
| `-L` | If `-R` is specified, all symbolic links are followed. |
| `-P` | If `-R` is specified, no symbolic links are followed. |
| `-f` | Forces an attempt to change user ID and/or group ID without reporting any errors. |
| `-h` | If the file is a symbolic link, the user ID and/or group ID of the link is changed. |

The `-H`, `-L`, and `-P` options are ignored unless `-R` is specified. Because they also override each other, the last option specified determines the action taken.

The `-L` option cannot be used with the `-h` option.

It is not necessary to provide both `<owner>` and `<group>`; however, one must be specified. If group is specified, it must be preceded with a colon (`:`).

The owner may be either a numeric user ID or a username. If a username exists for a numeric user ID, the associated username is used as for the owner. Similarly, the group may be either a numeric group ID or a group name. If a group name exists for a group ID, the associated group name is used for the group.

Unless invoked by the super user, `chown` clears set-user-id and set-group-id bits.

## cp

| | |
|---|---|
| `cp` | Copies files. |

```
cp [-R (-H ¦ -L ¦ -P)] [-f ¦ -i] [-pv] <source> <target>
cp [-R (-H ¦ -L ¦ -P)] [-f ¦ -i] [-pv] <source1> <source2> .. <directory>
```

In its first form, `cp` copies the contents of `<source>` to `<target>`.

In its second form, cp copies the contents of the list enumerated by <*source1*> <*source2*> .. to the directory named by <*directory*>. The names of the files themselves are not changed. If cp detects an attempt to copy to itself, that attempt fails.

-R If <*source*> is a directory, cp recursively copies the directory. This option also causes symbolic links to be copied rather than indirected through. Created directories have the same mode as the corresponding source directory.

-H If -R is specified, symbolic links on the command line are followed, but symbolic links in the tree traversal are not.

-L If -R is specified, all symbolic links are followed.

-P If -R is specified, no symbolic links are followed.

-f Forces an existing file to be overwritten. If permissions do not allow the copy to succeed, this forces the existing file to be removed and a new file to be created, without prompting for confirmation. The -i option is ignored if the -f option is specified.

-i Invokes an interactive mode that prompts for a confirmation before overwriting an existing file.

-p Causes cp to retain as much of the modification time, access time, file flags, file mode, user ID, and group ID information as permissions allow.

-v Verbose copy.

# crontab

crontab Maintains crontab files for individual users.

crontab [-u <*user*>] <*file*>

crontab [-u <*user*>] [-l ¦ -r ¦ -e]

crontab is the program that installs, removes, or lists the tables the cron (8) executes for users. Each user can have his own crontab, which is stored in /var/cron/tabs/. The crontab is not edited directly.

If /var/cron/allow exists, the <*user*> must be listed in the file to be able to use cron. If /var/cron/allow does not exist, but /var/cron/deny exists, <*user*> must not be listed in this file to use this command. If neither file exists (depending on site-dependent configuration), either only the super user may use this command or all users may be able to use this command.

The first form of the command installs a crontab from <*file*> or standard input, if - is given instead of <*file*>. The second form of the command displays, removes, or edits the installed crontab.

-u <user> Specifies the name of the user. If not specified, the user issuing the command is assumed. If crontab is being used inside an su command, -u should be used.

-l Lists the current crontab on standard output.

-r Removes the current crontab.

-e Edits the current crontab using the editor specified by the environment variables VISUAL or EDITOR. On exiting the editor, the modified crontab is automatically installed.

Basic format of a `crontab` statement, with value ranges shown here:

`minute hour day_of_month month day_of_week [<user>] <command>`

`0-59 0-23 1-31 1-12 0-7 (Sunday may be 0 or 7)`

Fields may be separated by spaces or tabs. `*` may be used as the value of a field to mean all possible values for that field. A field value may be further specified by providing a single value, a comma-separated list of values, a range of values, or a comma-separated list of single values or ranges of values.

Step values may be specified by use of *<range>/<number>*. For example, `0-23/2` would be every other hour. `0-23/2` is equivalent to the value list 0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22. Step values may also be specified by `*/`*<number>*. For example, every other hour could also be specified by `*/2` in that field.

Names may also be used for the month and day_of_week fields. Names are the first three characters of the actual name. Case does not matter. Lists or ranges of names, however, may not be used.

The *<user>* field is specified only in a system `crontab`.

# curl

`curl`

A utility for getting a URL with FTP, TELNET, LDAP, GOPHER, DICT, FILE, HTTP or HTTPS syntax.

`curl [<options>] [<URL>...]`

`-a`
`--append`

(FTP) When used in a `ftp` upload, this tells `curl` to append to the target file instead of overwriting it. If the file doesn't exist, it is created.

If this option is used twice, the second one disables append mode again.

`-A <agent string>`
`--user-agent`
`<agent string>`

(HTTP) Specifies the User-Agent string (to send to the HTTP server. Some badly done CGIs fail if it's not set to "Mozilla/4.0". To encode blanks in the string, surround the string with single quote marks. This can also be set with the `-H/--header` flag, of course.

If this option is used more than once, the last one is the one to be used.

`-b <name=data>`
`--cookie <name=data>`

(HTTP) Passes the data to the HTTP server as a cookie. The data is supposedly the data previously received from the server in a `Set-Cookie:` line. The data should be in the format `NAME1=VALUE1; NAME2=VALUE2`, but there's nothing to say you can't change it.

If no = is used in the line, it is treated as a filename to use to read previously stored cookie lines from, which should be used in this session if they match. Using this method also activates the cookie parser, which makes `curl` record incoming cookies too, which may be handy for using this in combination with the `-L/--location` option. The file format of the file to read cookies from should be plain HTTP headers or the Netscape cookie file format.

Note that the file specified with `-b/--cookie` is only used as input. No cookies are stored in the file. To store cookies, save the HTTP headers to a file using `-D/--dump-header`.

If this option is used more than once, the last one is the one to be used.

`-B`
`--use-ascii`

Uses ASCII transfer when getting an FTP file or LDAP info. For FTP, this can also be enforced by using an URL that ends with `;type=A`.

If this option is used twice, the second use disables ASCII usage.

`-C <offset>`
`--continue-at <offset>`

Continues/resumes a previous file transfer at the given offset. The given offset is the exact number of bytes that are skipped counted from the beginning of the source file before it is transferred to the destination. If used with uploads, the `ftp` server command `SIZE` is not be used by `curl`. Upload resume is for FTP only. HTTP resume is only possible with HTTP/1.1 or later servers.

If this option is used several times, the last one is used.

`-d <data>`
`--data <data>`

(HTTP) Sends the specified data in a `POST` request to the HTTP server in a way that can emulate as if a user has filled in a HTML form and pressed the submit button. Note that the data is sent exactly as specified with no extra processing (with all newlines cut off). The data is expected to be url-encoded. This causes `curl` to pass the data to the server using the content-type `application/x-www-form-urlencoded`. Compare to `-F`. If more than one `-d/--data` option is used on the same command line, the data pieces specified are merged together with a separating `&` character. Thus, `using-d name=daniel -d skill=lousy` generates a post chunk that looks like `name=daniel&skill=lousy`.

If this option is used several times, the ones following the first append data.

`--data-ascii <data>`

(HTTP) An alias for the `-d/--data` option.

If this option is used several times, the ones following the first append data.

`--data-binary <data>`

(HTTP) Posts data in a similar manner as `--data-ascii` does, although when using this option the entire context of the posted data is kept as is. If you want to post a binary file without the strip-newlines feature of the `--data-ascii` option, this is for you.

`-D <file>`
`--dump-header <file>`

(HTTP/FTP) Write the HTTP headers to this `<file>`. Write the FTP file info to this `<file>` if `-I/--head` is used.

Handy for storing the cookies that a HTTP site sends to you. The cookies could then be read in a second curl invoke by using the `-b/--cookie` option.

If this option is used several times, the last one is used.

`-e <URL>`
`--referer <URL>`

(HTTP) Sends the Referer Page information to the HTTP server. This can also be set with the `-H/--header` flag, of course. When used with `-L/--location`, you can append`;auto` to the referer URL to make `curl` automatically set the previous URL when it follows a `Location:` header. The `;auto` string can be used alone, even if you don't set an initial referer.

If this option is used several times, the last one is used.

`--egd-file <file>`

(HTTPS) Specifies the pathname to the Entropy Gathering Daemon socket. The socket is used to seed the random engine for SSL connections. See also the `--random-file` option.

`-E`
`--cert <certificate[:password]>`

(HTTPS) Tells `curl` to use the specified certificate file when getting a file with HTTPS. The certificate must be in PEM format. If the optional password isn't specified, it will be queried for on the terminal. Note that this certificate is the private key and the private certificate concatenated!

If this option is used several times, the last one is used.

`--cacert <CA certificate>`

(HTTPS) Tells `curl` to use the specified certificate file to verify the peer. The certificate must be in PEM format.

If this option is used several times, the last one is used.

`-f`
`--fail`

(HTTP) Fails silently (no output at all) on server errors. This is mostly done like this to better enable scripts, and so on to better deal with failed attempts. In normal cases when a HTTP server fails to deliver a document, it returns a HTML document stating so (which often also explains the failure). This flag will prevent `curl` from outputting that and fail silently instead.

If this option is used twice, the second again disables silent failure.

`-F`
`--form <name=content>`

(HTTP) This lets `curl` emulate a filled in form in which a user has pressed the submit button. This causes `curl` to `POST` data using the content-type `multipart/form-data` according to RFC1867. This enables uploading of binary files, and so on. To force the content part to be a file, prefix the filename with an `@` sign. To just get the content part from a file, prefix the filename with the letter <. The difference between `@` and < is that `@` makes a file get attached in the post as a file upload, whereas the < makes a text field and just gets the contents for that text field from a file.

This option can be used multiple times.

`-g`
`--globoff`

Switches off the URL globbing parser. When you set this option, you can specify URLs that contain the letters `{}[]` without having them being interpreted by `curl` itself. Note that these letters are not normal legal URL contents, but they should be encoded according to the URI standard.

`-h`
`--help`

Displays help.

`-H`
`--header <header>`

(HTTP) Extra header to use when getting a Web page. You may specify any number of extra headers. Note that if you should add a custom header that has the same name as one of the internal ones `curl` would use, your externally set header will be used instead of the internal one. This allows you to make even trickier stuff than `curl` would normally do. You should not replace internally set headers without knowing perfectly well what you're doing. Replacing an internal header with one without content on the right side of the colon will prevent that header from appearing.

This option can be used multiple times.

`-i`
`--include`

(HTTP) Includes the HTTP header in the output. The HTTP header includes things such as server name, date of the document, HTTP version and more.

If this option is used twice, the second again disables the header include.

`--interface <name>`

Performs an operation using a specified interface. You can enter interface name, IP address or hostname. An example could look like

```
curl --interface eth0:1 http://www.netscape.com/
```

If this option is used several times, the last one is used.

`-H`
`--head`

HTTP/FTP) Fetches the HTTP header only! HTTP servers feature the command `HEAD` that this uses to get nothing but the header of a document. When used on a FTP file, `curl` displays the file size only.

If this option is used twice, the second again disables header only.

`--krb4 <level>`

(FTP) Enables kerberos4 authentication and use. The level must be entered and should be one of `clear`, `safe`, `confidential`, or `private`. Should you use a level that is not one of these, `private` is used instead.

If this option is used several times, the last one is used

`-K`
`--config <config file>`

Specifies which config file to read `curl` arguments from. The config file is a text file in which command-line arguments can be written, which then will be used as if they were written on the actual command line. Options and their parameters must be specified on the same config file line. If the parameter is to contain whitespaces, the parameter must be enclosed within quotes. If the first column of a config line is a `#` character, the rest of the line will be treated as a comment.

Specifies the filename as `-` to make `curl` read the file from `STDIN`.

This option can be used multiple times.

`-l`
`--list-only`

(FTP) When listing an FTP directory, this switch forces a name-only view. Especially useful if you want to machine-parse the contents of an FTP directory because the normal directory view doesn't use a standard look or format.

If this option is used twice, the second again disables list only.

`-L`
`--location`

(HTTP/HTTPS) If the server reports that the requested page has a different location (indicated with the header line Location:), this flag instructs `curl` to reattempt the get on the new location. If used together with `-i` or `-I`, headers from all requested pages are shown. If this flag is used when making a `HTTP POST`, `curl` automatically switches to `GET` after the initial `POST` is done.

If this option is used twice, the second again disables location following.

`-m`
`--max-time <seconds>`

Maximum time in seconds that you allow the whole operation to take. This is useful for preventing your batch jobs from hanging for hours due to slow networks or links going down. See also the `--connect-timeout` option.

If this option is used several times, the last one is used.

`-M`
`--manual`

Manual. Displays the `curl` man page.

`-n`
`--netrc`

Makes `curl` scan the `.netrc` file in the user's home directory for login name and password. This is typically used for `ftp` on Unix. If used with http, `curl` will enable user authentication. See `netrc`(4) for details on the file format. `curl` will not complain if that file doesn't have the right permissions. (It should not be world nor group readable.) The environment variable `HOME` is used to find the home directory.

If this option is used twice, the second again disables netrc usage.

`-N`
`--no-buffer`

Disables the buffering of the output stream. In normal work situations, `curl` uses a standard buffered output stream with the effect that it will output the data in chunks, not necessarily exactly when the data arrives. Using this option disables that buffering.

If this option is used twice, the second again switches on buffering.

`-o`
`--output <file>`

Writes output to `<file>` instead of stdout. If you are using `{}` or `[]` to fetch multiple documents, you can use # followed by a number in the `<file>` specifier. That variable is replaced with the current string for the URL being fetched.

`-O`
`--remote-name`

Writes output to a local file named like the remote file we get. (Only the file part of the remote file is used; the path is cut off.)

You may use this option as many times as you have number of URLs.

`-p`
`--proxytunnel`

When an HTTP proxy is used, this option causes non-HTTP protocols to attempt to tunnel through the proxy instead of merely using it to do HTTP-like operations. The tunnel approach is made with the HTTP proxy CONNECT request and requires that the proxy allows direct connect to the remote port number curl wants to tunnel through to.

If this option is used twice, the second again disables proxy tunnel.

`-P`
`--ftpport <address>`

(FTP) Reverses the initiator/listener roles when connecting with `ftp`. This switch makes Curl use the `PORT` command instead of `PASV`. In practice, `PORT` tells the server to connect to the client's specified address and port, whereas `PASV` asks the server for an IP address and port to connect to. `<address>` should be one of

- interface—that is, "`eth0`" to specify which interface's IP address you want to use (Unix only)
- IP address—that is, "`192.168.10.1`" to specify exact IP number
- hostname—that is, "`my.host.domain`" to specify machine
- `-` (any single-letter string) to make it pick the machine's default

If this option is used several times, the last one is used.

`-q`

If used as the first parameter on the command line, the `$HOME/.curlrc` file will not be read and used as a config file.

`-Q`
`--quote <comand>`

(FTP) Sends an arbitrary command to the remote FTP server by using the `QUOTE` command of the server. Not all servers support this command, and the set of `QUOTE` commands are server specific! `QUOTE` commands are sent *before* the transfer is taking place. To make commands take place after a successful transfer, prefix them with a dash `-`. You may specify any amount of commands to be run before and after the transfer. If the server returns failure for one of the commands, the entire operation is aborted.

This option can be used multiple times.

`--random-file <file>`

(HTTPS) Specifies the pathname to file containing what will be considered as random data. The data is used to seed the random engine for SSL connections. See also the `--edg-file` option.

`-r`
`--range <range>`

(HTTP/FTP) Retrieves a byte range (that is, a partial document) from a HTTP/1.1 or FTP server. Ranges can be specified in a number of ways.

If this option is used several times, the last one is used.

`-s`
`--silent`

Silent mode. Doesn't show progress meter or error messages.

If this option is used twice, the second again disables mute.

`-S`
`--show-error`

When used with `-s`, it makes `curl` show an error message if it fails.

If this option is used twice, the second again disables show error.

`-t`
`--telnet-option <OPT=val>`

Passes options to the `telnet` protocol. Supported options are

`TTYPE=<term>` Sets the terminal type.

`XDISPLOC=<X display>` Sets the X11 display location.

`NEW_ENV=<var,val>` Sets an environment variable.

`-T`
`--upload-file <file>`

Transfers the specified local `<file>` to the remote server at `<URL>`. If there is no file part in the specified URL, `curl` will append the local filename. Note that you must use a trailing `/` on the last directory to really prove to `curl` that you aren't providing a filename, or `curl` will think that your last directory name is the remote filename to use. That will most likely cause the upload operation to fail. If this is used on an `http(s)` server, the PUT command will be used.

If this option is used several times, the last one is used.

`-u`
`--user <user:password>`

Specifies user and password to use when fetching. See `README.curl` for detailed examples of how to use this. If no password is specified, `curl` asks for it interactively.

If this option is used several times, the last instance is used.

`-U`
`--proxy-user <user:password>`

Specifies user and password to use for Proxy authentication. If no password is specified, `curl` asks for it interactively.

If this option is used several times, the last instance is used.

`--url <URL>`

Specifies a `URL` to fetch. This option is mostly handy when you want to specify URL(s) in a config file.

This option may be used any number of times. To control where this URL is written, use the `-o` or the `-O` options.

`-v`
`--verbose`

Makes the fetching more verbose/talkative. Mostly usable for debugging. Lines starting with > means data sent by curl, < means data received by curl that is hidden in normal cases, and lines starting with * means additional info provided by curl.

If this option is used twice, the second again disables verbose.

`-V`
`--version`

Displays the full version of `curl`, `libcurl`, and other third-party libraries linked with the executable.

`-w`
`--write-out <format>`

Defines what to display after a completed and successful operation. The format is a string that may contain plain text mixed with any number of variables. The string can be specified as `"string"`. To cause it to be read from a particular file, you specify it as `"@filename"` and to tell `curl` to read the format from `stdin`, you write `"@-"`.

If this option is used several times, the last one is used.

`-x`
`--proxy <proxyhost[:port]>`

Uses specified proxy. If the port number is not specified, it is assumed at port 1080.

If this option is used several times, the last one is used.

`-X`
`--request <command>`

(HTTP) Specifies a custom request to use when communicating with the HTTP server. The specified request is used instead of the standard GET. Read the HTTP 1.1 specification for details and explanations.

(FTP) Specifies a custom FTP command to use instead of LIST when doing file lists with ftp.

If this option is used several times, the last one is used.

`-y`
`--speed-time <time>`

If a download is slower than speed-limit bytes per second during a speed-time period, the download gets aborted. If `speed-time` is used, the default speed-limit will be 1 unless set with `-y`.

If this option is used several times, the last one is used.

`-Y`
`--speed-limit <speed>`

If a download is slower than this given speed, in bytes per second, for `speed-time` seconds, it gets aborted. `speed-time` is set with `-Y` and is `30` if not set.

If this option is used several times, the last one is used.

`-z`
`--time-cond <date expression>`

(HTTP) Requests to get a file that has been modified later than the given time and date, or one that has been modified before that time. The date expression can be all sorts of date strings, or if it doesn't match any internal ones, it tries to get the time from a given filename instead.

Start the date expression with a dash (`-`) to make it request a document older than the given date/time; default is a document that is newer than the specified date/time.

If this option is used several times, the last one is used.

`-3`
`--sslv3`

(HTTPS) Forces `curl` to use SSL version 3 when negotiating with a remote SSL server.

`-2`
`--sslv2`

(HTTPS) Forces `curl` to use SSL version 2 when negotiating with a remote SSL server.

`-#`
`--progress-bar`

Displays progress information as a progress bar instead of the default statistics.

If this option is used twice, the second again disables the progress bar.

`--crlf`

(FTP) Converts `LF` to `CRLF` in upload. Useful for MVS (OS/390).

If this option is used twice, the second again disables crlf converting.

`--stderr <file>`

Redirects all writes to `stderr` to the specified file instead. If the filename is a plain `-`, it is instead written to `stdout`. This option has no point when you're using a shell with decent redirecting capabilities.

If this option is used several times, the last value is used.

## defaults

```
defaults                                  Accesses the Mac OS X user defaults system.
defaults [currentHost ¦ -host <hostname>] read [<domain> [<key>]]
defaults [currentHost ¦ -host <hostname>] read-type <domain> <key>
defaults [currentHost ¦ -host <hostname>] write <domain> {'<plist>' ¦ <domain>
<key> [']<value>[']}
default [currentHost ¦ -host <hostname>] rename <domain> <old-key> <new_key>
defaults [currentHost ¦ -host <hostname>] delete [<domain> [<key>]]
defaults [currentHost ¦ -host <hostname>] { domains ¦ find <word> ¦ help }
```

`defaults` allows users to read, write, and delete Mac OS X user defaults from the command line. Applications use the defaults system to record user preferences and other information that must be maintained when applications aren't running, such as the default font for new documents. Because applications do access the defaults system while they are running, you should not modify the defaults of a running application.

User defaults belong to domains, which typically correspond to individual applications. Each domain has a dictionary of keys and values to represent its defaults. Keys are always strings, but values can be complex data structures made up of arrays, dictionaries, strings, and binary data. These data structures are stored as XML Property Lists.

Although all applications,  system services, and other programs have their own domains, they also share a domain called `NSGlobalDomain`. If a default is not specified in the application's domain, it uses the default listed in the `NSGlobalDomain` instead.

*<domain>* is specified as follows:

`<domain_name> ¦ -app <application_name> ¦ -globalDomain`

**SUBCOMMANDS**

`read`

Prints all the user's defaults for every domain to standard output.

`read <domain>`

Prints all the user's defaults for the specified `<domain>` to standard output.

`read <domain> <key>`

Prints the value for the default of the `<domain>` identified by `<key>`.

`write <domain >`

Writes `<value>` as the value for `<key>` in `<domain>`.

`<key> '<value>'`

`<value>` must be a property list and must be enclosed in single quotes. For example,

```
defaults write com.companyname.appname "Default
Color" '(255, 0, 0)'
```

Sets the default color in `com.companyname.appname` to the array containing 255, 0, 0 (red, green, blue components). Note that the key is in quotes because of the space in its name.

`write <domain> '<plist>'`
Overwrites the defaults information in `<domain>` with that specified in `<plist>`. `<domain>` must be a property list representation of a dictionary, and must be enclosed in single quotes. For example,

```
defaults write com.companyname.appname
'{ "Default Color" = (255, 0, 0); "Default
Font" = Helvetica; }'
```

Overwrites any previous defaults for `com.companyname.appname` and replaces them with the ones specified.

`delete <domain>`
Deletes all default information for `<domain>`.

`delete <domain> <key>`
Deletes the default named `<key>` in `<domain name>`.

`domains`
Prints the names of all domains in the user's defaults system.

`find <word>`
Searches for `<word>` in the domain names, keys, and values of the user's defaults, and prints out a list of matches.

`help`
Prints a list of possible command formats.

## OPTIONS

`-g`
When specifying a domain, `-g` can be used as a synonym for `NSGlobalDomain`.

Specifying `<value>` for preference keys:

`<value>`
Specifies `<value>` as a string value to use.

`'<value>'`
Specifies `<value>` as a string value to use.

`-string <string_value>`
Specifies `<string_value>` as the string to use.

`-data <hex_digits>`
Specifies `<hex_digits>` as the data to use.

`-int[eger] <integer_value>`
Specifies `<integer_value>` as the integer value to use.

`-bool[ean] true ¦ false ¦ yes ¦ no`
Specifies the Boolean value to use.

`-date <date_rep>`
Specifies the date value to use.

`-array`
Allows the user to specify an array as the value for the given preference key. For example,

```
defaults write <somedomain>
<preferenceKey> -array <element1>
<element2> <element3>
```

The specified array overwrites the value of the key if the key was present at the time of the write. If the key was not present, it is created with the new value.

`-array-add`
Allows the user to add new elements to the end of an array for a key that has an array as its value. Usage is the same as `-array`.

`-dict` Allows the user to add a dictionary to the defaults database for a domain. Keys and values are specified in order:

`defaults write <somedomain>`

`<preferenceKey> -dict <key1> <value1> <key2> <value2>` The specified dictionary overwrites the value of the key if the key was present at the time of the write. If the key was not present, it is created with the new value.

`-dict-add` Allows the user to add new key/value pairs to a dictionary for a key that has a dictionary as its value. Usage is the same as `-dict`. If the key was not present, it is created with the specified dictionary as its value.

Specifying a host for preferences:

`currentHost` Operations on the defaults database may apply only to a specific host. Specifying `currentHost` after the `defaults` command restricts the preferences operations to the host the user is currently logged in on, rather than any host the user may log in on.

## df

`df` Displays free disk space.

`df [-ikln] [-t <type>] [<file> ¦ <filesytem> ...]`

`-i` Includes statistics on the number of free inodes.

`-k` Reports number in kilobyte counts. Default is 512-byte block sizes.

`-l` Displays statistics only about mounted file systems with the `MNT_LOCAL` flag set. If a non-local file system is given as the argument, a warning is issued and no information is displayed.

`-n` Prints out previously obtained statistics from the file system. This option should be used if it is possible that one or more file systems are in a state such that there is a long delay before they can provide statistics.

`-t <type>` Displays information for file systems of the specified type. More than one type may be specified in a comma-separated list of the list of file system types.

If the environment variable `BLOCKSIZE` is set, and the `-k` option is not used, the block counts are displayed according to the environment variable.

## disktool

disktool Disk Arbitration Command Tool

```
disktool -[r][a][u][e][m][p][n][d][x][y][c][s][g][A][S][D][l] <deviceName>
[<options>]
```

Renames, ejects, mounts, or unmounts disks and volumes

**Information About Disks:**

`-l` Lists the disks currently known and available on the system.

**Controlling Arbitration:**

`-r` Refreshes Disk Arbitration. Causes arbitration to refresh its internal tables and look for new mounts/unmounts.

**Managing Disks:**

`-u` Unmounts a disk (for example, `disktool -u disk2`)

`-p` Unmounts a partition (for example, `disktool -p disk1s2`)

`-e` Ejects a disk (for example, `disktool -e disk2`)

`-m` Mounts a disk (for example, `disktool -m disk2`). Useful when a disk has been unmounted by hand (using `-p` or `-u` parameters)

`-a` Notifies of mount. Adds the disk to the Disk Arbitrations internal tables. Useful when you have already forced the mount and want to let applications know it (for example, `disktool -a disk1 AFPVolName AFPFlags`).

`-d` Notifies of dismount. Removes the disk from Disk Arbitration's internal tables. Useful when you have already forced the unmount and want to let applications know it (for example, `disktool -d disk1`).

**Controlling Disk Parameters:**

`-n` Renames volume. Renames the volume specified as the first argument (for example, `disktool -n disk1s2 newName`).

`-g` Gets the hfs encoding on a volume (for example, `disktool -g disk1s2`).

`-s` Sets the hfs encoding on a volume. (for example, `disktool -s disk1s2 4`).

`-A` Adopts the given device into the volinfo database.

`-D` Disowns the given device from the volinfo database.

`-S` Displays the status of the device in the volinfo database

# ditto

`ditto` Copies files and directories to a destination directory

```
ditto [-v] [-V] [-arch <arch>] [-bom <bom>] [-rsrcFork] <src ...> <dst_directory>
ditto [-v] [-V] [-arch <arch>] [-rsrcFork] <src_file ...> <dst_file>
```

In the first synopsis form, `ditto` copies one or more source files or directories to a destination directory. If the destination directory does not exist, it will be created before the first source is copied. If the destination directory already exists, the source directories are merged with the previous contents of the destination.

In the second synopsis form, `ditto` copies a file to the supplied dst_file. The parent directory for dst_file must exist; otherwise, ditto will fall back to the first synopsis form.

`ditto` overwrites existing files, symbolic links, and devices in the destination when these are copied from a source. The resulting files, links, and devices will have the same mode, owner, and group as the source items from which they are copied. `ditto` does not modify the mode, owner, or group of existing directories in the destination. Files cannot overwrite directories or vice versa.

`-v` Prints a line of output for each source directory copied.

`-V` Prints a line of output for every file, symbolic link, and device copied.

`-arch <`*`arch`*`>` Thin multi-architecture binaries ("fat binaries") to the specified architecture. If multiple `-arch` options are specified, the resulting destination file will be multi-architectural containing each of the specified architectures (if they are present in the source file). `arch` should be specified as `"ppc"`, `"i386"`, and so on.

`-bom <`*`bom`*`>` If this option is given, only files, links, devices, and directories that are present in the specified BOM file are copied.

`-rsrcFork` Preserve resource forks and HFS metadata. `ditto` will store this data in AppleDouble files on file-systems that do not support resource forks.

## du

`du` Displays disk usage statistics.

```
du [-H ¦ -L ¦ -P] [-I <mask>] [-a ¦ -s ¦ -d <depth>] [-chkrx] [<file> ...]
```

`du` displays the file system block usage for each file argument and for each directory in the file hierarchy rooted in each directory argument. If no file is specified, the block usage of the hierarchy rooted in the current directory is displayed.

`-H` Follows symbolic links on the command line. Symbolic links encountered during tree traversal are not followed.

`-L` Follows all symbolic links.

`-I <`*`mask`*`>` Ignore files/dirs matching the given mask.

`-P` Does not follow symbolic links.

`-a` Displays an entry for each file in the file hierarchy.

`-h` Human readable output—add usage unit suffixes: Byte, Kilobyte, and so on.

`-s` Displays only the grand total for the specified files.

`-d <`*`depth`*`>` Displays an entry for all files and directories depth directories deep.

`-c` Displays the grand total after all the arguments have been processed.

`-k` Displays the statistics in 1024-byte blocks. Default is 512-byte blocks.

`-r` Generates a warning message about directories that cannot be read. This is the default.

`-x` Does not traverse file system mount points.

# find

`find` Finds files.

```
find [-H ¦ -L ¦ -P] [-EXdsx] [-f <file>] <file> ... <expression>
```

`find` recursively descends the directory tree of each file listing, evaluating an *<expression>* composed of primaries and operands.

**Options**

`-E` Causes find to interpret regular expression patterns specified with `–regex` or `–iregex` as standard modern regular expressions rather than as basic regular expressions (BREs). See the `re_format`(7) manual page for a description of each format.

`-H` Causes the file information and file type returned for each symbolic link on the command line to be those of the file referenced rather than those of the link itself. If the file does not exist, the information is for the link itself. File information of symbolic links not on the command line is that of the link itself.

`-L` Causes the file information and file type returned for each symbolic link to be those of the referenced file rather than those of the link itself. If the referenced file does not exist, the information is for the link itself.

`-P` Causes the file information and file type returned for each symbolic link to be those of the link itself.

`-X` Permits `find` to be safely used with `xargs`. If a filename contains any delimiting characters used by `xargs`, an error message is displayed and the file is skipped. The delimiting characters include single quote, double quote, backslash, space, tab, and newline.

`-d` Causes a depth-first traversal of the hierarchy. In other words, directory contents are visited before the directory itself. The default is for a directory to be visited before its contents.

`-s` Causes `find` to traverse the file hierarchies in lexicographical order—that is, alphabetical order within each directory. Note: `find -s` and `find ¦ sort` may give different results.

`-x` Excludes `find` from traversing directories that have a device number different from that of the file from which the descent began.

`-h` Causes the file information and file type returned for each symbolic link to be those of the referenced file rather than those of the link itself. If the referenced file does not exist, the information returned is for the link itself.

`-f` Specifies a file hierarchy for `find` to traverse. File hierarchies may also be specified as operands immediately following the options listing.

### Primaries (Expressions)

All primaries that can take a numeric argument enable the number to be preceded by `+`, `-`, or nothing. *n* takes on the following meanings:

`+`*n* More than *n*

`-`*n* Less than *n*

`n` Exactly *n*

`-atime` *n* — True if the file was last accessed *n* days ago. Note that `find` itself will change the access time.

`-ctime` *n* — True if the file's status was changed *n* days ago.

`-mtime` *n* — True if the file was last modified *n* days ago.

`-newer` <*file*> — True if the current file has a more recent modification time than <*file*>.

`-exec` <*command*>`;` — True if <*command*> returns a zero-value exit status. Optional arguments may be passed to <*command*>. The expression must be terminated by a semicolon. If `{}` appear anywhere in the command name or arguments, they are replaced by the current pathname.

`-follow` — Follows symbolic links.

`-fstype` — True if the file is contained in a file system specified by `-fstype`. Issues the command `sysctl vfs` to determine the available types of file systems on the system. There are also two pseudo types: `local` and `rdonly`. `local` matches any file system physically mounted on the system where `find` is being executed; `rdonly` matches any mounted read-only file system.

`-group` <*gname*> — True if the file belongs to the specified group name. If <*gname*> is numeric and there is no such group name, <*gname*> is treated as the group ID.

`-user` <*uname*> — True if file belongs to the user <*uname*>. If <*uname*> is numeric and there is no such user <*uname*>, it is treated as the user ID.

`-nouser` — True if the file belongs to an unknown user.

`-nogroup` — True if the file belongs to an unknown group.

`-inum` *n* — True if the file has inode number *n*.

`-links` *n* — True if the file has *n* links.

`-ls` — Always true. Prints the following file statistics: inode number, size in 512-byte blocks, file permissions, number of hard links, owner, group, size in bytes, last modification time, and filename. If the file is a symbolic link, the display of the file it is linked to is preceded by `->`. The display from this `ls` is identical to that displayed by `ls -dgils`.

`-ok <command>`
Same as `-exec`, except that confirmation from the user is requested before executing <*command*>.

`-name <pattern>`
True if the filename contains <*pattern*>. Special shell pattern-matching characters (`[`, `]`, `*`, `?`) may be used as part of <*pattern*>. A backslash (`\`) is used to escape those characters to explicitly search for them as part of <*pattern*>.

`-path <pattern>`
True if the pathname contains <*pattern*>. Special shell pattern-matching characters (`[`, `]`, `*`, `?`) may be used as part of <*pattern*>. A backslash (`\`) is used to escape those characters to explicitly search for them as part of <*pattern*>. Slashes (`/`) are treated as normal characters and do not need to be escaped.

`-perm [-]<mode>`
<`mode`> may be either symbolic or octal (see `chmod`). If <*mode*> is symbolic, a starting value of `0` is assumed, and <*mode*> sets or clears permissions without regard to the process's file mode creation mask. If <*mode*> is octal, only bits 0777 of the file's mode bits are used in the comparison. If <*mode*> is preceded by a dash (`-`), this evaluates to true if at least all the bits in <*mode*> are set in the file's mode bits. If <*mode*> is not preceded by a dash, this evaluates to true if the bits in <*mode*> match exactly the file's mode bits. If <*mode*> is symbolic, the first character may not be a dash.

`-print0`
Always true. Prints the current pathname followed by a null character.

`-print`
Always true. Prints the current pathname followed by a newline character. If none of `-exec`, `-ls`, `-ok`, or `-print0` is specified, `-print` is assumed.

`-prune`
Always true. Does not descend into current file after the pattern has been matched. If `-d` is specified, `-prune` has no effect.

`-regex pattern`
True if the whole path of the file matches pattern using regular expression matching. To match a file named `"./Documents/zzyzygy"`, you can use the regular expression `".*/[gyz]*"` or `".*/Documents/.*"`, but not `"zzyzygy"` or `"/Documents/"` because the regular expression must match the *entire* filename, including the path portion (as opposed to the action of the `-name` argument, which would find the zzyzygy file with `-name zzyzygy`).

`-iregex pattern`
The same as `-regex`, except use a not case sensitive match.

`-size n[c]`
True if the file size, rounded up, is *n* 512-byte blocks. If `c` follows *n*, it is true if the file size is *n* bytes.

`-type` *`t`* True if the file is of the specified type. Possible file types are

`W` Whiteout

`b` Block special

`c` Character special

`d` Directory

`f` Regular file

`l` Symbolic link

`p` FIFO

`s` Socket

**Operators**

Primaries may be combined using the following operators (in order of decreasing precedence).

`(`*`expression`*`)` True if the parenthesized expression evaluates to true.

`!`*`expression`* True if the expression is false. (`!` is the unary, not the operator)

*`expression`* `[-and]` *`expression`*

*`expression expression`* True if both expressions are true. The second expression is not evaluated if the first is false. (`-and` is the logical AND operator.)

*`expression`* `-or` *`expression`* True if either expression is true. The second expression is not evaluated if the first is true. (`-or` is the logical OR operator.)

# ftp

`ftp` File transfer program.

```
ftp [-AadefginpRtvV] [-o <outfile>] [-P <port>] [-r <seconds>]
      [-T <dir>,<max>[,<inc>][[<user>@]<host> [<port>]]] [<host>:<path>[/]]
      [file:///<file>] [ftp://[<user>[:<pass>]@]<host>[:<port>]/<path>[/]]
      [http://[<user>[:<pass>]@]<host>[:<port>]/<path>] [...]
ftp -u <url> <file> [...]
```

The remote host with which `ftp` is to communicate can be specified on the command line. Done this way, `ftp` immediately tries to establish a connection with the remote host. Otherwise, `ftp` enters its command interpreter mode, awaits commands from the user, and displays the prompt `ftp>`.

`-A` Forces active mode `ftp`. By default, `ftp` tries to use passive mode `ftp` and falls back to active mode if passive is not supported by the server.

`-a` Causes `ftp` to bypass normal login procedure and use an anonymous login instead.

`-d` Enables debugging.

`-e` Disables command-line editing.

`-f` Forces a cache reload for transfers that go through the FTP or HTTP proxies.

`-g`
Disables filename globbing.

`-i`
Turns off interactive mode when transferring multiple files.

`-n`
Does not attempt auto-login upon initial connection. If auto-login is not disabled, `ftp` checks for a `.netrc` file in the user's directory for an entry describing an account on the remote machine. If no entry is available, `ftp` prompts for the login name on the remote machine (defaults to the login name on the local machine), and if necessary, prompts for a password.

`-p`
Enables passive mode operation for use behind connection filtering firewalls. This option has been deprecated as `ftp` now tries to use passive mode by default, falling back to active mode if the server does not support passive connections.

`-R`
Restarts all non-proxied auto-fetches.

`-t`
Enables packet tracing.

`-v`
Enables verbose and progress. Default if output is to a terminal (and for progress, if `ftp` is in the foreground). Shows all responses from the remote server as well as transfer statistics.

`-V`
Disables verbose and progress, overriding the default of enabled when output is to a terminal.

`-o` *<output>*
When auto-fetching files, saves the contents in `output`. If `output` is not `-` or doesn't start with `¦`, only the first file specified is retrieved into `output`; all other files are retrieved into the basename of their remote name.

`-P` *<port>*
Sets the port number to *<port>*.

`-r` *<seconds>*
Retries the connection attempt if it failed, pausing for *<seconds>* seconds.

`-T` *<direction>*,*<maximum>*[,*<increment>*]
Sets the maximum transfer rate for *<direction>* to *<maximum>* bytes/second, and if specified, the *<increment>* to *<increment>* bytes/second.

`-u` *<url>* *<file>*
Uploads files on the command line to *<url>*, where *<url>* is one of the `ftp` URL types as supported by autofetch (with an optional target filename for single file uploads), and file is one or more local files to be uploaded.

When `ftp` is in its command interpreter mode awaiting instructions from the user, there are many commands that the user might issue. Some of them include

`ascii`
Sets the file transfer type to network `ASCII`. Although this is supposed to be the default, it is not uncommon for an FTP server to indicate that `binary` is its default.

`binary`
Sets the file transfer type to support binary image transfer.

`quit`
Terminates the `ftp` session and exits `ftp`. An end of file also terminates the session and exits.

| | |
|---|---|
| `cd <remote_directory>` | Changes the current working directory on the remote host to `<remote_directory>`. |
| `lcd <directory>` | Changes the working directory on the local machine. If no directory is specified, the user's home directory is used. |
| `close` | Terminates the `ftp` session with the remote host and returns to the command interpreter. |
| `dir [<remote-directory> [<local_file>]]` | Prints a listing of the directory on the remote machine. Most Unix systems produce an `ls -l` output. If `<remote_directory>` is not specified, the current directory is assumed. If `<local_file>` is not specified, or is `-`, the output is sent to the terminal. |
| `open <hostname> [<port>]` | Attempts to establish an `ftp` connection on `<hostname>` at `<port>`, if `<port>` is specified. |
| `delete <remote_file>` | Deletes the specified `<remote_file>` on the remote machine. |
| `get <remote_file> [<local-file>]` | Downloads `<remote_file>` from the remote machine to the local machine. If `<local_file>` is not specified, the file is also saved on the local machine with the name `<remote_file>`. |
| `put <local_file> [<remote_file>]` | Uploads the specified `<local_file>` to the remote host. If `<remote_file>` is not specified, the file is saved on the remote host with the name `<local_file>`. |
| `mput <local_files>` | Uploads the specified `<local_files>`. |
| `help [<command>]` | Displays a message describing `<command>`. If `<command>` is not specified, a listing of known commands is displayed. |
| `ls [<remote_directory> [<local_file>]]` | Prints a list of the files in a directory on the remote machine. If `<remote_directory>` is not specified, the current working directory is assumed. If `<local_file>` is not specified, or is `-`, the output is printed to a terminal. Note that if nothing is listed, the directory might only have directories in it. Try `ls -l` or `dir` for a complete listing. |
| `mkdir <directory>` | Makes the specified `<directory>` on the remote machine. |
| `rmdir <directory>` | Removes the specified `<directory>` from the remote machine. |
| `passive [auto]` | Toggles passive mode if no argument is given. If `auto` is given, acts as if `FTPMODE` is set to `auto`. |
| `pwd` | Prints the current working directory on the remote host. |

# grep

grep — Prints lines matching a pattern.

egrep

fgrep.

grep [*options*] <*pattern*> <*file1*> <*file2*> ...

grep [*options*] [-e <*pattern*> ¦ -f <*file*>] <*file1*> <*file2*>

grep searches the list of files enumerated by <*file1*> <*file2*> ..., or standard input if no file or - is specified. By default, the matching lines are printed.

Two additional variants of the program are available as egrep (same as grep -E) or fgrep (same as grep -F).

-A <*num*> — Prints <*num*> lines of trailing context after matching lines.

--after-context=<*num*> — Same as -A <*num*>.

-a — Processes a binary file as if it were a text file. Equivalent to -binary-files=text option.

--text — Same as -a.

-B <*num*> — Prints <*num*> lines of leading context before matching lines.

--before-context=<*num*> — Same as -B <*num*>.

-C <*num*> — Prints <*num*> lines of output context. Default is 2.

-<*num*> — Same as -C <*num*>.

--context[=<*num*>] — Same as -C <*num*>.

-b — Prints the byte offset within the input file before each line of output.

--byte-offset — Same as -b.

--binary-files=<*type*> — Assumes a file is type <*type*> if the first few bytes of a file contain binary data.

Default <*type*> is binary, and grep normally outputs a one-line message indicating the file is binary, or nothing if there is no match. If <*type*> is without-match, it is assumed that a binary file does not match. Equivalent to -I option. If <*type*> is text, it processes the file as though it were a text file. Equivalent to -a option. Warning: Using this option could result in binary garbage being output to a terminal—some of which could be interpreted by the terminal as commands, resulting in unwanted side effects.

-I — Assumes that a binary file does not match. Equivalent to -binary-files=without-match option.

-c — Prints a count of matching lines for each file. Combined with -v, counts nonmatching lines.

--count — Same as -c.

-v — Inverts matching to select nonmatching lines.

--invert-match — Same as -v.

| | |
|---|---|
| `-d <action>` | If input file is a directory, uses `<action>` to process it. If `<action>` is `read`, `grep` reads directories as if they were normal files. This is the default. If `<action>` is `skip`, `grep` silently skips directories. If `<action>` is `recurse`, `grep` recursively reads files under the directory. Equivalent to `-r`. |
| `--directories=<action>` | Same as `-d <action>`. |
| `-r` | Recursively reads files under directories. Equivalent to `-d recurse` option. |
| `--recursive` | Same as `-r`. |
| `-f <file>` | Reads a list of patterns from `<file>`, which contains one pattern per line. An empty file has no patterns and matches nothing. |
| `--file=<file>` | Same as `-f <file>`. |
| `-e <pattern>` | Uses `<pattern>` as the pattern. Useful for protecting patterns beginning with `-`. |
| `-regexp=<pattern>` | Same as `-e <pattern>`. |
| `-G` | Interprets `<pattern>` as a basic regular expression. This is the default behavior. |
| `--basic-regexp` | Same as `-G`. |
| `-E` | Interprets `<pattern>` as an extended regular expression. Equivalent to `egrep`. |
| `-extended-regexp` | Same as `-E`. |
| `-F` | Interprets `<pattern>` as a list of fixed strings, separated by newlines—any of which is to be matched. Equivalent to `fgrep`. |
| `--fixed-strings` | Same as `-F`. |
| `-H` | Prints the filename for each match. |
| `--with-filename` | Same as `-H`. |
| `-h` | Suppresses filenames on output when multiple files are searched. |
| `--no-filename` | Same as `-h`. |
| `--help` | Displays a brief help message. |
| `-i` | Ignores case in `<pattern>` and input files. |
| `--ignore-case` | Same as `-i`. |
| `-L` | Prints a list of files that do not have matches. Stops scanning after the first match. |
| `-l` | Prints a list of files that contain matches. |
| `--mmap` | If possible, uses `mmap(2)` system call rather than the default `read(2)` system call. Sometimes `-mmap` results in better performance. However, it can cause unexpected behavior, such as core dumps, if the file shrinks while `grep` is reading it or if an I/O error occurs. |
| `-n` | Output includes the line number where the match occurs. |

`--line-number` Same as `-n`.

`-q` Quiet. Suppresses normal output. Scanning stops on the first match. Also see the `-s` and `-no-messages` options.

`--quiet` Same as `-q`.

`--silent` Same as `-q`.

`-s` Suppresses error messages about nonexistent or unreadable files.

`--no-messages` Same as `-s`.

`-V` Prints the version number of `grep` to standard error. Includes the version number in all bug reports.

`--version` Same as `-V`.

`-w` Selects only lines that have matches that form whole words.

`--word-regexp` Same as `-w`.

`-x` Selects only those matches that exactly match the whole line.

`--line-regexp` Same as `-x`.

`-Z` Outputs a zero byte (the ASCII NULL character) instead of the character that normally follows a filename. This option makes the output unambiguous, even for filenames containing unusual characters such as newlines.

`--null` Same as `-Z`.

`-y` Obsolete equivalent for `-i`.

`-U` Has no effect on platforms other than MS-DOS and MS Windows. On those platforms, `-U` treats files as binary files to affect how CR characters are handled.

`--binary` Same as `-U`.

`-u` Has no effect on platforms other than MS-DOS and MS Windows. On those platforms, reports Unix-style byte offsets; that is, with CR characters stripped off.

`--unix-byte-offsets` Same as `-u`.

## gzip, gunzip, zcat

`gzip` Compresses or expands files.

`gunzip`

`zcat`

```
gzip [-acdfhlLnNrtvV19] [-S <suffix>] <file1> <file2> ...
gunzip [-acfhlLnNrtvV] [-S <suffix>] <file1> <file2> ...
zcat [-fhLV] <file1> <file2> ...
```

gzip reduces the size of a file and renames the file by adding the .gz extension. It keeps the same ownership modes, and access and modification times. If no files are specified, or if the filename - is specified, standard input is compressed to standard output. gzip compresses regular files, but ignores symbolic links.

Compressed files can be restored to their original form by using gunzip, gzip -d, or zcat.

gunzip takes a list of files from the command line, whose names end in .gz, -gz, .z, -z, _z, or .Z, and which also begin with the correct magic number, and replaces them with expanded files without the original extension. gunzip also recognizes the extensions .tgz and .taz as short versions of .tar.gz and .tar.Z, respectively. If necessary, gzip uses the .tgz extension to compress a .tar file.

zcat is equivalent to gunzip -c. It uncompresses either a list of files on the command line or from standard input and writes uncompressed data to standard output. zcat uncompresses files that have the right magic number, whether or not they end in .gz.

Compression is always formed, even if the compressed file is slightly larger than the original file.

-a
ASCII text mode. Converts end-of-lines using local conventions. Supported only on some non-Unix systems.

--ascii
Same as -a.

-c
Writes output to standard output and keeps the original files unchanged.

--stdout
Same as -c.

--to-stdout
Same as -c.

-d
Decompresses.

--decompress
Same as -d.

--uncompress
Same as -d.

-f
Forces compression or decompression, even if the file has multiple links, if the corresponding file already exists, or if the compressed data is read from or written to a terminal. If -f is not used and gzip is not working in the background, the user is prompted before a file is overwritten.

-h
Displays a help screen and quits.

--help
Same as -h.

-l
Lists the following fields for each compressed file:

compressed (compressed size)

uncompressed (uncompressed size)

ratio (compression ratio; 0.0% if unknown)

uncompressed_name (name of uncompressed file)

Uncompressed size is -1 for files not in gzip format. To get an uncompressed size for such files, use

zcat <*file1.Z*> ¦ wc -c

Combined with -verbose, it also displays

method (compression method)

crc (32-bit CRC of the uncompressed data)

date and time (time stamp of the uncompressed file)

Compression methods supported are deflate, compress, lzh, and pack. crc is listed as ffffffff when the file is not in gzip format.

| | |
|---|---|
| `--list` | Same as `-l`. |
| `-L` | Displays the `gzip` license and quits. |
| `--license` | Same as `-L`. |
| `-n` | When compressing, it does not save the original filename and time stamp by default. (Always saves the original name if it has to be truncated.)<br>When decompressing, it does not restore the original name (removes only `.gz`) and time stamp (only copies it from compressed file), if present. This is the default. |
| `--no-name` | Same as `-n`. |
| `-N` | When compressing, it always saves the original filename and time stamp. This is the default.<br>When decompressing, it restores the original time stamp and filename, if present. |
| `--name` | Same as `-N`. |
| `-q` | Suppresses all warnings. |
| `--quiet` | Same as `-q`. |
| `-r` | Traverses the directory structure recursively.<br>If a filename specified on the command line is a directory, `gzip`/`gunzip` descends into the directory and compresses/decompresses the files in that directory. |
| `--recursive` | Same as `-r`. |
| `-S` *`<suffix>`* | Uses *`<suffix>`* instead of `.gz`. Any suffix can be used, but we recommend that suffixes other than `.z` and `.gz` be avoided to avoid confusion when transferring the file to another system.<br>A null suffix (`-S ""`) forces `gunzip` to try decompression on all listed files, regardless of suffix. |
| `--suffix` *`<suffix>`* | Same as `-S` *`<suffix>`*. |
| `-t` | Test. Checks the integrity of the compressed file. |
| `--test` | Same as `-t`. |
| `-v` | Verbose. Displays the name and percentage reduction for each file compressed or decompressed. |
| `--verbose` | Same as `-v`. |
| `-V` | Version. Displays the version number and compilation |
| options and quits. | |
| `--version` | Same as `-V`. |
| `-`*`<n>`* | |
| `--fast` | |
| `--best` | Regulates the speed of compression as specified by `-`*`<n>`*, where `-1` (or `--fast`) is the fastest compression method (least compression) and `-9` (or `--best`) is the slowest compression method (most compression). Default compression option is `-6`. |

## halt, reboot

`halt` Stops the system.

`reboot` Restarts the system.

`halt [-lnq]`

`reboot [-lnq]`

The `halt` and `reboot` utilities flush the system cache to disk, send all running processes a SIGTERM and subsequently a SIGKILL and, respectively, halts or restarts the system. The action is logged, including adding a shutdown record into the login accounting file.

`-l` The halt or reboot is not logged to the system log..

`-n` Does not flush the file system cache. This option probably should not be used.

`-q` Quickly and ungracefully halts/restarts the system, and only flushes the file system cache. This option probably should not be used.

Normally, the shutdown(8) utility is used when the system needs to be halted or restarted, giving users advance warning of their impending doom and cleanly terminating specific programs.

## head

`head` Displays the first lines of a file.

`head [-n <`*`number`*`>] <`*`file1`*`> <`*`file2`*`> ...`

`-`*`n`* `<`*`number`*`>` Displays the first `<`*`number`*`>` of lines. If *`n`* is not specified, the default is `10`.

## ifconfig

`ifconfig` Configures network interface parameters.

```
ifconfig [-L] [-m] <interface> [create] <address_family> [<address[/prefixlength]>
[<dest address>]] [<parameters>]

ifconfig <interface> destroy
ifconfig -a [-L] [-d] [-m] [-u] [<address_family>]
ifconfig -l [-d] [-u] [<address_family>]
ifconfig [-L] [-d] [-m] [-u]
```

`ifconfig` assigns an address to a network interface and/or configures network interface parameters. It must be used at boot time to define the network address of each network interface. It may also be used at a later time to redefine an interface's network address or other operating parameters.

Only the super user can modify the configuration of a network interface.

`-m` If passed before an interface name, `ifconfig` displays all the supported media for the specified interface.

`-L` Displays address lifetime for IPv6 addresses, as time offset string.

`-a` Produces a full listing of all available interfaces.

-l

Produces a name-only listing of all available interfaces.

-d

Limits a listing to those interfaces that are down.

-u

Limits a listing to those interfaces that are up.

Available options for ifconfig are

*<address>*

For the DARPA-Internet family, the address is either a hostname in the hostname database or a DARPA-Internet address expressed in the Internet standard dot notation.

*<address family>*

Specifies the *<address family>*, which affects interpretation of the remaining parameters. The address or protocol families currently supported are inet, iso, and ns.

*<dest address>*

Specifies the address of the correspondent on the other end of a point to point link.

*<interface>*

The *<interface>* parameter is a string of the form *<name of physical unit>*, such as en0.

The following parameters may be set with ifconfig:

add

Another name for the alias parameter. Introduced for compatibility with BSD/OS.

alias

Establishes an additional network address for this interface. This is sometimes useful when changing network numbers, while still accepting packets for the old interface. A *<netmask>* should be used with this parameter. If the new *<alias>* address is on the same subnet as an existing address assigned to this interface, the netmask must be 255.255.255.255. If a netmask is not supplied, the command uses the one implied by the address itself. If the all ones netmask is used, the system handles route installation. If another is used, a route to that address might have to be added by hand; for example, route add -host xx.xx.xx.xx -interface 127.0.0.1, where xx.xx.xx.xx is the alias. In either case, the route might have to be deleted by hand when the alias is removed (-alias or delete).

-alias

Removes the network address specified.

anycast

(Inet6 only.) Specifies that the address configured is an anycast address. Based on the current specification, only routers may configure anycast addresses. Anycast addresses will not be used as source address of any of outgoing IPv6 packets.

arp

Enables the use of the Address Resolution Protocol in mapping between network-level addresses and link-level addresses (default). This is currently implemented for mapping between DARPA-Internet addresses and 10 Mb/s Ethernet addresses.

-arp

Disables the use of the Address Resolution Protocol.

`broadcast`
(inet only) Specifies the address to use to represent broadcasts to the network. The default broadcast address is the address with a host part of all 1s.

`debug`
Enables driver-dependent bugging code. This usually turns on extra console logging.

`-debug`
Disables driver-dependent debugging code.

`delete`
Removes the network address specified. This is used if you incorrectly specified an alias or if it's no longer needed.

`down`
Marks an interface as down. When an interface is marked down, the system does not attempt to transmit messages through that interface. If possible, the interface is reset to disable reception as well. This doesn't automatically disable routes using the interface.

`ether`
Another name for the `lladdr` parameter.

`lladdr <addr>`
Sets the link-level address on an interface. This can be used to—for example, set a new MAC address on an ethernet interface, though the mechanism used is not ethernet specific. The address `<addr>` is specified as a series of colon-separated hex digits. If the interface is already up when this option is used, it is briefly brought down and then brought back up again in order to ensure that the receive filter in the underlying ethernet hardware is properly reprogrammed.

`media <type>`
If the driver supports the media selection system, sets the media type of the interface to type. Some interfaces support the mutually exclusive use of one of several different physical media connectors. For example, a 10Mb/s Ethernet interface might support the use of either AUI or twisted pair connectors. Setting the media type to `10base5/AUI` would change the currently active connector to the AUI port. Setting it to `10baseT/UTP` would activate twisted pair. Refer to the interfaces' driver specific documentation or man page for a complete list of the available types.

`mediaopt <opts>`
If the driver supports the media selection system, sets the specified media options on the interface. The `<opts>` argument is a comma delimited list of options to apply to the interface. Refer to the interfaces' driver specific man page for a complete list of available options.

`-mediaopt <opts>`
If the driver supports the media selection system, disables the specified media options on the interface.

`tunnel <src-addr> <dest-addr>`
(IP tunnel devices only.) Configures the physical source and destination address for IP tunnel interfaces (gif(4)). The arguments `<src_addr>` and `<dest_addr>` are interpreted as the outer source/destination for the encapsulating IPv4/IPv6 header.

`deletetunnel`
Unconfigures the physical source and destination address for IP tunnel interfaces previously configured with `tunnel`.

`create`

Create the specified network pseudo-device. If the interface is given without a unit number, tries to create a new device with an arbitrary unit number. If creation of an arbitrary device is successful, the new device name is printed to standard output.

`destroy`

Destroys the specified network pseudo-device.

`plumb`

Another name for the `create` parameter. Included for Solaris compatibility.

`unplumb`

Another name for the `destroy` parameter. Included for Solaris compatibility.

`metric <`*`n`*`>`

Sets the routing metric of the interface to <*n*>; the default is `0`. The routing metric is used by the routing protocol. Higher metrics make a less favorable route. Metrics are counted as addition hops to the destination network or host.

`mtu <n>`

Sets the maximum transmission unit of the interface to `n`; default is interface specific. The MTU is used to limit the size of packets that are transmitted on an interface. Not all interfaces support setting the MTU, and some interfaces have range restrictions.

`netmask <`*`mask`*`>`

(inet and ISO) Specifies how much of the address to reserve for subdividing networks into subnetworks. The mask includes the network part of the local address and the subnet part, which is taken from the host field of the address. The mask can be specified as a single hexadecimal number beginning with `0x`, as a dot-notation Internet address, or as a pseudo-network name listed in the network table networks. The mask contains 1s for the bit positions in the 32-bit address that are to be used for the network and subnet parts, and 0s for the host part.

`prefixlen <len>`

(Inet6 only.) Specifies that `len` bits are reserved for subdividing networks into sub-networks. The `<len>` must be integer, and for syntactical reasons it must be between `0` to `128`. It is almost always `64` under the current IPv6 assignment rule. If the parameter is omitted, `64` is used.

`remove`

Another name for the `-alias` parameter. Introduced for compatibility with BSD/OS.

`link[0-2]`

Enables special processing of the link level of the interface.

`-link[0-2]`

Disables special processing at the link level with the specified interface.

`up`

Marks an interface as `up`. Can be used to enable an interface after `ifconfig down` has been run. It happens automatically when setting the first address on an interface. If the interface was reset when previously marked down, the hardware is reinitialized.

## kill

`kill` Sends a signal to a process or terminates a process

`kill [-<signal> ¦ -s <signal name>] <pid> ...`

`kill -l [exit-status]`

`-l [exit-status]` With no argument, lists all the signal names; otherwise lists the signal associated with the status *`exit-status`*.

`-s <signal name>` A symbolic signal name specifying the signal to be sent instead of the default `TERM`. This is an alternative to the next, more popular, form.

`<signal>` Specifies which signal to send to a process. If *`<signal>`* is not specified, the `TERM` (terminate) signal is sent. *`<signal>`* may be a number or name.

`<pid>` Specifies the process ID that should receive a signal. The process ID can be determined by running `ps`.

Signal `KILL (9)` is a sure way to kill a process. Signal `HUP` is another common signal to send to a process. You often can send a `HUP` signal to a process to get it to reread its configuration file.

## killall

`killall` Kills processes by name.

`killall [-d ¦ -v] [-help ¦ -h ¦ -?] [-l] [-m] [-s] [-u <user>] [-t <tty>] [-c <procname>] [-<SIGNAL>] [<procname> ...]`

`Killall` kills processes selected by name, as opposed to the selection by pid as done by `kill`. By default, it sends a `TERM` signal to all processes with a real UID identical to the caller of `killall` that match the name `<procname>`.

`-d ¦ -v` Verbose mode. For a single `-d` option, a list of the processes that will be sent the signal will be printed, or a message indicating that no matching processes have been found. For `–v`, the process ID is printed.

`-help` Gives help on the command usage and exits.

`-l` Lists the names of the available signals and exits, as in `kill`.

`-m` Matches the argument `<procname>` as a (not case sensitive) regular expression against the names of processes found. Caution! This is dangerous; a single dot will match any process running under the real UID of the caller.

`-s` Silent mode. Shows only what would be done, but does not send any signal.

`-<SIGNAL>` Sends the specified `<SIGNAL>` instead of the default `TERM`. The signal may be specified either as a name (with or without a leading SIG), or numerically.

`-u <user>` Limits potentially matching processes to those belonging to the specified `<user>`.

-t <tty>
Limits potentially matching processes to those running on the specified <tty>.

-c <procname>
When used with the -u or -t flags, limits potentially matching processes to those matching the specified <procname>.

# last

last
Indicates last logins of users and ttys.

last [-n] [-f *<file>*] [-h *<host>*] [-t *<tty>*] [*<user1>* *<user2>* ...]

last lists the sessions of specified users, ttys, and hosts, in reverse time order. Each line of output contains the username, the tty from which the session was conducted, any hostname, the start and stop times for the session, and the duration of the session. If the session is still in progress or was cut short by a crash or shutdown, last indicates that.

-n
Limits the report to n lines.

-f *<file>*
Reads *<file>* instead of the default /var/log/wtmp.

-h *<host>*
Lists sessions from *<host>*. *<host>* may be a name or Internet number.

-t *<tty>*
Lists sessions on *<tty>*. *<tty>* may be given fully or abbreviated. For example, last -t p3 is equivalent to last -t ttyp3.

If multiple arguments are given, the information that applies to any of the arguments is printed. For example, last root -t console would list all sessions of root as well as all sessions on the console.

The pseudo-user reboot logs in at system reboot, so last reboot gives an indication of the mean time between reboots.

# less

less
Pages through data or text files.

less -?

less --help

less -V

less --version

less [-[+]aBcCdeEfFgGiIJmMnNqQrRsSuUVwWX~][-b *<bufs>*][-h *<lines>*][-j *<line>*] [-k *<keyfile>*] [--{oO} *<logfile>*] [-p *<pattern>*][-t *<tag>*] [-T *<tagsfile>*] [-x *<tab>*] [-y *<lines>*] [-[-z] *<lines>*][+[+]*<cmd>*] [--] [*<file1>*...]

## Summary of less Commands

Commands marked with * may be preceded by a number, *N*.

Notes in parentheses indicate the behavior if *N* is given.

h H
Displays this help.

q :q Q :Q ZZ
Exits.

### MOVING

| | | | | | | |
|---|---|---|---|---|---|---|
| e | ^E | j | ^*N* | CR | * | Forward one line (or *N* lines). |
| y | ^Y | k | ^K | ^P | * | Backward one line (or *N* lines). |
| f | ^F | ^V | SPACE | | * | Forward one window (or *N* lines). |
| b | ^B | ESC-v | | | * | Backward one window (or *N* lines). |
| z | | | | | * | Forward one window (and set window to *N*). |
| w | | | | | * | Backward one window (and set window to *N*). |
| ESC-SPACE | | | | | * | Forward one window, but don't stop at end-of-file. |
| d | ^D | | | | * | Forward one half-window (and set half-window to *N*). |
| u | ^U | | | | * | Backward one half-window (and set half-window to *N*). |
| ESC-( | RightArrow | | | | * | Left 8 character positions (or *N* positions). |
| ESC-) | LeftArrow | | | | * | Right 8 character positions (or *N* positions). |
| F | | | | | | Forward forever; like "tail -f". |
| r | ^R | ^L | | | | Repaint screen. |
| R | | | | | | Repaint screen, discarding buffered input. |

### SEARCHING

| | | |
|---|---|---|
| /pattern | * | Search forward for (*N*-th) matching line. |
| ?pattern | * | Search backward for (*N*-th) matching line. |
| N | * | Repeat previous search (for *N*-th occurrence). |
| N | * | Repeat previous search in reverse direction. |
| ESC-n | * | Repeat previous search, spanning files. |
| ESC-N | * | Repeat previous search, reverse dir. and spanning files. |
| ESC-u | | Undo (toggle) search highlighting. |

## OPTIONS

Most options may be changed either on the command line, or from within less by using the - or -- command.

Options may be given in one of two forms: either a single character preceded by a -, or a name preceded by --.

| | | | | |
|---|---|---|---|---|
| -? | | --help | | Displays help (from command line). |
| -a | | --search-skip-screen | | Forwards search skips current screen. |
| -b [N] | --buffers=[N] | | | Number of buffers. |
| -B | | --auto-buffers | | Doesn't automatically allocate buffers for pipes. |
| -c | -C | --clear-screen<br>--CLEAR-SCREEN | | Repaints by scrolling/clearing. |
| -d | --dumb | | | Dumb terminal. |
| -e | -E | --quit-at-eof<br>--QUIT-AT-EOF | | Quits at end of file. |
| -f | | --force | | Forces open non-regular files. |
| -F | | --quit-if-one-screen | | Quits automatically if the file can be displayed on a single screen. |
| -g | | --hilite-search | | Highlights only last match for searches. |
| -G | | --HILITE-SEARCH | | Doesn't highlight any matches for searches. |
| -h [N] | --max-back-scroll=[N] | | | Backward scroll limit. |
| -I | | --ignore-case | | Ignores case in searches. |
| -I | | --IGNORE-CASE | | Ignores case in searches and in search patterns. |
| -j [N] | --jump-target=[N] | | | Screens position of target lines. |
| -J | | --status-column | | Displays a status column at the left side of the screen with the number of lines that matched the current search. |
| -k [file] | | --lesskey-file=[file] | | Uses a lesskey file. |
| -m | -M | --long-prompt<br>--LONG-PROMPT | | Sets prompt style. |
| -n | -N | --line-numbers<br>--LINE-NUMBERS | | Uses line numbers. |
| -o [file] | | --log-file=[file] | | Copies to log file (standard input only). |
| -O [file] | | --LOG-FILE=[file] | | Copies to log file (unconditionally overwrite). |
| -p [pattern] | | --pattern=[pattern] | | Starts at pattern (from command line). |
| -P [prompt] | | --prompt=[prompt] | | Defines new prompt. |
| -q | -Q | --quiet<br>--QUIET | --silent<br>–SILENT | Quiets the terminal bell. |

| | | |
|---|---|---|
| `-r` | `--raw-control-chars` | Outputs “raw” control characters. |
| `-R` | `--RAW-CONTROL-CHARS` | Outputs “raw” control characters and tracks screen appearance. |
| `-s` | `--squeeze-blank-lines` | Squeezes multiple blank lines. |
| `-S` | `--chop-long-lines` | Chops long lines. |
| `-t [tag]` | `--tag=[tag]` | Finds a tag. |
| `-T [tagsfile]` | `--tag-file=[tagsfile]` | Uses an alternate tags file. |
| `-u -U` | `--underline-special`<br>`--UNDERLINE-SPECIAL` | Changes handling of backspaces. |
| `-V` | `--version` | Displays the version number of `"less"`. |
| `-w` | `--hilite-unread` | Highlights first new line after forward-screen. |
| `-W` | `--HILITE-UNREAD` | Highlights first new line after any forward movement. |
| `-x [N]` | `--tabs=[N]` | Sets tab stops. |
| `-X` | `--no-init` | Doesn’t use termcap init/deinit strings. |
| `-y [N]` | `--max-forw-scroll=[N]` | Forwards scroll limit. |
| `-z [N]` | `--window=[N]` | Sets size of window. |
| `-" [c[c]]` | `--quotes=[c[c]]` | Sets shell quote characters. |
| `-~` | `--tilde` | Doesn’t display tildes after end of file. |

## locate

| | |
|---|---|
| `locate` | Finds files. |

`locate <pattern>`

Searches a database for –all pathnames that match `<pattern>`. The database is rebuilt periodically and contains the names of all publicly accessible files.

Shell and globbing characters (`*`, `?`, `\`, `[`, and `]`) may be used in `<pattern>`, although they must be escaped. Preceding a character by `\` eliminates any special meaning for it. No characters must be explicitly matched, including `/`.

As a special case, a pattern with no globbing characters (`foo`) is matched as (`*foo*`).

Useful files:

| | |
|---|---|
| `/var/db/locate.database` | Database |
| `/usr/libexec/locate.updatedb` | Script to update database |

# ln, link

`ln` Makes links.

```
ln [-fhinsv] <source> <target>
```

```
ln [-fhinsv] <source1> <source2> <source3> .. <directory>
```

In the first form, `ln` links *<source>* to *<target>*. If *<target>* is a directory, a link named *<source>* is placed in *<target>*.

In the second form, `ln` makes links to the files enumerated by *<source1>* *<source2.* *<source3>* .. in *<directory>*. The links have the same names as the sources in the list.

There are two types of links: hard links and symbolic links. The default is hard links. A hard link to a file is indistinguishable from the original directory entry. Hard links may not normally refer to directories and may not span file systems.

A symbolic link refers by name to the file to which it is linked. Symbolic links may refer to directories and may span file systems.

`-f` Forces the link to occur by unlinking any already existing links.

`-h` If *<target>* or *<directory>* is a symbolic link, it is not followed. This is most useful when used with `-f`, to replace a symbolic link that might point to a directory.

`-i` (Interactive) Causes `ln` to write a prompt to standard error if the target file exists.

`-n` Same as `-h`. Retained for compatibility with other implementations of `ln`.

`-s` Creates a symbolic link; this is most like the idea of aliases you're already familiar with.

`-v` Verbose

# lp

`lp` Sends a job to the printer.

```
lp [ -E ] [ -c ] [ -d <printer> ] [ -h <hostname> ] [ -m ] [-n <num-copies>]
[ -o <option> ] [ -q <priority> ] [ -s ] [ -t <title> ] [ -H <handling> ]
[ -P <page-list> ] [ <file1> <file2> .. ]
```

```
lp [ -E ] [ -c ] [ -h <server> ] [ -i <job-id> ] [ -n <num-copies>] [ -o <option> ]
[ -q <priority> ] [ -t <title> ] [ -H <handling> ] [ -P <page-list> ]
```

`-E` Forces encryption when connecting to the server.

`-c` Option is provided for backward compatibility only. On systems that support it, this option forces the print file to be copied to the spool directory before printing. In CUPS, print files are always sent to the scheduler via IPP, which has the same effect.

`-d` *<printer>* Prints to the specified *<printer>*.

`-h` *<server>* Specifies the print server hostname. The default is localhost or the value of the `CUPS_SERVER` environment variable.

| | |
|---|---|
| -i <*job-id*> | Specifies an existing job to modify. |
| -m | Sends email when the job is completed (not supported in CUPS 1.1.). |
| -n <*num-copies*> | Sets the number of copies to print from 1 to 100. |
| -o <*option*> | Sets a job option |
| -q <*priority*> | Sets the job priority from 1 (lowest) to 100 (highest). The default priority is 50. |
| -s | Silent mode. Does not report the resulting job IDs. |
| -t <*title*> | Sets the job name. |
| -H <*handling*> | Specifies when the job should be printed. A value of immediate prints the file immediately, a value of hold holds the job indefinitely, and a time value (HH:MM) will hold the job until the specified time. Use a value of resume with the -i option to resume a held job. |
| -P <*page-list*> | Specifies which pages to print in the document. The list can contain a list of numbers and ranges (#-#) separated by commas (for example, 1,3-5,16). |

# lpadmin

lpadmin Configures CUPS printers and classes.

```
lpadmin [ -E ] [ -h <server> ] -d <destination>
lpadmin [ -E ] [ -h <server> ] -p <printer> <option(s)>
lpadmin [ -E ] [ -h <server> ] -x <destination>
```

lpadmin configures printer and class queues provided by CUPS. It can also be used to set the system default printer or class.

When specified before the -d, -p, or -x options, the –E option forces encryption when connecting to the server.

The first form of the command sets the default printer or class to <destination>. Subsequent print jobs submitted via the lp(1) or lpr(1) commands use this destination unless the user specifies otherwise.

The second form of the command configures the named <printer>.

The third form of the command deletes the printer or class <destination>. Any jobs that are pending for the <destination> are removed and any job that is currently printing is aborted.

**Printer Queue Configuration Options**

| | |
|---|---|
| -c <class> | Adds the named <printer> to <class>. If <class> does not exist, it is created automatically. |
| -i <interface> | Sets a System V style interface script or the printer. This option cannot be specified with the –P option (PPD file) and is intended for providing support for legacy printer drivers. |
| -m <model> | Sets a standard System V interface script or PPD file from the model directory. |

`-o <name>=<value>` Sets a PPD or server option for the printer. PPD options can be listed using the `-l` option with the `lpoptions`(1) command.

`-o job-k-limit=<value>` Sets the kilobyte limit for per-user quotas. The `<value>` is an integer number of kilobytes; one kilobyte is 1024 bytes.

`-o job-page-limit=<value>` Sets the page limit for per-user quotas. The `<value>` is the integer number of pages that can be printed; double-sided pages are counted as two pages.

`-o job-quota-period=<value>` Sets the accounting period for per-user quotas. The `<value>` is an integer number of seconds; `86,400` seconds are in one day.

`-r <class>` Removes the named `<printer>` from `<class>`. If the resulting class becomes empty, it is removed.

`-u allow:<user>,<user>`

`-u deny:<user>,<user>`

`-u allow:all`

`-u deny:none` Sets user-level access control on a printer. The latter two forms turn user-level access control off.

`-v <device-uri>` Sets the `device-uri` attribute of the printer queue. If `<device-uri>` is a filename, it is automatically converted to the form `file:/file/name`.

`-D <info>` Provides a textual description of the printer.

`-E` Enables the printer and accepts jobs; this is the same as running the `accept(8)` and `enable(8)` programs on the printer.

`-L <location>` Provides a textual location of the printer.

`-P <ppd-file>` Specifies a PostScript Printer Description file to use with the printer. If specified, this option overrides the `-i` option (interface script).

# lpinfo

`lpinfo` Shows available printing devices and drivers.

`lpinfo [ -E ] [ -l ] [ -m ] [ -v ]`

`-E` Forces encryption when connecting to the server.

`-l` Shows a “long” listing of devices or drivers.

`-m` Shows the available printer drivers on the system. This option is useful for discovering what `-m` models are available for use with the `lpadmin` command.

`-v` Shows the available printer devices on the system.

# lpoptions

| | |
|---|---|
| `lpoptions` | Displays or sets printer options and defaults. |
| `lpoptions -d <printer>` | |
| `lpoptions [-p <printer>] -l` | |
| `lpoptions -p <printer> -o <option>[=<value>] ...` | |
| `lpoptions -x <printer>` | |
| `-d <printer>` | Sets the default printer to `<printer>`. Overrides the system default printer for the current user. |
| `-l` | Lists the printer specific options and their current settings. |
| `-o <option>=<value>` | Specifies a new option for the named destination. |
| `-p <printer>` | Sets the destination to `<printer>`. |
| `-x <printer>` | Removes the options for the named destination. |

If no options are specified using the `-o` option, the current options for the named printer are reported on the standard output.

Options set with the `lpoptions` command are used by the `lp(1)` and `lpr(1)` commands when submitting jobs.

# lpr

| | |
|---|---|
| `lpr` | Sends a job to the printer. |

```
lpr [ -E ] [ -P <printer> ] [ -# <num-copies> [ -l ] [ -o
    <option> ] [ -p] [ -r ] [ -C/J/T <title> ] [ <file1> <file2> .. ]
```

`lpr` submits files for printing. Files named on the command line are sent to the specified printer (or the default system printer if none is specified). If no files are listed on the command line, `lpr` reads the print file from the standard input.

| | |
|---|---|
| `-E` | Forces encryption when connecting to the server. |
| `-P` *<printer>* | Specifies *<printer>* as the printer. Otherwise, the site's default printer is used. |
| *-# <num-copies>* | Sets the number of copies to print from 1 to 100. |
| `-l` | Specifies that the print file is already formatted for the destination and should be sent without filtering. Option is equivalent to `-oraw`. |
| `-o` *<option>* | Sets a job option. |
| `-p` | Specifies that the print file should be formatted with a shaded header with the date, time, job name, and page number. Option is equivalent to `-oprettyprint` and is only useful when printing text files. |
| `-r` | Removes the named print files after printing them. |
| `-C` *<title>* | Sets the job name. |
| `-J` *<title>* | Sets the job name. |
| `-T` *<title>* | Sets the job name. |

Options `c`, `d`, `f`, `g`, `i`, `m`, `n`, `t`, `v` and `w` are not supported by the CUPS system and produce a warning message if used.

# lprm

lprm — Removes print jobs from the queue.

```
lprm [ -E ] [ - ] [ -P <printer> ] [ <job#1> <job#2> .. ]
```

-E — Forces encryption when connecting to the server.

- — Removes all print jobs in the queue.

-P<*printer*> — Specifies <*printer*> as the printer. Otherwise, the site's default is used.

<*job#*> — Removes from the queue the print job specified by <*job#*>. The <*job#*> can be determined by using lpq(1).

# lpstat

lpstat — Prints CUPS status information.

```
lpstat [ -E ] [-a [<printer(s)>]] [-c [<class(es)>]] [ -d ] [ -h <server> ] [ -l ]
[-o [<destination(s)>]] [ -p [<printer(s)>]] [ -r ] [ -R ] [ -s ] [ -t ] [ -u
[<user(s)>]] [ -v [<printer(s)>]]
```

-E — Forces encryption when connecting to the server.

-a [<*printer(s)*>] — Shows the accepting state of printer queues. If no printers are specified, all printers are listed.

-c [<*class(es)*>] — Shows the printer classes and the printers that belong to them. If no classes are specified then all classes are listed.

-d — Shows the current default destination.

-h <*server*> — Specifies the CUPS server to communicate with.

-l — Shows a long listing of printers, classes, or jobs.

-o [<destination(s)>] — Shows the jobs queue on the specified destinations. If no destinations are specified, all jobs are shown.

-p [<*printer(s)*>] — Shows the printers and whether or not they are enabled for printing. If no printers are specified, all printers are listed.

-r — Shows whether the CUPS server is running.

-R — Shows the ranking of print jobs.

-s — Shows a status summary, including the default destination, a list of classes and their member printers, and a list of printers and their associated -d, -c, and -p options.

-t — Shows all status information. This is equivalent to using the -r, -d, -c, -d, -v, -a, -p, and -o options.

-u [<*user(s)*>] — Shows a list of print jobs queued by the specified users. If no users are specified, lists the jobs queued by the current user.

-v [<*printer(s)*>] — Shows the printers and what device they are attached to. If no printers are specified, all printers are listed.

## lpq

`lpq` Displays the queue of print jobs.

`lpq [ -E ] [ -P <printer> ] [ -a ] [ -l ] [ +<interval> ]`

`-E` Forces encryption when connecting to the server.

`-P <printer>` Specifies `<printer>` as the printer. Otherwise, the site's default printer is used.

`-a` Displays the queues for all printers.

`-l` Displays the queue information in long format. Includes the name of the host from which the job originated.

`+<interval>` Displays a continuous report of the jobs in the queue once every `<interval>` seconds until the queue is empty.

## ls

`ls` Lists directory contents.

`ls -ABCFGHLPRTWZabcdfghiklmnopqrstuwx1] <file1> <file2> ...`

`-A` Lists all entries except for "`.`" and "`..`". Always set for super user.

`-B` Forces printing of non-printable characters (as defined by `ctype(3)` and current locale settings) in filenames as \xxx, where *xxx* is the numeric value of the character in octal.

`-C` Forces multicolumn output. This is the default when output is to a terminal.

`-F` Displays a symbol, if applicable, after each file to denote the following: slash (`/`) for a directory; asterisk (`*`) for an executable; an at sign (`@`) for a symbolic link; a percent sign (`%`) for a whiteout; an equal sign (=) for a socket; a vertical bar (`¦`) for a FIFO.

`-G` Enables colorized output.

`-H` Symbolic links on the command line are followed. This option is assumed if none of the `-F`, `-d`, or `-l` options are specified.

`-L` If the argument is a symbolic link, the file or directory the link references rather than the link itself is displayed.

`-P` If argument is a symbolic link, lists the link itself rather than the object the link references. This option cancels the `-H` and `-L` options.

`-R` Recursively lists subdirectories.

`-T` Displays complete time information, including month, day, hour, minute, second, and year.

`-W` Displays whiteouts.

-a Lists all files in the directory, including files whose names begin with a dot (.).

-c Uses time when file status was last changed for sorting or printing.

-d If the argument is a directory, it is listed as a plain file, rather than listing its contents. If the argument is a symbolic link, its link information is not displayed.

-f Does not sort output.

-g Does nothing. Kept for compatibility with older versions of ls.

-h When used with the -l option, uses unit suffixes: Byte, Kilobyte, Megabyte, Gigabyte, Terabyte, and Petabyte in order to reduce the number of digits to three or less using base 2 for sizes.

-i Lists the argument's serial number (inode number).

-k Displays sizes in kilobytes.

-l Lists in long format. Displays file mode, number of links, owner name, group name, size of the file in bytes, date and time file was last modified, and the file. If displayed to a terminal, the first line of output is the total number of 512-byte blocks used by the files in the directory.

-m Streams output format; lists files across the page, separated by commas.

-n Displays user and group ID as numbers rather than names in a long (-l) output.

-o Includes file flags in a long (-l) output.

-p Writes a slash (`/') after each filename if that file is a directory.

-q Forces printing of nongraphic characters in filenames as character ?. This is the default when output is to a terminal.

-r Reverses sort order to reverse alphabetic order; smallest first or oldest first, as appropriate.

-s Displays file size in 512-byte blocks, where partial units are rounded up to the next integer value. If the output is to a terminal, first line displayed is the total number of 512-byte blocks used by files in the directory.

-t Sorts by time modified (most recently modified first) before sorting in alphabetic order.

-u Uses time of last access for sorting (-t) or printing (-l).

-w Forces raw printing of non-printable characters. This is the default when output is not to a terminal.

-x Forces multicolumn output sorted across the page rather than down the page.

`-v` Forces unedited printing of nongraphic characters. This is the default when output is not to a terminal.

`-1` Forces output to one entry per line. This is the default when output is not to a terminal.

## man

`man` Formats and displays online manual pages.

```
man [-adfhkotw] [-M <path>] [-P <pager>] [-S <list>] [-m <machine>] [-p <string>]
[<section>] <name1> <name2> ...
```

`-a` Displays all the manual pages for a specified section and name combination. (The default is to display only the first page found.)

`-d` Displays debugging information, rather than manual pages.

`-f <keyword>` Displays a list of manual pages that contain complete word matches to the `<keyword>`. Same as `whatis`.

`-h` Displays the help for `man`.

`-k <keyword>` Displays a list of manual pages that contain the `<keyword>`. Same as `apropos`.

`-o` Looks for original, non-localized manpages only.

`-t` Uses `troff` to format the manual pages and outputs to stdout. The `troff` output can then be passed to a filter before being printed.

`-w` Lists the pathnames of manual pages that would be displayed for the specified section and name combination.

`-M <path>` Overrides the list of standard directories where `man` searches for manual pages. The path specified must be a colon-separated list of directories. The search path can also be specified by the `MANPATH` environment variable.

`-P <pager>` Uses the specified `<pager>` to display the manual pages.

`-S <list>` Searches the specified colon-separated section `<list>`. Overrides the `MANSECT` environment variable.

`-m <machine>` Searches for alternate architecture man pages.

`-p <string>` Specifies the sequence of preprocessors to run before `nroff` or `troff`. Some of the preprocessors and the letters used to designate them are: `eqn` (`e`), `grap` (`g`), `pic` (`p`), `tbl` (`t`), `vgrind` (`v`), `refer` (`r`). Overrides the `MANROFFSEQ` environment variable.

The optional `<section>` argument restricts `man`'s search to the specified section.

## mkdir

`mkdir` Makes directories.

```
mkdir [-p-v] [-m <mode>] <dir1> <dir2> ..
```

`mkdir` creates the named directories in the order specified, using mode `rwxrwxrwx (0777)` as modified by the current `umask (2)`.

The user must have write permission in the parent directory.

`-p` Creates all nonexistent parent directories first. If this option is not specified, the full path prefix of each operand must already exist. Intermediate directories are created with permission bits `rwxrwxrwx (0777)` as modified by the current `umask (2)`, plus write and execute permission for the owner. For example, if you are creating the directory `/Users/jray/Images/Vacation` and the `Images` directory doesn't already exist, the `-p` option will automatically force its creation.

`-m` *<mode>* Sets the permission bits of the created directory to *<mode>*. *<mode>* can be in any formats specified to the `chmod (1)` utility. If a symbolic mode is specified, the operation characters `+` and `-` are interpreted relative to an initial mode of `a=rwx`.

`-v` Be verbose.

## mount

`mount` Mounts file systems.

```
mount
mount [-adfruvw] [-t ufs ¦ lfs ¦ <external_type>]
mount [-dfruvw] <special> ¦ <node>
mount [-dfruvw] [-o <options>] [-t ufs ¦ lfs ¦ <external_type>] <special> ¦ <node>
```

`mount` invokes a file system–specific program to prepare and graft the *<special>* device or remote node (`rhost:path`) on the file system tree at the point *<node>*. If neither *<special>* nor *<node>* is specified, the appropriate information is taken from the `fstab` file.

The system maintains a list of currently mounted file systems. If no arguments are given to `mount`, this list is displayed.

`-a` All the file systems described in `fstab`(5) are mounted. Exceptions are those marked as `noauto` or are excluded by the `–t` flag.

`-d` Causes everything to be done except for the actual system call. Useful in conjunction with the `-v` option to determine what the `mount` command is trying to do.

`-f` Forces the revocation of write access when trying to downgrade a file system `mount` status from read-write to read-only.

`-r`
Mounts the file system read-only (even `root` may not write to it). The same as the `rdonly` option to the `-o` option.

`-u`
Indicates that the status of an already mounted file system should be changed. Any of the options available in `-o` may be changed. The file system may be changed from read-only to read/write, or vice versa. An attempt to change from read/write to read-only fails if any files on the file system are currently open for writing unless `-f` is also specified.

`-v`
Enables verbose mode.

`-w`
Sets the file system object to read/write.

`-t ufs ¦ lfs ¦<`*`external_type`*`>`
Specifies a file system type. Default is type `ufs`. The option can also be used to indicate that the actions should be performed only on the specified file system type. More than one type may be specified in a comma-separated list. The prefix `no` added to the type list may be used to specify that the actions should not take place on a given type. For example, `mount -a -t nonfs,mfs` indicates that all file systems should be mounted except those of type NFS and MFS. `mount` attempts to execute a program called `mount_XXX`, where *XXX* is the specified typename.

`-o`
Specifies certain options. The options are specified in a comma-separated list.

The following options are available for the `-o` option:

`async`
Specifies that all I/O to the file system should be done asynchronously. This is a dangerous flag to set, and should not be used unless you are prepared to re-create the file system if the system crashes.

`force`
Same as `-f`. Forces the revocation of write access when trying to downgrade a file system `mount` status from read/write to read-only.

`noauto`
Skips this file system when `mount` is run with the `-a` flag.

`nodev`
Does not interpret character or block special devices on the file system. The option is useful for a server that has file systems containing special devices for architectures other than its own.

`noexec`
Does not allow the execution of any binaries on the mounted file system. This option is useful for a server containing binaries for an architecture other than its own.

`nosuid`
Does not allow `set-user-identifier` or `set-group-identifier` bits to take effect.

`rdonly`
Same as `-r`. Mounts the file system read-only. Even `root` may not write to it.

`sync` Specifies that all I/O to the file system should be done synchronously.

`update` Same as `-u`. Indicates that the status of an already mounted file system should be changed.

`union` Causes the namespace at the `mount` point to appear as the union of the mounted file system root and the existing directory. Lookups are done on the mounted file system first. If operations fail because of a nonexistent file, the underlying file system is accessed instead. All creates are done in the mounted file system.

Any additional options specific to a given file system type may be passed as a comma-separated list. The options are distinguished by a leading `-`. Options that take a value have the syntax `-<`*`option`*`>=<`*`value`*`>`.

## mv

`mv` Moves files.

```
mv [-finv] <source> <target>
mv [-finv] <source1> <source2> <source3> .. <directory>
```

In the first form, `mv` renames `<`*`source`*`>` to the name provided by `<`*`target`*`>`. If `<`*`source`*`>` is a file, a file is renamed. Likewise, if `<`*`source`*`>` is a directory, a directory is renamed.

In the second form, `mv` moves the list enumerated by `<`*`source1`*`> <`*`source2`*`> <`*`source3`*`> ..` to the directory named by `<`*`directory`*`>`.

`-f` Forces an existing file to be overwritten.

`-i` Invokes an interactive mode that prompts for a confirmation before overwriting an existing file.

`-n` Do not overwrite an existing file.

`-v` Be verbose.

## netstat

`netstat` Shows network status.

```
netstat [-AaLlnW] [-f <address_family> ¦ -p <protocol>] [-M <core>] [-N <system>]
netstat [-gilns] [-f <address_family>] [-M <core>] [-N <system>]
netstat -i ¦ -I <interface> [-w <wait>] [-abdgt] [-M <core>] [-N <system>]
netstat -s [-s] [-f <address_family> ¦ -p <protocol>] [-M core] [-N <system>]
netstat -i ¦ -I interface -s [-f <address_family> ¦ -p <protocol>] [-M <core>]
[-N <system>]
netstat -m [-M <core>] [-N <system>]
netstat -r [-Aaln] [-f <address_family>] [-M <core>] [-N <system>]
netstat -rs [-s] [-M <core>] [-N <system>]
```

The netstat command symbolically displays the contents of various network-related data structures. There are a number of output formats, depending on the options for the information presented. The following forms (respectively by their order above) are available:

1. Displays a list of active sockets for each protocol.
2. Presents the contents of one of the other network data structures according to the option selected.
3. With a wait interval specified, netstat will continuously display the information regarding packet traffic on the configured network interfaces.
4. Displays statistics for the specified protocol or address family.
5. Displays per-interface statistics for the specified protocol or address family.
6. Displays mbuf(9) statistics.
7. Displays routing table for the specified address family.
8. Displays routing statistics.

-A
With the default display, shows the address of any protocol control blocks associated with sockets; used for debugging.

-a
With the default display, shows the state of all sockets; normally sockets used by server processes are not shown. With the routing table display (option -r, as described below), shows protocol-cloned routes (routes generated by a RTF_PRCLONING parent route); normally, these routes are not shown.

-b
With the interface display (option -i, as described below), shows the number of bytes in and out.

-d
With either interface display (option -i or an interval, as described below), shows the number of dropped packets.

-f <address-family>
Limits statistics or address control block reports to those of the specified address family. The following address families are recognized: inet, for AF_INET, inet6, for AF_INET6 and unix, for AF_UNIX.

-g
Shows information related to multicast (group address) routing. By default, show the IP Multicast virtual-interface and routing tables. If the -s option is also present, shows multicast routing statistics.

-I <interface>
Shows information about the specified interface; used with a wait interval as described below. If the -s option is present, shows per-interface protocol statistics on the interface for the specified <address_family> or <protocol>, or for all protocol families.

`-i` Shows the state of interfaces which have been auto-configured (interfaces statically configured into a system, but not located at boot time are not shown). If the `-a` options is also present, multicast addresses currently in use are shown for each Ethernet interface and for each IP interface address. Multicast addresses are shown on separate lines following the interface address with which they are associated. If the `-s` option is present, shows per-interface statistics on all interfaces for the specified `<address_family>` or `<protocol>`, or for all protocol families.

`-L` Shows the size of the various listen queues. The first count shows the number of unaccepted connections. The second count shows the amount of unaccepted incomplete connections. The third count is the maximum number of queued connections.

`-l` Prints full IPv6 address.

`-M` Extracts values associated with the name list from the specified core instead of the default `/dev/kmem`.

`-m` Shows statistics recorded by the memory management routines. (The network manages a private pool of memory buffers.)

`-N` Extracts the name list from the specified system instead of the default `/kernel`.

`-n` Shows network addresses as numbers. (Normally `netstat` interprets addresses and attempts to display them symbolically.) This option may be used with any of the display formats.

`-p <protocol>` Shows statistics about `<protocol>`, which is either a well-known name for a protocol or an alias for it. Some protocol names and aliases are listed in the file `/etc/protocols`. The special protocol name `bdg` is used to show bridging statistics. A null response typically means that there are no interesting numbers to report. The program complains if `<protocol>` is unknown or if there is no statistics routine for it.

`-r` Shows the routing tables. Use with `-a` to show protocol-cloned routes. When `-s` is also present, shows routing statistics instead. When `-l` is also present, `netstat` assumes more columns are there and the maximum transmission unit (mtu) are also displayed.

`-s` Shows per-protocol statistics. If this option is repeated, counters with a value of zero are suppressed.

`-W` In certain displays, avoids truncating addresses even if this causes some fields to overflow.

`-w <wait>` Shows network interface statistics at intervals of `<wait>` seconds.

### Output

The default display, for active sockets, shows the local and remote addresses, sends and receive queue sizes (in bytes), protocol, and the internal state of the protocol. Address formats are of the form `<host>.<port>` or `<network>.<port>` if a socket's address specifies a network but no specific host address. If known, the host and network addresses are displayed symbolically according to the databases `/etc/hosts` and `/etc/networks`, respectively. If a symbolic name for an address is unknown, or if the `-n` option is specified, the address is printed numerically, according to the address family. Unspecified, or wildcard, addresses and ports appear as `*`.

## nidump

`nidump` Extracts text or flat-file–format data from NetInfo.

`nidump [-t] { -r <directory> ¦ <format> } <domain>`

`nidump` reads the specified NetInfo domain and dumps a portion of its contents to standard output. When a flat-file administration format is specified, `nidump` provides output in the syntax of the corresponding flat file. Allowed values for `<format>` are `aliases`, `bootparams`, `bootptab`, `exports`, `fstab`, `group`, `hosts`, `networks`, `passwd`, `printcap`, `protocols`, `rpc`, and `services`.

If `-r` is used, the first argument is interpreted as a NetInfo directory path, and its contents are dumped in a generic NetInfo format.

`-t` Interprets the domain as a tagged name.

`-r` Dumps the specified directory in raw format. Directories are delimited in curly brackets. Properties within a directory are listed in the form `property = value;`. Parentheses introduce a comma-separated list of items. The special property name `CHILDREN` is used to hold a directory's children, if any. Spacing and line breaks are significant only within double quotes, which can be used to protect any names with meta characters.

## niload

`niload` `niload` populates NetInfo directories with multiple properties at once.

`niload [-v] [-d] [-m] [-p] [-t] {-r <directory> ¦ <format>} <domain>`

`niload` loads information from standard output into the specified NetInfo `<domain>`. If `<format>` is specified, the input is interpreted according to the flat-file format `<format>`. Acceptable values for `<format>` are `aliases`, `bootparams`, `bootptab`, `exports`, `fstab`, `group`, `hosts`, `networks`, `passwd`, `printcap`, `protocols`, `rpc`, and `services`.

If `-r <directory>` is specified instead of a flat-file format, the input is interpreted as raw NetInfo data, as generated by `nidump -r`, and is loaded into `<directory>`.

`niload` overwrites entries in the existing directory with those contained in the input. Entries that are in the directory, but not in the input, are not deleted unless `-d` is specified. `niload` must be run as the `superuser` on the master NetInfo server for `<domain>`, unless `-p` is specified.

`-v` Verbose mode. Prints `+` for each entry loaded, and `-` for each entry deleted (flat-file formats only).

`-d`
Deletes entries that are in the directory, but not in the input.

`-m`
Merges properties and values. Existing properties will be preserved in the database if they are not present in the input. For example, if a user record has a “picture” property, loading a passwd-format record for this user will preserve the property. Property values are also merged.

`-p`
Prompts for the `root` password of the given domain so that the command can be run from locations other than the master.

`-u <user>`
Authenticates as `<user>`. Implies `-p`.

`-P <password>`
Provides `<password>` on the command line. Overrides `-p`.

`-t`
Interprets the domain as a tagged domain. For example, `trotter/network` refers to the `domain` network on the machine `trotter`. Machine name can be specified as an actual name or an IP address.

`-r`
Loads entries in raw format, as generated by `nidump -r`. The first argument should be the path of a NetInfo directory into which the information is loaded. The specified directory may be renamed as a result of contents of the input, particularly if the input includes a top-level `name` property. If the specified directory does not exist, it is created.

`<domain>`
NetInfo `<domain>` that is receiving input. If `.` is the value for `<domain>`, it is referring to the local NetInfo database.

## niutil

`niutil`
The NetInfo Utility `niutil` is used to edit the NetInfo database.

```
niutil -create [opts] <domain> <path>
niutil -destroy [opts] <domain> <path>
niutil -createprop [opts] <domain> <path> <key> [<val>...]
niutil -appendprop [opts] <domain> <path> <key> <val>...
niutil -mergeprop [opts] <domain> <path> <key> <val>...
niutil -insertval [opts] <domain> <path> <key> <val> <index>
niutil -destroyprop [opts] <domain> <path> <key>
niutil -destroyval [opts] <domain> <path> <key> <val>
niutil -renameprop [opts] <domain> <path> <oldkey> <newkey>
niutil -read [opts] <domain> <path>
niutil -list [opts] <domain> <path>
niutil -rparent [opts] <domain>
niutil -resync [opts] <domain>
niutil -statistics [opts] <domain>
```

`niutil` enables you to perform arbitrary reads and writes on the specified NetInfo `<domain>`. To perform writes, `niutil` must be run as `root` on the NetInfo master for the database, unless `-p`, `-P`, or `-u` is specified. The directory specified by `<path>` is separated by `/` characters. A numeric ID may be used for a path in place of a string. Property names may be given in a path with an `=`. The default property name is `name`. The following examples refer to a user with user ID 3:

```
/name=users/uid=3
/users/uid=3
```

`-t <host>/<tag>` Interprets the domain as a tagged domain. For example, `parrish/network` is the domain tagged network on machine `parrish`.

`-p` Prompts for the `root` password or the password of `<user>` if combined with `-u`.

`-u <user>` Authenticates as `<user>`. Implies `-p`.

`-P <password>` Provides the `root` password or the password of `<user>` if combined with `-u`. Overrides `-p`.

`-T <seconds>` Sets the read and write timeout to `<seconds>`. Default is 30 seconds.

## Operations

`-create <domain> <path>` Creates a new directory with the specified path.

`-destroy <domain>` Destroys the directory with the specified path. `<path>`

`-createprop <domain> <path>` Creates a new property in the directory `<path>`. `<key>` is the name of the property. Zero or more property values `<key> [<val>...]` may be specified. If the named property already exists, it is overwritten.

`-appendprop <domain> <path>` Appends new values to an existing property in directory `<path>`. `<key>` is the name of the property. Zero or more property values `<key> <val>..` may be specified. If the named property does not exist, it is created.

`-mergeprop <domain> <path>` Merges new values into an existing property in the directory `<path>`. `<key>` is the name of the property. Zero or more `<key> <val>..` property values may be specified. The values are appended to the property only if they do not already exist. If the named property does not exist, it is created.

`-insertval <domain> <path>` Inserts a new value into an existing property in the directory `<path>` at position `<propindex>`. `<key>` is the name of the `<key> <val>` property. If the named property does not exist, it is created. `<propindex>`

`-destroyprop <domain> <path> <key>` Destroys the property with name `<key>` in the specified `<path>`.

`-destroyval <domain> <path> <key> <val>`

Destroys the specified value in the property named `<key>` in the specified `<path>`.

`-renameprop <domain> <path> <oldkey> <newkey>`

Renames the property with the name `<oldkey>` in the specified `<path>`.

`-read <domain> <path>`

Reads the properties associated with the directory `<path>` in the specified `<domain>`.

`-list <domain> <path>`

Lists the directories in the specified `<domain>` and `<path>`. Directory IDs are listed along with directory names.

`-readprop <domain> <path><path> <key>`

Reads the value of the property named `<key>` in the directory of the specified `<domain>`.

`-readval <domain> <path> <key> <index>`

Reads the value at the given index of the named property in the specified directory.

`-rparent <domain>`

Prints the current NetInfo parent of a server. The server should be explicitly given using the `-t <host>/<tag>` option.

`-resync <domain>`

Resynchronizes NetInfo. If a domain name is given, the master resynchronizes all clones. If the `-t <clone>/<tag>` option is used instead, only that clone is resynchronized. Using `-t <master>/<tag>` resynchronizes the whole domain.

`-statistics <domain>`

Prints server statistics on the specified `<domain>`.

`-domainname <domain>`

Prints the domain name of the given domain.

`<domain>`

A value of `.` for `<domain>` refers to the local NetInfo database.

## nohup

`nohup`

Invokes a command immune to hang-ups.

`nohup <utility> [<arg> ...]`

`nohup` invokes `<utility>` with its arguments and at this time sets the signal `SIGHUP` to be ignored. If the standard output is a terminal, the standard output is appended to the file `nohup.out` in the current directory. If standard error is a terminal, it is directed to the same place as the standard output.

The following variable is utilized by `nohup`:

`HOME`

If the output file `nohup.out` cannot be created in the current directory, the `nohup` utility uses the directory named by `HOME` to create the file.

## open

`open` Opens files and directories.

```
open [-a <application>] <file> ...
open [-e] <file> ...
```

The open command opens a file (or a directory or URL), just as if you had double-clicked the file's icon. If no application name is specified, the default application as determined via LaunchServices is used to open the specified files.

If the file is in the form of a URL, the file will be opened as a URL.

You can specify one or more filenames (or pathnames), which are interpreted relative to the shell or Terminal window's current working directory. For example, the following command would open all Word files in the current working directory:

```
open *.doc
```

`-a <application>` Specifies the application to use for opening the file.

`-e` Causes the file to be opened with /Applications/TextEdit.app.

## osacompile

`osacompile` Compiles OSA scripts.

```
osacompile [-l <language>] [-e <command>] [-o <name>] [-d] [-r <type:id>]
[-t <type>] [-c <creator>] [-xsu] [<file ...>]
```

osacompile compiles the given files, or standard input if none are listed, into a single output script. Files may be plain text or other compiled scripts. The options are as follows:

`-l <language>` Overrides the language for any plain text files. Normally, plain text files are compiled as AppleScript.

`-e <command>` Enters one line of a script. Script commands given via -e are prepended to the normal source, if any. Multiple -e commands may be given to build up a multi-line script. Because most scripts use characters that are special to many shell programs (For example, AppleScript uses single and double quote marks, "(", ")", and "*"), the command will have to be correctly quoted and escaped to get it past the shell intact.

`-o <name>` Places the output in the filename. If -o is not specified, the resulting script is placed in the file "a.scpt".

`-d` Places the resulting script in the data fork of the output file.

`-r <type:id>` Places the resulting script in the resource fork of the output file, in the specified resource.

`-t <type>` Sets the output file type to type. Type is a four-character code. If this option is omitted and the output file does not exist, the type is set to "osas"—that is, a compiled script.

-c *<creator>* Sets the output file creator to creator. Creator is a four-character code. If this option is omitted and the output file does not exist, the creator is set to "ToyS"—that is, Script Editor.

-x Saves the resulting script as execute only.

-s Stay-open applet.

-u Uses startup screen.

If no options are specified, osacompile produces a classic Mac OS format script file—that is, type "osas" (compiled script), creator "ToyS" (Script Editor), with the script data in the scpt:128 resource and nothing in the data fork. This format is compatible with all Mac OS and Mac OS X systems.

The -d and -r options are not exclusive. If exactly one is specified, the script is written only to that fork. If both are specified, the script is written to both forks.

## osascript

osascript Executes OSA scripts.

osascript [-l *<language>*] [-e *<command>*] [-s *<flags>*] [*<programfile>*]

osascript executes the given script file, or standard input if none is given. Scripts may be plain text or compiled scripts. osascript was designed for use with AppleScript, but will work with any Open Scripting Architecture (OSA) language. To get a list of the OSA languages installed on your system, use osalang(1). The options are as follows:

-e *<command>* Enter one line of a script. If -e is given, osascript will not look for a filename in the argument list. Multiple -e commands may be given to build up a multi-line script. Because most scripts use characters that are special to many shell programs (for example, AppleScript uses single and double quote marks, ``('', ``)'', and ``*''), the command will have to be correctly quoted and escaped to get it past the shell intact.

-l *<language>* Overrides the language for any plain text files. Normally, plain text files are compiled as AppleScript.

-s *<flags>* Modifies the output style. The flags argument is a string consisting of any of the modifier characters e, h, o, and s. Multiple modifiers can be concatenated in the same string, and multiple -s options can be specified. The modifiers come in exclusive pairs; if conflicting modifiers are specified, the last one takes precedence. The meanings of the modifier characters are as follows:

h Prints values in human-readable form (default).

s Prints values in recompilable source form.

`osascript` normally prints its results in human-readable form: strings do not have quotes around them, characters are not escaped, braces for lists and records are omitted, and so on. This is generally more useful, but can introduce ambiguities. For example, the lists `` `{"foo", "bar"}' `` and `` `{{"foo", {"bar"}}}' `` would both be displayed as `` `foo, bar' ``. To see the results in an unambiguous form that could be recompiled into the same value, use the `s` modifier.

`e` Prints script errors to `stderr` (default).

`o` Prints script errors to `stdout`.

## passwd

`passwd` Modifies a user's password.

`passwd [-l] [-k] [-y] [<`*`user`*`>]`

`passwd` changes the user's local, Kerberos, or YP password. The user is first prompted for her old password. The user is next prompted for a new password, and then prompted again to retype the new password for verification.

The new password should be at least six characters in length. It should use a variety of lowercase letters, uppercase letters, numbers, and metacharacters.

`-l` Updates the user's local password.

`-k` Updates the Kerberos database, even if the user has a local password. After the password has been verified, `passwd` transmits the information to the Kerberos authenticating host.

`-y` Updates the YP password, even if the user has a local password. The `rpc.yppasswdd` (8) daemon should be running on the YP master server.

If no flags are specified, the following occurs:

If Kerberos is active, the user's Kerberos password is changed, even if the user has a local password.

If the password is not in the local database, an attempt to update the YP password occurs.

To change another user's Kerberos password, run `kinit (1)` followed by `passwd`. The superuser is not required to supply the user's password if only the local password is being modified.

## pbcopy

`pbcopy` Copies data from `STDIN` into the clipboard/pasteboard.

`pbcopy [-help] [-pboard general¦ruler¦find¦font]`

`-help` Displays `pbcopy` help

`[-pboard general¦ruler¦find¦font]` Chooses the pasteboard that will receive the data. By default, *general* is used.

# pbpaste

`pbpaste` Writes textual data from the MacOS X clipboard/pasteboard to `STDOUT`.

```
pbpaste -help
```

```
pbpaste [-pboard general¦ruler¦find¦font] [-Prefer rtf¦ps¦ascii]
```

`pbpaste` pastes textual data from the clipboard/pasteboard to the command line via `STDOUT`. The `-P` option allows you to suggest a preferred output format, but is not necessarily obeyed.

`-P rtf` Prefers output in Rich Text Format if available.

`-P ps` Prefers output in PostScript format if available.

`-P ascii` Prefers American Standard Code for Information Interchange (yup, that's what ascii stands for) plain text format.

`[-pboard general¦ruler¦find¦font]` Chooses the pasteboard that will supply the data. By default, *general* is used.

# ping

`ping` Sends `ICMP ECHO_REQUEST` packets to network hosts.

```
ping [-Rdfnqrv] [-c <count>] [-i <wait>] [-l <preload>]
    [-p <pattern>] [-s <packetsize>] <host>
```

`ping` uses the ICMP protocol's mandatory `ECHO_REQUEST` datagram to elicit an `ICMP ECHO_RESPONSE` from a host or gateway. `ECHO_REQUEST` datagrams (pings) have an IP and ICMP header, followed by a `struct timeval` and then an arbitrary number of pad bytes used to fill out the packet.

`-d` Sets the `SO_DEBUG` option on the socket being used.

`-f` Flood `ping`. Outputs packets as fast as they come or one hundred times per second, whichever is more. Only `root` may use this option. This option can be very hard on a network and should be used with caution.

`-n` Displays numeric output only. Does not make any attempt to lookup symbolic names for host addresses.

`-q` Enables quiet output. Displays only the summary lines at startup time and when finished.

`-R` Record route. Includes `RECORD_ROUTE` option in the `ECHO_REQUEST` packet and displays the route buffer on returned packets. The IP header is large enough for only nine such routes. Many hosts ignore or discard this option.

`-r` Bypasses the normal routing tables and sends directly to a host on the attached network. If the host is not on a directly attached network, an error is returned. This option can be used to ping a local host through an interface that has no route through it.

`-v` Enables verbose output. Lists ICMP packets received other than `ECHO_RESPONSE` packets.

`-c <count>` Stops after sending and receiving `<count>` `ECHO_RESPONSE` packets.

`-i <wait>` Sets the interval between sending each packet to `<wait>` seconds. Default is to wait one second. This option is incompatible with the `-f` option.

`-l <preload>` Sends `<preload>` number of packets as fast as possible before falling into its normal mode of behavior. Only `root` may set a preload value.

`-p <pattern>` Up to 16 pad bytes can be specified to fill out a packet that is sent. This is useful for diagnosing data-dependent problems in a network. For example, `-p ff` causes the sent packet to be filled with all 1s.

`-s <packetsize>` Specifies the number of data bytes to be sent. The default is 56, which translates to 64 ICMP data bytes when combined with the 8 bytes of ICMP header data.

## ps

`ps` Displays process status report.

`ps [-aCcefhjMmrSTuvwx] [-O <fmt>] [-o <fmt>] [-p <pid>] [-t <tty>] [-U <username>]`

`ps [-L]`

`-a` Includes information about processes owned by others in addition to yours.

`-C` Changes the way CPU percentage is calculated by using a raw CPU calculation that ignores resident time. This normally has no effect.

`-c` Changes the command column output to contain just the executable name rather than the full command line.

`-e` Displays the environment.

`-f` Shows command-line and environment information about swapped-out processes. This is honored only if the user's user ID is `0`.

`-h` Repeats the header information so that there is one header per page of information.

`-j` Prints information associated with the following keywords: `user`, `pid`, `ppid`, `pgid`, `sess`, `jobc`, `state`, `tt`, `time`, and `command`.

`-l` Displays information associated with the following keywords: `uid`, `pid`, `ppid`, `cpu`, `pri`, `nice`, `vsz`, `rss`, `wchan`, `state`, `tt`, `time`, and `command`.

`-M` Prints the threads corresponding with each task.

`-m` Sorts by memory usage rather than by process ID.

`-r` Sorts by current CPU usage rather than by process ID.

`-S` Changes the way the process time is calculated by summing all exited children to their parent process.

`-T` Displays information about processes attached to the device associated with standard output.

`-u` Displays information associated with the following keywords: `user`, `pid`, `%cpu`, `%mem`, `vsz`, `rss`, `tt`, `state`, `start`, `time`, and `command`. The `-u` option implies the `-r` option.

`-v` Displays information associated with the following keywords: `pid`, `state`, `time`, `sl`, `re`, `pagein`, `vsz`, `rss`, `lim`, `tsiz`, `%cpu`, `%mem`, and `command`. The `-v` option implies the `-m` option.

`-w` Uses 132 columns to display information instead of the default, which is your window size. If the `-w` option is specified more than once, `ps` uses as many columns as necessary, regardless of your window size.

`-x` Displays information about processes without controlling terminals.

`-O <`*`fmt`*`>` Adds the information associated with the space- or comma-separated list of keywords specified, after the process ID, in the default information displayed. Keywords may be further defined with an = and a string. Keywords further specified in this manner are displayed in the header as specified rather than using the standard header.

`-o <`*`fmt`*`>` Displays information associated with the space- or comma-separated list of keywords specified. Keywords may be further defined with an = and a string. Keywords further specified in this manner are displayed in the header as specified rather than using the standard header.

`-p <`*`pid`*`>` Displays information associated with the specified process ID <*`pid`*>.

`-t <`*`tty`*`>` Displays information about processes attached to the specified terminal device <*`tty`*>.

`-U <`*`username`*`>` Displays information about processes belonging to the specified <*`username`*>.

`-L` Lists the set of available keywords.

The following is a list of the definitions of the keywords that some of the options already include. More keywords are available than are defined here.

| | |
|---|---|
| `%cpu` | Percentage CPU usage (alias `pcpu`) |
| `%mem` | Percentage memory usage (alias `pmem`) |
| `command` | Command and arguments |
| `cpu` | Short-term CPU usage factor (for scheduling) |
| `jobc` | Job control count |
| `lim` | Memory use limit |
| `nice` | Nice value (alias to `ni`) |
| `pagein` | Pageins (total page faults) |

| | |
|---|---|
| `pgid` | Process group number |
| `pid` | Process ID |
| `ppid` | Parent process ID |
| `pri` | Scheduling priority |
| `re` | Core residency time (in seconds; `127` = infinity) |
| `rss` | Resident set size (real memory) |
| `rsz` | Resident set size + (text size/text use count) (alias `rs-size`) |
| `sess` | Session pointer |
| `sl` | Sleep time (in seconds; `127` = infinity) |
| `start` | Time started |
| `state` | Symbolic process state (alias `stat`) |
| `tsiz` | Text size (in kilobytes) |
| `tt` | Control terminal name (two-letter abbreviation) |
| `uid` | Effective user ID |
| `user` | Username (from `uid`) |
| `vsz` | Size of process in virtual memory in kilobytes (alias `vsize`) |
| `wchan` | Wait channel (as a symbolic name) |

# pwd

| | |
|---|---|
| `pwd` | Prints current working directory. |
| `pwd [-L¦P]` | |
| `-L` | Prints the logical path to the current working directory, as defined by the shell in the environment variable `PWD`. |
| `-P` | Default. Prints the physical path to the current working directory, with symbolic links resolved. |

# rm

| | |
|---|---|
| `rm` | Removes files. |
| `rm [-dfiPRrvW]` *`<file1> <file2>`* `...` | |
| `-d` | Attempts to remove directories as well as other types of files. |
| `-f` | Forces the removal of files without prompting the user for confirmation. If the file does not exist, no error diagnostic is displayed. The `-f` option overrides any previous `-i` options. |
| `-i` | Invokes an interactive mode that prompts for confirmation before removing a file. The `-i` option overrides any previous `-f` options. |

-P Overwrites regular files before deleting them. Files are overwritten three times before being deleted; first with byte pattern 0xff, and then 0x00, and then 0xff.

-R or -r Attempts to recursively remove files. Implies -d option.

-v Be verbose.

-W Attempts to undelete files. This option can be used to recover only files covered by whiteouts.

rm removes symbolic links, but not the files referenced by the links.

Also, attempting to remove the files . and . is an error.

## rmdir

rmdir Removes directories.

```
rmdir [-p] <directory1> <directory2> ...
```

rmdir removes each *<directory>* argument specified, provided it is empty. Arguments are processed in the order listed on the command line. To remove a parent directory and subdirectories of the parent directory, the subdirectories must be listed first.

-p Attempts to remove the specified directory and its parent directories, if they are empty.

## scp

scp Secure remote copy.

```
scp [-pqrvBC46] [-F <ssh_config>] [-S <program>] [-P <port>] [-c <cipher>]
[-i <identity_file>] [-o <ssh_option>] [[<user>@]<host1>:]<file1> [...]
[[<user>@]<host2>:]<file2>
```

-p Preserves modification times, access times, and modes from the original file.

-q Disables the progress meter.

-r Recursively copies entire directories.

-v Verbose mode. Causes scp and ssh to print debugging messages.

-B Selects batch mode, which prevents passwords or passphrases from being requested.

-C Enables compression. Passes the flag to ssh(1) to enable compression.

-4 Forces scp to use IPv4 addresses only.

-6 Forces scp to use IPv6 addresses only.

-F *<ssh_config>* Specifies an alternative per-user configuration file for ssh. Option is directly passed to ssh(1).

-S *<program>* Specifies *<program>* to use for the encrypted connection. Program must understand ssh(1) options.

-P *<port>* Specifies the port to connect to on the remote host.

-c *<cipher>* Selects the cipher to use for encrypting the data transfer. Option is passed directly to ssh(1).

-i *<identity_file>* Specifies the file from which the identity (private key) for RSA authentication is read.

-o *<ssh_option>* Passes specified options to ssh in the format used in ssh_config(5)

## screencapture

screencapture Takes pictures of the current state of the screen.

```
screencapture [-[i¦m]wsWx] <file>
```

```
screencapture [-[i¦m]cwsWx]
```

screencapture takes pictures of the current state of the screen or screens present for on the machine, or of windows or selectable regions of the screen. screencapture saves its output in .pdf format or places it on the clipboard.

-i Captures the screen interactively, by selection or window. Pressing the spacebar toggles between

- region selection (crosshair cursor)
- window selection (camera cursor)

Pressing <esc> cancels the capture.

-c Places the screen capture on the clipboard, instead of into a file.

-m Only captures the main monitor. Undefined if -i is present.

-w Only allows window selection mode.

-s Only allows mouse selection mode.

-W Starts interaction in window selection mode.

-x Does not play sounds.

## sftp

sftp Secure file transfer program.

```
sftp [-vC1] [-b <batchfile>] [-o <ssh_option>] [-s <subsystem> ¦ <sftp_server>]
[-B <buffer_size>] [-F <ssh_config>] [-P <sftp_server path>] [-R <num_requests>]
[-S <program>] <host>
```

```
sftp [[<user>@]<host>[:<file1> [<file2>]]]
```

```
sftp [[<user>@]<host>[:<dir>[/]]]
```

The first usage initiates an interactive session.

The second usage retrieves files automatically if a non-interactive authentication is used. Otherwise, it retrieves the specified files after interactive authentication.

The third usage causes sftp to start in an interactive session in the specified directory.

-b *<batchfile>*
Batch mode. Reads a series of commands from an input batchfile instead of stdin. Because it lacks user interaction, it should be used in conjunction with non-interactive authentication. sftp aborts if any of the following commands fail: get, put, rename, ln, rm, mkdir, chdir, lchdir, and lmkdir.

-o *<ssh_option>*
Passes options to ssh in the format used in the ssh configuration file. Useful for specifying options for which there is no separate sftp command-line flag. For example, to specify an alternate port use: sftp -oPort=24.

-s <subsystem> ¦ <sftp_server>
Specifies the SSH2 subsystem or the path for an sftp server on the remote host. A path is useful for using sftp over protocol version 1, or when the remote sshd does not have an sftp subsystem configured.

-v
Raises logging level. Option is also passed to ssh.

-B *<buffer_size>*
Specifies the size of the buffer that sftp uses when transferring files. Larger buffers require fewer round trips at the cost of higher memory consumption. Default is 32768 bytes.

-C
Enables compression (via ssh's –C flag).

-F *<ssh_config>*
Specifies an alternative per-user configuration file for ssh. Option is passed directly to ssh.

-P *<sftp_server path>*
Connects directly to a local sftp-server (rather than via ssh). May be useful in debugging the client and server.

-R *<num_requests>*
Specifies how many requests may be outstanding at any one time. Increasing this may slightly improve file transfer speed but will increase memory usage. Default is 16 outstanding requests.

-S *<program>*
Specifies *<program>* as the program to use for the encrypted connection. The program must understand ssh options.

-1
Specifies the use of protocol version 1.

### Common Interactive Commands

cd *<path>*
Changes remote directory to *<path>*.

lcd *<path>*
Changes local directory to *<path>*.

exit
Quits sftp.

get [*<flags>*] *<remote-path>* [*<local-path>*]
Retrieves the *<remote-path>* and stores it on the local machine. If the local pathname is not specified, it is given the same name it has on the remote machine. If the -P flag is specified, the file's full permission and access time are copied too.

help
Displays help text.

lls [*<ls-options>* [*<path>*]]
Displays local directory listing of either *<path>* or current directory if *<path>* is not specified.

`lmkdir <path>` Creates local directory specified by `<path>`.

`lpwd` Prints local working directory.

`ls [<path>]` Displays remote directory listing of either <path> or current directory if <path> is not specified.

`mkdir <path>` Creates remote directory specified by `<path>`.

`put [<flags>] <local-path> [<remote-path>]` Uploads `<local-path>` and stores it on the remote machine. If the remote path name is not specified, it is given the same name it has on the local machine. If the `-P` flag is specified, the file's full permission and access time are copied too.

`pwd` Displays remote working directory.

`quit` Quits `sftp`.

`rmdir <path>` Removes remote directory specified by `<path>`.

`rm <path>` Deletes remote file specified by `<path>`.

## shutdown

`shutdown` Closes down the system at a given time.

`shutdown [-] [-fhkrn] <time> [<warning_message>]`

`shutdown` provides an automatic way for the super user to politely notify users of an impending shutdown.

`-f` `shutdown` arranges for file systems to not be checked upon reboot.

`-h` Halts the system at the specified `<time>` when shutdown executes `halt` (8).

`-k` Kicks everybody off. The `-k` option does not actually halt the system, but does leave the system multiuser with logins disabled for all users except the super user.

`-r` Shuts the system down and executes `reboot` (8) at the specified `<time>`.

`-n` Prevents normal `sync` (2) before stopping.

`<time>` The time when the system is to be brought down. `<time>` can be the word `now` for immediate shutdown, or a future time in one of two formats: `<+number>` or `<yymmddhhmm>`, where the year, month, and day may be defaulted to the current system values. The first form brings the system down in `<number>` minutes and the second at the absolute time specified.

`<warning_message>` Any other arguments comprise the warning message that is broadcast to users currently logged on the system.

`-` Reads the warning message from standard input.

Starting at ten hours before shutdown, the system displays the shutdown warning message. Warning messages are displayed at regular intervals, with the messages being displayed more frequently as impending shutdown approaches. Five minutes before shutdown, or immediately, if shutdown is in less than five minutes, logins are disabled by creating an `/etc/nologin` and copying the warning message there. The file is removed just before shutdown occurs.

At shutdown time, a message is written in the system log, with the time of shutdown, who initiated shutdown, and the reason.

## ssh

`ssh`

`slogin` Secure shell remote login client.

```
ssh [-l <login_name>] <hostname> ¦ <user>@<hostname> [<command>]
```

```
ssh [-aAfgknqtTvxXCNP1246] [-b <bind_address>] [-c <cipher_spec>] [-e <escap_char>]
[-i <identity_file>] [-l <login_name>] [-m <mac_spec>] [-o <option>] [-p <port>]
[-F <configfile>] [-L <port>:<host>:<hostport>] [-R <port>:<host>:<hostport>]
[-D <port>] [<hostname> ¦ <user>@<hostname>] [<command>]
```

`-a` Disables forwarding of the authentication agent connection.

`-A` Enables forwarding of the authentication agent connection. This can also be specified on a per-host basis in a configuration file.

`-f` Requests `ssh` to go to background just before command execution. Implies `-n`. The recommended way to start X11 programs at a remote site is `ssh -f <host> xterm`.

`-g` Allows remote hosts to control local forwarded ports.

`-k` Disables forwarding of Kerberos tickets and AFS tokens. This may also be specified on a per-host basis in a configuration file.

`-n` Redirects `stdin` from `/dev/null`.

`-q` Quiet mode.  Causes warning and diagnostic messages to be suppressed.

`-t` Forces pseudo-tty allocation. Useful for executing arbitrary screen-based programs on a remote machine.

`-T` Disables pseudo-tty allocation.

`-v` Verbose mode. Causes debugging messages to be printed.

`-x` Disables X11 forwarding.

`-X` Enables X11 forwarding. This can also be specified on a per-host basis in a configuration file.

`-C` Requests compression of all data.

`-N` Does not execute a remote command. Useful for just forwarding ports. SSH2 only.

| | |
|---|---|
| `-P` | Uses a nonprivileged port for outgoing connections. Useful if your firewall does not permit connections from privileged ports. Turns off RhostsAuthentication and RhostsRSAAuthentication. |
| `-1` | Forces SSH1 protocol only. |
| `-2` | Forces SSH2 protocol only. |
| `-4` | Forces `ssh` to use IPv4 addresses only. |
| `-6` | Forces `ssh` to use IPv6 addresses only. |
| `-b` *<bind_address>* | Specifies the interface to transmit from on machines with multiple interfaces or aliased addresses. |
| `-c blowfish¦3des¦des` | Selects the cipher to use for the session. `3des` is the default. |
| `-c` *<cipher_spec>* | Additionally, for SSH2, a comma-separated list of ciphers. |
| `-e ch¦^ch¦none` | Sets escape character for sessions with a pty (default: ~). The escape character is only recognized at the beginning of a line. Followed by a `.` closes the connection; followed by `^Z` suspends the connection; followed by itself sends the escape character once. Setting it to `none` disables any escapes and makes the session fully transparent. |
| `-i` *<identity_file>* | Specifies the file from which the identity (private key) for RSA authentication is read. Default is `$HOME/.ssh/identity`. |
| `-l` *<login_name>* | Specifies the user to log in as on the remote machine. This may also be specified on a per-host basis in a configuration file. |
| `-m` *<mac_spec>* | Additionally, for SSH2, a comma-separated list of MAC (message authentication code) algorithms can be specified in order of preference. |
| `-o` *<option>* | Can be used for giving options in the format used in the configuration file. Useful for specifying options that have no separate command-line flag. Option has the same format as a line in the configuration file. |
| `-p` *<port>* | Specifies the port to connect to on the remote host. This can be specified on a per-host basis in the configuration file. |
| `-D` *<port>* | Specifies a local "dynamic" application-level port forwarding. Currently the SOCKS4 protocol is supported, and `ssh` will act as a SOCKS4 server. Dynamic port forwardings can also be specified in the configuration file. |
| `-F` *<configfile>* | Specifies an alternative per-user configuration file. If a configuration file is given on the command line, the systemwide configuration file (`/etc/ssh_config`) is ignored. Default per-user configuration file is `$HOME/.ssh/config`. |

`-L <port>:<host>:<hostport>` Specifies that the given port on the client (local) host is to be forwarded to the given host and port on the remote side.

`-R <port>:<host>:<hostport>` Specifies that the given port on the remote (server) host is to be forwarded to the given host and port on the local side.

# strings

`strings` Finds the printable strings in an object or binary file.

```
strings [-] [-a] [-o] [-<number>] [<file> ...]
```

`strings` looks for ASCII strings in binary files or standard input. `strings` is useful for identifying random object files and many other things. A string is any sequence of four (the default) or more printing characters ending with a newline or a `null`. Unless the `-` flag is given, `strings` looks in all sections of the object files except the (`_TEXT`, `_text`) section. If no files are specified, standard input is read.

`-` Looks for strings in all bytes of the files (the default for non-object files).

`-a` Looks for `strings` in all sections of the object file (including the (`_TEXT`, `_text`) section).

`-o` Writes each string preceded by its byte offset from the start of the file.

`-<number>` The decimal *<number>* is used as the minimum string length rather than the default of 4.

# su

`su` Substitute user identity.

```
su [-flm] [<login>] [-c <shell arguments>]
```

`su` requests the password for login and switches to that user and group ID after obtaining proper authentication. A shell is then executed, and any additional shell arguments after the login name are passed to the shell.

If `su` is executed with no username as an argument, root is assumed.

If `su` is executed by root, no password is requested and a shell with the appropriate user ID is executed.

`-c` Invokes the following command in a subshell as the specified user.

`-f` If the invoked shell is `csh(1)`, this option prevents it from reading the .cshrc file.

`-l` Simulates a full login. The environment is discarded except for HOME, SHELL, PATH, TERM, and USER. HOME and SHELL are modified as above. USER is set to the target login. PATH is set to `/bin:/usr/bin`. TERM is imported from your current environment. The invoked shell is the target login's, and `su` will change directory to the target login's home directory. The `-l` option is synonymous with `-`, as in `su -`.

-m Leaves the environment unmodified. The invoked shell is your login shell, and no directory changes are made. As a security precaution, if the target user's shell is a non-standard shell (not listed in /etc/shells) and the caller's real uid is non-zero (not root), su will fail.

The -l and -m options are mutually exclusive; the last one specified overrides any previous ones.

Only users in group wheel (normally gid 0) or group admin (normally gid 20) can su to "root".

## sudo

sudo Executes a command as another user.

```
sudo -V¦-h¦-l¦-L¦-v¦-k¦-K¦[-H][-P][-S][-b] ¦ [-p <prompt>] [-u <username>¦<#uid>]
<command>
```

```
sudo -V¦-h¦-l¦-L¦-v¦-k¦-K¦[-H][-P][-S][-b] ¦ [-p <prompt>] [-u <username>¦<#uid>] -s
```

sudo allows a permitted user to execute a <command> as root or another user, as specified in /etc/sudoers. The real and effective uid and gid are set to match those of the target user as specified in the passwd file or Netinfo map. By default, sudo requires that users authenticate themselves with a password. (Note: By default, this is the user's password, not the root password.) Once a user has been authenticated, a time stamp is updated and the user may then use sudo without a password for a short period of time after the time stamp (five minutes unless overridden in sudoers). The time stamp is updated every time a command is executed through sudo, providing a sliding-window during which the user may use commands as the alternate user without re-entering the required password.

sudo determines who is an authorized user by consulting the file /etc/sudoers. By giving sudo the -v flag, a user can update the time stamp without running a command.

If a user who is not listed in /etc/sudoers tries to run a command via sudo, mail is sent to the proper authorities, as defined at configure time or /etc/sudoers. Note that the mail will not be sent if an unauthorized user tries to run sudo with the -l or -v flags. This allows users to determine for themselves whether or not they are allowed to use sudo.

sudo can log attempts as well as errors to syslog(3), a log file, or both. By default, sudo will log via syslog(3).

When used with the -s option instead of a <command>, sudo executes the target user's shell in a manner similar to the su command. The changing of the effective user and executing of the shell are logged, but commands executed while in that shell are not recorded.

-V Causes sudo to print the version number and exit. If the invoking user is root, the -V option will print out a list of the defaults sudo was compiled with.

-l Lists out the allowed (and forbidden) commands for the user on the current host.

-L Lists out the parameters that may be set in a Defaults line, along with a short description for each. This option is useful in conjunction with grep.

-h Causes sudo to print a usage message and exit.

-v Update the user's time stamp, prompting for the user's password if necessary. This extends the sudo timeout for another five minutes (or whatever the timeout is set to in sudoers).

`-k` — Invalidates the user's time stamp by setting the time on it to the epoch. The next time `sudo` is run, a password will be required.

This option does not require a password and was added to allow a user to revoke sudo permissions from a `.logout` file.

`-K` — Removes the user's time stamp entirely. Like `-k`, this option does not require a password.

`-b` — Tells `sudo` to run the given command in the background. Note that if you use the `-b` option, you cannot use shell job control to manipulate the process.

`-p` — Allows you to override the default password prompt and use a custom one. If the password prompt contains the `%u` escape, `%u` will be replaced with the user's login name. Similarly, `%h` will be replaced with the local hostname.

`-S` — Causes `sudo` to read the password from standard input instead of the terminal device.

`-P` — Preserves the calling user's group vector unaltered. By default, `sudo` will initialize the group vector to the list of groups the target user is in. The real and effective group IDs, however, are still set to match the target user.

`-H` — Sets the `$HOME` environment variable to the homedir of the target user (root by default) as specified in `/etc/passwd` or Netinfo. By default, `sudo` does not modify `$HOME`.

`sudo` tries to be safe when executing commands. To accomplish this, most shell variables specifying load paths for dynamically loaded libraries, user paths, and similar routes by which commands may be spoofed are ignored when searching for commands and when loading dynamic modules. This will not affect general use of the `sudo` command, but may result in unexpected behavior in some situations.

## SystemStarter

`SystemStarter` — Starts, stops, and restarts system services.

```
SystemStarter [-gvxdDqn] [<action> [<service>]]
```

The `SystemStarter` utility may be used to start, stop, and restart the system services that are described in the `/Library/StartupItems/` and `/System/Library/StartupItems/` paths.

The optional `<action>` argument specifies which action `SystemStarter` performs on the startup items. The optional `<service>` argument specifies which startup items to perform the action on. If no service is specified, all startup items are acted on; otherwise, only the item providing the service, any items it requires, or any items that depend on it will be acted on.

During boot, `SystemStarter` is invoked by `rc`(8) and is responsible for starting all startup items in an order that satisfies each item's requirements.

**Actions**

start — Starts all items or starts the item that provides the specified `<service>` and all items providing services it requires.

stop — Stops all items or stops the item that provides the specified `<service>` and all items that depend on it.

restart — Restarts all items or restarts the item providing the specified `<service>`.

**Options**

`-g` — Graphical startup

`-v` — Verbose (text mode) startup

`-x` — Safe mode startup (only runs Apple-provided items)

`-d` — Prints debugging output

`-D` — Prints debugging output and dependencies

`-q` — Quiet (disables debugging output)

`-n` — Doesn’t actually perform action on items (no-run mode)

## tail

`tail` — Displays the last part of a file.

```
tail [-f ¦ -F ¦ -r] [-b <number> ¦ -c <number> ¦ -n <number>] <file>
tail [-f ¦ -F ¦ -r] [-b <number> ¦ -c <number> ¦ -n <number>]
```

`-f` — Waits for and displays additional data that *`<file>`* receives, rather than stopping at the end of the file.

`-F` — Similar to `-f`, except that every five seconds, `tail` checks whether *`<file>`* has been shortened or moved. If so, `tail` closes the current file, opens the filename given, displays its entire contents, and waits for more data. This option is especially useful for monitoring log files that undergo rotation.

`-r` — Displays the file in reverse order, by line. The default is to display the entire file in reverse. This option also modifies the `-b`, `-c`, and `-n` options to specify the number of units to be displayed, rather than the number of units to display from the beginning or end of the input.

`-b <number>` — Specifies location in number of 512-byte blocks.

`-c <number>` — Specifies location in number of bytes.

`-n <number>` — Specifies location in number of lines.

# (gnu)tar

| | |
|---|---|
| `gnutar` | Creates, extracts, or appends to tape archives. |

```
tar [-] < A¦c¦d¦r¦t¦u¦x> [options] <file1> <file2> .. [-C <directory>]
```

`tar` saves files to and restores files from a single file. Although that single file might have originally been intended to be magnetic tape, magnetic tape is not required.

One of the following flags is required:

| | |
|---|---|
| `-A, --catenate, --concatenate` | Appends the contents of named file, which must itself be a gnutar archive, to the end of the archive (erasing the old end-of-archive block). |
| `-c, --create` | Creates a new archive or overwrites an existing one. |
| `-d, --diff, --compare` | Finds differences between files in the archive and corresponding files in the file system. |
| `--delete` | Deletes named files from the archive. |
| `-r, --append` | Appends the specified files to an archive. This works only on media on which an end-of-file mark can be overwritten. |
| `-t, --list` | Lists the contents of an archive. If any files are listed on the command line, only those files are listed. |
| `-u, --update` | Appends the named files if the on-disk version has a modification date more recent than their copy in the archive. |
| `-x, --extract, --get` | Extracts files from an archive. If any files are listed on the command line, only those files are extracted. If more than one copy of a file exists in an archive, earlier copies are overwritten by later copies. |
| `-r` | Appends the specified files to an archive. This works only on media on which an end-of-file mark can be overwritten. |
| `-u` | Alias to `-r`. |

In addition to the required flags, these are some common options may be used (see the manpage for a full list):

| | |
|---|---|
| `-f` *`<archive>`* | Filename where the archive is stored. |
| `-z` | Filters the archive through `gzip` |
| `-Z` | Filters the archive through `compress`. |
| `-j` | Filters the archive through `bzip2` |
| `-m` | Does not preserve modification time. |
| `-p` | Preserves user ID, group ID, file mode, and access and modification times. |
| `-v` | Verbose mode. |
| `-h` | Follows symbolic links as if they were normal files or directories. |
| `-P` | Does not follow any symbolic links. |
| `-X` | Does not cross mount points in the file system. |

`[-014578]`

Selects a backup device, `/dev/rmtN`, where *N* is the argument given.

`-C <directory>`

Sets the working directory for the files. When extracting, files are extracted into the specified directory. When creating, specified files are matched from the directory.

`-s <replstr>`

Modifies the filenames or archive member names specified by the pattern or file operands according to the substitution expression `<replstr>`, using the syntax of `ed(1)` in this format:

`/old/new/π`

*old* is the old expression. *new* is the new expression.

The optional trailing `g` applies the substitution globally. That is, it continues to apply the substitution. The first unsuccessful substitution stops the `g` option.

The optional trailing `p` causes the final result of a successful substitution to be written to standard error in this format:

`<original pathname> >> <new pathname>`

Multiple `-s <replstr>` options can be specified. They are applied in the order listed.

# top

`top`

Displays system usage statistics.

```
top [-u] [-w] [-k] [-s <interval>] [-e ¦ -d ¦ -a] [-l <samples>] [<number>]
```

`top`

`-u`

Sorts by CPU usage and displays usage starting with the highest usage.

`-w`

Generates additional columns of output data. The additional columns include `VPRVT` and the delta information for `#PRTS`, `RSHRD`, `RSIZE`, and `VSIZE`.

`-k`

Causes `top` to traverse and report the memory object map for pid 0 (kernel task). This option is optional because it is expensive to traverse the object maps, as the kernel task may have a large number of entries.

`-s <interval>`

Samples processes at the specified `<interval>`. Default is one-second intervals.

`-e`

Switches to event-counting mode in which counts reported are absolute counters. Options `-w` and `-k` are ignored.

`-d`

Switches to an event-counting mode in which counts are reported as deltas relative to the previous sample. Options `-w` and `-k` are ignored.

`-a` — Switches to an event-counting mode in which counts are reported as cumulative counters relative to when `top` was launched. Options `-w` and `-k` are ignored.

`-l <`*`samples`*`>` — Switches from default screen mode to a logging mode suitable for saving the output to a file. If `<`*`samples`*`>` is specified, `top` samples the number of samples specified before exiting. The default is `1`.

`<`*`number`*`>` — Limits the number of processes displayed to `<`*`number`*`>`.

Pressing the `Q` key causes `top` to exit immediately.

Columns displayed in default data mode:

| | |
|---|---|
| `PID` | Unix process ID |
| `COMMAND` | Unix command name |
| `%CPU` | Percentage of CPU used (kernel and user) |
| `TIME` | Absolute CPU consumption (min:secs.hundredths) |
| `#TH` | Number of threads |
| `#PRTS` (delta) | Number of MACH ports |
| `#MERG` | Number of memory regions |
| `VPRVT` (`-w` only) | Private address space currently allocated |
| `RPRVT` (delta) | Resident shared memory (as represented by the resident page count of each shared memory object) |
| `RSHRD` (delta) | Total resident memory (real pages that this process currently has associated with it; some may be shared by other processes) |
| `VSIZE` (delta) | Total address space currently allocated (including shared) |

Columns displayed in event-counting modes:

| | |
|---|---|
| `PID` | Unix process ID |
| `COMMAND` | Unix command name |
| `%CPU` | Percentage of CPU used (kernel and user) |
| `TIME` | Absolute CPU consumption (min:secs.hundredths) |
| `FAULTS` | Number of page faults |
| `PAGEINS` | Number of requests for pages from a pager |
| `COW_FAULTS` | Number of faults that caused a page to be copied |
| `MSGS_SENT` | Number of mach messages sent by the process |
| `MSGS_RCVD` | Number of mach messages received by the process |
| `BSDSYSCALL` | Number of BSD system calls made by the process |
| `MACHSYSCALL` | Number of MACH system calls made by the process |
| `CSWITCH` | Number of context switches to this process |

# traceroute

`traceroute`

Prints the route packets take to a network host.

```
traceroute [-d] [-m <max_ttl>] [-n] [-p <port>] [-q <nqueries>] [-r] [-s <src_addr>]
[-t <tos>] [-w <waittime>] <host> [<packetsize>]
```

`traceroute` uses the IP protocol time-to-live field and attempts to elicit an `ICMP TIME_EXCEEDED` response from each gateway along the path to the same host.

The only mandatory parameter is *<host>*, the destination host or IP number. The default probe datagram length is 38 bytes, but this can be increased by specifying a packet size (in bytes) after the destination hostname.

`-d`

Turns on socket-level debugging.

`-m` *<max_ttl>*

Sets the maximum time-to-live (maximum number of hops) used in outgoing probe packets. The default is 30 hops. The same default is used for TCP connections.

`-n`

Prints hop addresses numerically rather than symbolically and numerically (saves a nameserver address-to-name lookup for each gateway found on the path).

`-p` *<port>*

Sets the base UDP port number used in probes to *<port>*. Default is 33434. `traceroute` hopes that nothing is listening on UDP *<base>* to *<base+nohops+1>* at the destination host so that an `ICMP PORT_UNREACHABLE` message will be returned to terminate the route tracing. If something is listening on a port in the default range, this option can be used to pick an unused port range.

`-q` *<nqueries>*

Sets the number of probes per ttl to *<nqueries>*. Default is three probes.

`-r`

Bypasses the normal routing tables and sends directly to a host on an attached network. If the host is not on a directly attached network, an error is returned. This option can be used to ping a local host through an interface that has no route through it.

`-s` *<src_addr>*

Uses the following IP address (which must be given as an IP number, not a hostname) as the source address in outgoing probe packets. On hosts with more than one IP address, this option can be used to force the source address to be something other than the IP address of the interface that the probe packet is sent on. If the IP address is not one of this machine's interfaces, an error is returned and nothing is sent.

`-t` *<tos>*

Sets the type-of-service in probe packets to *<tos>*. Default is 0. Value must be a decimal integer in the range 0 to 255. This option can be used to see whether different types-of-service result in different paths. Not all values of TOS are legal or meaningful. See the IP spec for definitions. Useful values are probably `-t 16` (low delay) and `-t 8` (high throughput).

`-v` Sets to verbose output. Lists ICMP packets received other than `TIME_EXCEEDED` and `UNREACHABLE` packets.

`-w <`*`waittime`*`>` Sets the time to wait for a response to a probe to <*waittime*> seconds. Default is three seconds.

## umount

`umount` Unmounts file systems.

`umount [-fv] <`*`special`*`> ¦ <`*`node`*`>`

`umount -a ¦ -A [-fv] [-h <`*`host`*`>] [-t <`*`type`*`>]`

`-f` Forcibly unmounts the file system. Active special devices continue to work, but all other files return errors if further accesses are attempted. The root file system cannot be forcibly unmounted.

`-v` Enables verbose mode.

`-a` All the file systems described in fstab (5) are unmounted.

`-A` All the currently mounted file systems except the root are unmounted.

`-h <`*`host`*`>` Unmounts only file systems mounted from the specified <*host*>. This option implies the `-A` option and, unless otherwise specified with the `-t` option, will only unmount NFS file systems.

`-t <`*`type`*`>_` Is used to indicate that actions should only be taken on file systems of the specified <*type*>. More than one type may be specified in a comma-separated list. The list of filesystem types can be prefixed with `no` to specify the filesystem types for which action should not be taken. For example, the `umount` command: `umount -a -t nfs,mfs` unmounts all file systems of the type NFS and MFS.

## uptime

`uptime` Shows how long the system has been running.

`uptime`

`uptime` displays the current time, the length of time the system has been up, the number of users, and the load average of the system over the last 1, 5, and 15 minutes.

## uuencode, uudecode

| | |
|---|---|
| uuencode | Encodes a binary file. |
| uudecode | Decodes a binary file. |

uuencode [<*file*>] <*name*>

uudecode [<*file*> ...]

uuencode and uudecode are used to transmit binary files over transmission mediums that only support simple ASCII data.

uuencode reads <*file*> (or by default the standard input) and writes an encoded version to the standard output. The encoding uses only printing ASCII characters and includes the mode of the file and the operand <*name*> for use by uudecode.

uudecode transforms uuencoded files (or by default the standard input) into the original form. The resulting file is named <*name*> and has the mode of the original file except that setuid and execute bits are not retained. uudecode ignores any leading and trailing lines.

## w

| | |
|---|---|
| w | Displays who the present users are and what they are doing |

w [-dhin] [-M <*core*>] [-N <*system*>] [<*user*>]

w displays a summary of the current activity on the system, including what each user is doing. The first line displays the current time of day, how long the system has been running, the number of users logged in to the system, and the load averages. The load average numbers give the number of jobs in the run queue average over 1, 5, and 15 minutes.

The output fields are the user's login name, the name of the terminal where the user is logged on, the host from which the user is logged in, the time the user logged in, the time since the user last typed anything, and the name and arguments of the current process.

| | |
|---|---|
| -d | Dumps out the entire process list on a per controlling tty basis, instead of just the top level process. |
| -h | Suppresses the heading. |
| -i | Sorts output by idle time. |
| -n | Shows network addresses as numbers. Normally w interprets addresses and attempts to display them symbolically. |
| -M <*core*> | Extracts values associated with the name list from the specified core instead of the default /dev/kmem. |
| -N <*system*> | Extracts the name list from the specified system instead of the default /netbsd. |
| <*user*> | If specified, restricts output to <*user*>. |

# which

`which` Locates a program file, including `aliases` and `paths`. (`csh`(1)) only)

`which <name1> <name2> ...`

`which` displays the location of the specified commands, and displays which files would have been executed had the names been given as commands. Both aliases and paths are taken from the user's `.cshrc` file.

# who

`who` Displays who is logged in.

`who [-mTuH] [<file>]`

`who am i`

`who` displays a list of all users currently logged on, showing for each user the login name, tty name, the date and time of login, and hostname, if not local.

`-m` Only prints information about the current terminal (POSIX way of saying "Who am I?").

`-T` Prints a character after the username indicating the state of the terminal line: `+` if the terminal is writable; `-` if it is not writable; `?` if a bad line is encountered.

`-u` Prints the idle time for each user.

`-H` Writes column headings above the regular output.

`am i` Returns the invoker's real username.

`<file>` Gathers information from the specified `<file>`, rather than the default `/var/run/utmp`. An alternative `<file>` is usually `/var/log/wtmp`. The `wtmp` file contains a record of every login, logout, crash, shutdown and date change since `wtmp` was truncated or created. If `/var/log/wtmp` is being used as the file, the username may be empty or one of these special characters: `¦`, `}`, `~`. Logouts produce an output line without any username.

# whoami

`whoami` Displays the effective user ID.

`whoami`

`whoami` has been made obsolete by the `id`(1) utility, and is equivalent to `id -un`. The command `id -p` is suggested for normal interactive use.

`whoami` displays your effective user ID as a name.