FreeBSD command reference

Command structure

Each line you type at the Unix shell consists of a **command** optionally followed by some **arguments**, e.g.

```
ls -l /etc/passwd
 |  |      |
cmd arg1   arg2
```

Almost all commands are just programs in the filesystem, e.g. "ls" is actually `/bin/ls`. A few are built-in to the shell. All commands and filenames are *case-sensitive*.

Unless told otherwise, the command will run in the "foreground" - that is, you won't be returned to the shell prompt until it has finished. You can press Ctrl + C to terminate it.

Colour code

`command [args...]`	Command which *shows* information
`command [args...]`	Command which *modifies* your current session or system settings, but changes will be lost when you exit your shell or reboot
`command [args...]`	Command which *permanently affects* the state of your system

Getting out of trouble

`^C` *(Ctrl-C)*	Terminate the current command
`^U` *(Ctrl-U)*	Clear to start of line
`reset` `stty sane`	Reset terminal settings. If in xterm, try Ctrl+Middle mouse button and select "Do Full Reset"
`exit` `logout`	Exit from the shell
`ESC :q! ENTER`	Quit from **vi** without saving

Finding documentation

`man cmd`
`man 5 cmd`
`man -a cmd`
	Show manual page for command "cmd". If a page with the same name exists in multiple sections, you can give the section number, or **-a** to show pages from all sections.

`man -k str`	Search for string"str" in the manual index
`man hier`	Description of directory structure
`cd /usr/share/doc; ls` `cd /usr/share/examples; ls`	Browse system documentation and examples. Note especially **/usr/share/doc/en/books/handbook/index.html**
`cd /usr/local/share/doc; ls` `cd /usr/local/share/examples`	Browse package documentation and examples
`On the web:` **`www.freebsd.org`**	Includes handbook, searchable mailing list archives

System status

`Alt-F1 ... Alt-F8`	Switch between virtual consoles
`date`	Show current date and time
`ntpdate -b serv1 serv2 ...`	Synchronise clock to given NTP server(s)
`uptime`	Display time since last reboot and load stats
`w`	Show who is currently logged in
`last -10`	Show last 10 logins

Directories

`pwd`	Show current directory ("print working directory")
`cd subdir`	Move into a subdirectory of the current directory
`cd ..`	Move up one level, to the parent directory

`cd /`
`cd /absolute/path`
`cd ~username`
`cd`
	Change current directory: to the filesystem root, to an absolute location, to a particular user's home directory, or to your own home directory

`ls` `ls path`	List contents of current directory or given directory
`ls -l`	List directory in **long** form *(lowercase 'L', not number one)*
`ls -a`	List **all** files, including hidden files
`ls -d`	List **directory** itself, rather than its contents
`ls -ld path`	Example of combining flags
`mkdir path`	Create a directory
`rmdir path`	Delete an empty directory
`rm -rf subdir`	Recursively delete a directory and all its contents - *DANGEROUS!*

Files

`file filename`	Read first few bytes of file and guess its type
`less filename`	Read contents of file in pager. space = next page, b = previous page, q = quit / = search forward, ? = search backwards, n = repeat search
`less -Mi filename`	**-M** = show filename, **-i** = case-insensitive searching
`grep [-i] pattern filename`	Show all lines which contain the given pattern; **-i** = case-insensitive
`wc -l filename`	Count lines in file *(lowercase 'L', not one)*
`head -num filename` `tail -num filename`	Show first/last *num* lines of file; defaults to 10 lines
`tail -f filename`	Show last 10 lines of file then wait and show new lines as they are added (^C to exit). Especially useful for log files.
`strings filename | less`	Extract printable text strings from a binary file
`touch filename`	Create file if it does not exist, or update its timestamp
`rm filename`	Delete (remove) file
`cp filename newname`	Copy one file
`cp file1 file2 ... subdir/`	Copy a file or files into another directory. (The trailing slash on the subdir is not essential, but prevents errors when you are copying one file and 'subdir' does not exist)
`mv oldname newname`	Rename one file or directory
`mv file1 file2 ... subdir/`	Move a file or files into another directory
`ln filename newname`	Make a *hard link* from file to newname (both names point to the same filesystem inode). Both names must be on same filesystem.
`ln -s path newname`	Make newname a *symbolic* or *soft link* pointing to *path*, which may be a file or directory and can be anywhere on the filesystem.

Searching for files

`locate str`	Search for filenames matching *str* in the locate database
`/etc/periodic/weekly/310.locate`	Rebuild the locate database
`find path -type f`	Find all files under the given path (use "." for current directory)
`find path -type f -name 'foo*'`	Find all files under the given path whose name begins "foo"
`find path -type f | xargs cmd`	Find all files under path and apply *cmd* to each of them
`find path -type f -print0 | xargs -0 cmd`	Safer version of above (works with filenames that contain spaces)

Compressed files and archives

`gzip -dc filename.gz | less` `bzip2 -dc filename.bz2 | less`	Read compressed text file, without uncompressing it on disk
`tar -tzf filename.tgz` *or .tar.gz* `tar -tjf filename.tbz2` *or .tar.bz2*	Show contents of compressed tar archive. Add **-v** for more detail
`tar -xvzf [-C dir] filename.tgz` `tar -xvjf [-C dir] filename.tbz2`	Extract contents of compressed archive [into specified directory, otherwise into current directory]
`nroff -mandoc foo.1 | less`	Format a man page file

Processes

`ps auxw`	Show all processes
`ps auxw | grep procname`	Show all processes matching pattern "procname" (note that "grep procname" itself may be shown)
`top`	Show continuously the most active processes (**q** to quit)
`kill pid` `kill -TERM pid`	Send a 'terminate' signal to the given process: requests process to clean up quickly and exit
`kill -1 pid` `kill -HUP pid`	Send a 'hangup' signal to the given process: some processes use this as a request to re-read their config files. *(one, not letter 'L')*
`kill -9 pid` `kill -KILL pid`	Send a 'kill' signal to the given process: the process is killed immediately and cannot clean up first. Use only as a last resort.
`killall [-1|-9] procname`	Send signal to all processes whose name is "procname"

Account customisations

`~/.profile`	`EDITOR=joe; export EDITOR` `PAGER=less; export PAGER`	Change your default editor and pager
`~/.bash_profile`	`. .profile` `PS1='[\u@\h \W]\$ '; export PS1`	bash prompt which displays your current username, host, and directory
`~/.netrc`	`default login ftp password user@site`	Make ftp client login automatically
`~/.xinitrc`	`exec startkde`	Choose 'kde' desktop

X Window System

`startx`	Start graphical environment
Ctrl-Alt-F1 ... Alt-F9	Switch to text console while in X;return to X
Ctrl-Alt-Backspace	Emergency exit from X
`xterm -sb -sl 500 -ls`	Run xterm with 500 lines of scrollback (much better than Konsole)
`xset b off`	Disable terminal beep in Xenvironment

Shell facilities

`which foo`	Search for command *foo* in PATH and show where it was found
`history 20`	Display the 20 most recently entered commands
`!num`	Re-execute command *num* from history
`cmd1; cmd2`	Run *cmd1* followed by *cmd2*
`cmd1 && cmd2`	Run *cmd1*, then *cmd2* only if *cmd1* was successful ($? = 0)

Argument expansion

`~/file` `~user/file`	Expands to `/home/yourname/file` or `/home/user/file`
`/somepath/*.txt`	Expands to all filenames matching that pattern. *matches any characters; ? matches any one char; [abc] matches only those characters; [a-z] matches any in that range.
`$var`	Substitute value of environment variable 'var'

The special meaning of characters (including space which normally separates arguments) can be removed by preceeding them with a backslash; or by "quoting" or 'quoting' the whole argument. See *man sh* or *man csh*.

Environment

`printenv`		Show all environment variables
`printenv PATH` `echo $PATH`		Show single environment variable *`PATH'*
`foo="value"; export foo` `setenv foo "value"`	*[sh]* *[csh]*	Set environment variable *`foo'*
`unset foo` `unsetenv foo`	*[sh]* *[csh]*	Unset environment variable *`foo'*

Environment variables can be set at login time in **~/.profile** [sh], **~/.bash_profile** [bash], or **~/.cshrc** [csh]

File redirection

`^D` *(Ctrl-D)*		Send end-of-file on standard input
`cmd1 | cmd2`		Pipe output of *cmd1* to input of *cmd2*
cmd `>out.txt`		Redirect command standard output to file
cmd `2>err.txt`	*[sh]*	Redirect command error output to file
cmd `>out.txt 2>&1` *cmd* `>&out.txt`	*[sh]* *[csh]*	Redirect both standard and error output to file
cmd `>>out.txt`		Append to out.txt instead of replacing it
cmd `<in.txt`		Redirect command standard input from file

Job control

`^C` *(Ctrl-C)*	Terminate current foreground process
`^Z` *(Ctrl-Z)*	Suspend current foreground process (makes suspended job)
`jobs`	List jobs under this shell
`kill %n`	Terminate job number *n*
`fg` `fg %n`	Restart suspended process in foreground
`bg` `bg %n`	Restart suspended process in background
cmd `&`	Start command as background job

'vi' editor

This is the standard Unix editor and is always available. You must be *extremely* careful though, because the effect of hitting a key will depend on what mode you are in at that time. If in any doubt, hit ESC to get back to command mode, then enter one of the commands shown here.

:q! *[Enter]*	Quit without saving
:wq *[Enter]*	Write and quit
:wq! *[Enter]*	Write and quit, forcing overwrite of read-only file
:w filename *[Enter]*	Write out to a different file
^L *(Ctrl-L)*	Redraw screen
^	Move to start of line
$	Move to end of line
h j k l	Move cursor left / down / up / right (alternative to cursor keys)
:num *[Enter]*	Go to line number *num*
G	Go to last line
/pattern *[Enter]*	Search forwards for pattern
?pattern *[Enter]*	Search backwards for pattern
n	Repeat last search
i text ESC	Insert text before cursor position
A text ESC	Append text after end of line
o text ESC	Open new line after current one and insert text
x	Delete character under cursor
r char	Replace character under cursor with another single character
dd	Delete entire line
yy	Copy current line ("yank")
num yy	Copy *num* lines, starting with the current line
p	Paste copy buffer *after* current line

'ee' editor

This is a simpler alternative to 'vi' and is installed as part of the FreeBSD base system. However it may not always be available (there is **/rescue/vi** for emergencies when /usr is not mounted, but no emergency 'ee').

You don't need to remember anything in this table; all commands are described on-screen.

ESC	Pop-up menu
^C	Command prompt
^C quit *[Enter]*	Quit without saving
^C exit *[Enter]*	Write and quit
^C write *[Enter]*	Write out to a different file
^A	Move to start of line
^E	Move to end of line
^C num *[Enter]*	Go to line number *num*
^Y string *[Enter]*	Search forwards for string
^X	Repeat last search
^K	Delete entire line

'joe' editor

'joe' is a powerful editor and a lot more forgiving than 'vi', but needs to be installed as a separate package and may not always be available. You can get away with knowing only ^K X, and even that is shown in the on-screen help!

^K H *(Ctrl-K, H)*	Toggle help on/off
^C	Quit without saving
^K X	Write and quit
^K D	Write (optionally to a different filename) without quitting
^R	Redraw screen
^T T	Toggle insert/overwrite mode
^A	Move to start of line (or use 'Home')
^E	Move to end of line (or use 'End')
^K L num *[Enter]*	Go to line number *num*
^K V	Got to last line
^K F pattern *[Enter]*	Search for pattern; gives options for backwards and replace
^L	Repeat last search
^Y	Delete entire line
^_	Undo (on some terminals, Ctrl-Shift-Underscore is required)
^K B	Mark start of block
^K K	Mark end of block
^K C	Copy block to current cursor position
^K M	Move block to current cursor position
^K Y	Delete block
^K W	Write block to a file
^K R	Insert file at current cursor position

You can get alternative key bindings by invoking as 'jmacs', 'jstar' or 'jpico' which correspond to emacs, WordStar and pico respectively.

System Administration

User accounts

`id`	Show current uid, gid and supplementary groups
`whoami`	Show current username only
`su`	Change uid to root *(note: user must be in "wheel" group)*
`su username`	Change uid to *username*
`su -` `su - username`	As above, but also reinitialise environment as per a full login
`cat /etc/passwd`	Show all accounts
`cat /etc/group`	Show all groups
`pw useradd username -m`	Create user; **-m** = make home directory
`passwd` `passwd username`	Set or change password for self or for another account (root only)
`pw usermod username -G wheel`	Add user to "wheel" group (or just edit /etc/group directly)
`pw userdel username -r`	Delete user; **-r** = remove home directory and all its contents
`cat /etc/master.passwd`	Show all accounts (including encrypted passwords)
`vipw`	Lock master.passwd, edit it, and rebuild password databases

Filesystems

`mount`	Show mounted filesystems
`df` `df -h`	Show used and free space in all mounted filesystems *(-h = "human readable", e.g. shows 1G instead of 1048576)*
`du -c [path]`	Add up space used by files/directories under *path* (or current dir)
`mount -r -t cd9660 /dev/acd0 /cdrom`	Mount device **/dev/acd0** [IDE CD] on directory **/cdrom**; filesystem type is **cd9660; -r** = read-only.
`umount /cdrom`	Unmount device (must not be in use)
`mount -t msdos /dev/fd0 /mnt` `umount /mnt`	Similar for MS-DOS floppy disk
`fstat`	List processes with open files
`cat /etc/fstab`	Show filesystem table
`mount /cdrom`	Mount **/cdrom** using parameters from **/etc/fstab**
`mount -a`	Mount all filesystems in **/etc/fstab** except those labelled "noauto" (this is done at normal bootup, but is useful when booting into single-user mode)
`fsck -y /dev/ad0s1d`	Repair UFS filesystem on **/dev/ad0s1d**. *NOTE: must be unmounted or mounted read-only*

Slices and Partitions

`fdisk /dev/ad0`	Show slices ("partitions" in DOS terminology) on device
`disklabel /dev/ad0s1`	Show FreeBSD partitions within a slice
`/stand/sysinstall`	Has options for partitioning and slicing, should you need to add another disk to an already-installed FreeBSD system
`iostat 2` `gstat -I 2s`	Show disk I/O statistics every 2 seconds

Packages

`pkg_info`	Show summary list of installed packages
`pkg_info foo-1.2.3` `pkg_info foo\*`	Show detailled description of package "foo"
`pkg_info -L foo\*`	Show list of files included in package "foo"
`pkg_info -W /usr/local/bin/foo`	Find which package contains file **/usr/local/bin/foo**
`pkg_add foo-1.2.3.tbz`	Install package from file
`pkg_add -r foo`	Install package from default FTP server
`PACKAGEROOT="ftp://ftp.uk.freebsd.org" pkg_add -r foo`	Install package from alternative FTP server
`pkg_delete foo-1.2.3`	Uninstall package
`rehash` *[csh]*	After installing a package, rescan PATH for new executables. *(Only needed if you are using csh)*

Kernel modules

`kldstat`	Show loaded modules
`kldload if_wi`	Load named module and any modules it depends on
`kldunload if_wi`	Unload module

Networking

`ifconfig -a`	Show all interfaces
`ifconfig fxp0 192.168.0.1/24`	Configure an interface
`netstat -r -n`	Show forwarding table (routes)
`route add default 192.168.0.254`	Add static default route
`ping 1.2.3.4`	Send test packets, display responses (^C to exit)
`traceroute -n 1.2.3.4`	Send test packets and display intermediate routers found
`tcpdump -i fxp0 -n -s1500 -X` `tcpdump -i fxp0 -n tcp port 80`	Show entire packets sent and received on given interface; second form shows only packet headers to/from TCP port 80
`telnet 1.2.3.4 80`	Open TCP connection to port 80 on host 1.2.3.4
`vi /etc/rc.conf` `vi /etc/resolv.conf`	Edit startup configuration file, DNS resolver configuration file (see "Important Configuration Files")
`/etc/rc.d/netif start`	Initialise network interfaces from settings in **/etc/rc.conf**
`/etc/rc.d/routing start`	Initialise static routes from settings in **/etc/rc.conf**
`/etc/rc.d/dhclient start`	Configure interfaces marked "DHCP" in **/etc/rc.conf**
`netstat -finet -n`	Show active network connections [add **-a** for listening sockets]
`sockstat -4 -l`	Show processes listening on IPv4 sockets

Shutdown

`reboot`	Reboot immediately
`halt`	Shutdown immediately
`halt -p`	Shutdown immediately and turn off power if possible
`shutdown -h 5 "Sys maintenance"`	Halt in 5 minutes, send warning message to logged-in users

Important Configuration Files

Many of these are documented in section 5 of the manual. e.g. "man 5 crontab"

/etc/crontab	Regular scheduled tasks
/etc/group	Binds supplementary groups to users (won't take effect until they next login)
/etc/hosts	Local mappings between IP addresses and hostnames
/etc/inetd.conf	Controls services started from inet, but which don't have their own daemon processes. e.g. ftpd
/etc/localtime	(Binary file, not editable) describes the current time zone `# cp /usr/share/zoneinfo/Africa/Maputo /etc/localtime`
/etc/mail/mailer.conf	Configures which MTA is used when local processes generate mail
/etc/make.conf	Defaults for when building software applications/ports
/etc/motd	"Message of the day" displayed on login
/etc/newsyslog.conf	Configures automatic rotation of log files
/etc/periodic/...	Various scripts which are run at scheduled times

/etc/rc.conf	Master configuration file. See **/etc/defaults/rc.conf** for allowable settings (but don't edit them there, because changes will be lost on upgrade)
	`# Network settings`
	`hostname="foo.example.com"`
	`ifconfig_fxp0="192.168.0.1/24"  # or "DHCP"`
	`defaultrouter="192.168.0.254"`
	`# Set clock at bootup`
	`ntpdate_enable="YES"`
	`ntpdate_flags="-b ntp-1.example.net ntp-2.example.net"`
	`# Enable services`
	`inetd_enable="YES"`
	`sshd_enable="YES"`

/etc/rc.d/...	Startup scripts, run as `/etc/rc.d/foo start` or `/etc/rc.d/foo stop` Will not work unless the relevant service_enable="YES" exists in /etc/rc.conf
/etc/rc.local	Create this script to perform additional commands at system startup

/etc/resolv.conf	Configure DNS client
	`search example.com`
	`nameserver 192.0.2.1`
	`nameserver 192.0.2.2`

/etc/ssh/sshd_config	Configure ssh daemon (e.g. permit or refuse root logins)
/etc/sysctl.conf	Set run-time kernel variables at bootup, e.g. `net.inet.ip.forwarding=1  # if this machine is a router`
/etc/syslog.conf	Configure destinations of log messages. After changing: `# killall -1 syslogd`
/etc/ttys	Configure logins on serial lines or modems

/etc/X11/xorg.conf	XWindow server (display) configuration. To create:
	`# Xorg -configure`
	`# mv /root/xorg.conf.new /etc/X11/xorg.conf`

/usr/local/etc/...	Configuration files for third-party programs (ports/packages)
/usr/share/skel/...	Skeleton files which populate a new user's home directory
~/.ssh/authorized_keys	Public keys corresponding to the private keys which are permitted to login to this account using SSH RSA/DSA authentication

Other important files and directories

/boot/kernel/kernel /boot/kernel/*.so	The kernel itself, and its loadable modules

/boot/loader.conf	Kernel configuration at startup time. See /boot/defaults/loader.conf and /usr/src/sys/i386/conf/GENERIC.hints
	`hint.acpi.0.disabled=1  # disable ACPI`
	`if_wi_load="YES"  # load the 'wi' network driver`
	`snd_driver_load="YES"  # load all sound drivers`

/dev/null	The "bit bucket". To discard all output from a command (stdout and stderr): `# somecommand >/dev/null 2>&1  [sh]`
/rescue/...	Statically-linked binaries for use in emergencies
/root	Home directory for 'root' user (so it's still available when other filesystems are not mounted)
/stand/sysinstall	Run this to re-enter the installation menu
/usr/src/sys/i386/conf/MYKERNEL	Configuration file to build kernel "MYKERNEL" (see "GENERIC" for the default kernel which comes with FreeBSD)
/var/db/pkg/...	Where pkg_add records installed packages (don't alter them!)
/var/log/maillog	Mail log file
/var/log/messages	General system log file
/var/mail/username	Default location for user's mailbox
/var/run/inetd.pid	File containing process ID of running 'inetd' daemon
/var/spool/mqueue/...	Sendmail queued messages
/var/tmp	Temporary files; applications should write large files here rather than /tmp as it's usually on a larger filesystem

File permissions

`ls -l filename`
`ls -ld directory`
	Show permissions on file or directory.
	`  ,------- type (-=file, d=directory)`
	`  / ,------ rwx perms for user (owner)`
	`  / /  ,---- rwx perms for group`
	`/ /  /  ,-- rwx perms for other`
	`-rwxrwxrwx`
	For a file: **r** allows read; **w** allows write/append; **x** allows execute.
	For a directory: **r** allows listing contents; **w** allows creation or deletion of files within directory; **x** allows directory to be entered

`chown user path`
`chgrp group path`
`chown user:group path`
	Change the owner, group, or both, of a file or directory.

`chmod [ugoa]+[rwx] path` `chmod [ugoa]-[rwx] path`	Add or remove permission mode bits. **u** = user (owner), **g** = group, **o** = other, **a** = all (ugo) e.g. "chmod go+r *file*" adds the 'r' permission to group and other.

`chmod nnn path`	Change all the mode bits at once to octal value *nnn*.
	e.g. "chmod 640 *file*" sets rw- for user, r-- for group, --- for other.
	`0 ---  1 --x  2 -w-  3 -wx`
	`4 r--  5 r-x  6 rw-  7 rwx`

umask `umask nnn`	Show or set the file creation mask for this session; these are the permission bits which will *not* be set on newly-created files. For example, "umask 022" means that newly-created files have no more than rwxr-xr-x permissions.