

# Commonly used terminal commands that span nearly all Linux distributions

## | A |

**alias** – Create an alias, aliases allow a string to be substituted for a word when it is used as the first word of a simple command. This is a built in bash command.

example: user1@foo ]\$ alias la='ls -lAXh --color=always|less -R' (Now issuing the command 'la' will actually run a long listing, in color, sorted by extension.) Also see: **unalias**

**apropos** – Apropos searches a set of database files containing short descriptions of system commands for keywords and displays the result on the standard output.

example: user1@foo ]\$ apropos search (this command displays a list of possible commands and their respective descriptions with the string "search" in the command or in the command description.)

## | B |

**bcwipe** – Bcwipe repeatedly overwrites special patterns to the files to be destroyed. In normal mode, 35 passes are used (of which 8 are random). Patterns used were recommended in article by Peter Gutmann ([http://www.cs.auckland.ac.nz/~pgut001/pubs/secure\\_del.html](http://www.cs.auckland.ac.nz/~pgut001/pubs/secure_del.html)) entitled "secure deletion of data from magnetic and solid-state memory". In quick mode, U.S. DoD(Department of Defence) 5200.28 standard are used with 7 pass wiping. In custom mode U.S. DoD 5200.28 standard are used with user defined number of passes.

example: user1@foo ]\$ bcwipe -ng -r -v /home/temp (this command will do a 35 pass; recursive; verbose DoD wipe on /home/temp. Usually, bcwipe is used on an entire drive or partition where sensitive or personal data has been kept to prevent recovery of the data if the drive or system is being sold or given to an unknown party.)

## | C |

**cd** – Change directory, used to navigating directory structures via the command line.

example: user1@foo ]\$ cd /etc/X11 (this command changes the current working directory (cwd) to /etc/X11)

**chkconfig** – Updates and queries runlevel information for system services.

example: user1@foo ]\$ chkconfig --level 2345 httpd on (the command adds the httpd daemon to start at runlevels 2,3,4 and 5.)

**chmod** – Change file mode bits.

example: user1@foo ]\$ chmod -Rf 700 /home (this command changes everything under /home and all subdirectories to rwx only for the owner of the file or directory or root.)

**chown** – Change file ownership and group.

example: user1@foo ]\$ chown -R user1:user1 /home/testing (this command changes the file ownership and group ownership of every subdirectory under /home/testing to user1.)

**cp** – Copy files, used for copying files to and from different directories.

example: user1@foo ]\$ cp ~/foo.txt /mnt/storage (this command copies the file "foo.txt" from /home/user1/foo.txt to /mnt/storage/foo.txt)

## | D |

**dd** – Copy a file, converting and formatting according to the specified options.

example: root@foo ]# dd if=/dev/dvd of=dvd\_foo.iso (this command creates an iso image from the device /dev/dvd and writes to a local file (dvd\_foo.iso) in the (cwd) current working directory.)

**df** – Displays total disk usage for mounted filesystems.

example: user1@foo ]\$ df -h (this command displays the used/free space ratio on all currently mounted filesystems.)

**diff** – In the simplest case, diff compares the contents of the two files from-file and to-file. A file name of – stands for text read from the standard input. As a special case, diff – – compares a copy of standard input to itself. If from-file is a directory and to-file is not, diff compares the file in from-file whose file name is that of to-file, and vice versa. The non-directory file must not be -. If both from-file and to-file are directories, diff compares corresponding files in both directories, in alphabetical order; this comparison is not recursive unless the -r or –recursive option is given. diff never compares the actual contents of a directory as if it were a file. The file that is fully specified may not be standard input, because standard input is nameless and the notion of “file with the same name” does not apply. diff options begin with -, so normally from-file and to-file may not begin

with -. However, — as an argument by itself treats the remaining arguments as file names even if they begin with -.

example: user1@foo ]\$ diff /var/log/messages /var/log/messages.saved (this command displays the differences between the 2 files.)

**dstat** – Displays a variety of realtime system stats.

example: user1@foo ]\$ dstat -f (this command displays realtime stats in the system console.)

**du** – Displays total disk usage for named directories.

example: user1@foo ]\$ du -chs /home (this command displays in user readable format the space usage for the directory /home)

## | E |

**exit** – Exits a login terminal or a session as a certain user.

example: root@foo ]# exit (this command will exit the current (# root) shell and drop the user back to the (\$ user) prompt, if logged in via su or su -. )

## | F |

**fdisk** – Fdisk is a powerful disk partitioning utility.

example: root@foo ]# fdisk -l (this command lists the currently detected partitions and partition types. Use the man page to understand the full scope of this very useful and powerful command.)

**find** – Find searches a given directory structure for a named string and displays pertinent results.

example: user1@foo ]\$ find ~ -name foo (this searches the user1's home and any subdirectories for any occurrence of “foo” and displays the results, if any.)

**fsck** – Fsck initiates filesystem integrity check and repair on unmounted filesystems.

example: root@foo ]# fsck -y /dev/hda1 (this command forces a filesystem check on the partition hda1 and fixes any errors encountered without prompting the user to “ok” any confirmation dialogs. use with caution and read the man page.)

**fuser** – Identify processes using files or sockets.

example: user1@foo ]\$ fuser -m /mnt/floppy (this command searches for processes using /mnt/floppy and displays the results.)

## | G |

**grep** – Grep searches the named input file(s) (or standard input if no files are named, or the file name – is given) for lines containing a match to the given pattern. By default, grep prints the matching lines.

example: user1@foo ]\$ rpm -qa | grep foo (this command searches the rpm database for a string “foo” and prints any matches it encounters.)

**groupadd** – Create a new group on the system.

example: user1@foo ]\$ groupadd mytestgroup (this command creates a group on the system called mytestgroup.)

## | H |

**head** – Output the first part of files to the terminal

example: user1@foo ]\$ head mytextfile.txt (this command outputs the the first N lines of the file mytextfile.txt. by default this is 10 lines.)

**hostname** – Hostname is the program that is used to either set or display the current host, domain or node name of the system. These names are used by many of the networking programs to identify the machine. The domain name is also used by NIS/YP.

example: user1@foo ]\$ hostname (this command displays the current hostname.)

## | I |

**ifconfig** – Ifconfig is used to configure the kernel-resident network interfaces. It is used at boot time to set up interfaces as necessary. After that, it is usually only needed when debugging or when system tuning is needed. If no arguments are given, ifconfig displays the status of the currently active interfaces. If a single interface argument is given, it displays the status of the given interface only; if a single -a argument is given, it displays the status of all interfaces, even those that are down. Otherwise, it configures an interface.

example: user1@foo ]\$ ifconfig (this command issued by itself will print all currently configured network interfaces.)

**ifdown** – Brings a network interface down (offline) manually.

example: user1@foo ]\$ ifdown eth0 (this command brings the network interface ‘eth0’ down or offline.)

**iftop** – Display bandwidth usage on an interface by host.

example: user1@foo ]\$ iftop -i wlan0 (this command listens on the wlan0 interface and displays information about traffic statistics.)

**ifup** – Brings a network interface up (online) manually.

example: user1@foo ]\$ ifup eth0 (this command brings the network interface ‘eth0’ up or online.)

**init** – Process control initialization.

example: user1@foo ]\$ init 3 (this command brings the system into init 3 text-only mode.)

**iptraf** – An interactive colorful IP LAN traffic monitor.

example: user1@foo ]\$ iptraf (this command starts iptraf in the terminal.)

## | J |

**jwhois** – Jwhois searches Whois servers for the object on the command line. The host to query is taken from a global configuration file, a configuration file specified on the command line, or selected directly on the command line.

example: user1@foo]\$ jwhois google.com (this command queries the whois database for the contact and domain registration details of google.com.)

## | K |

**kill** – Terminate a currently running or rogue process. also see: killall

example: user1@foo]\$ kill -9 27068 (this command terminates a process with a process id of 27058 found by using the command ps aux.)

## | L |

**last** – Lists users who have recently logged into or out of the system. Also displays time, date and duration.  
example: user1@foo]\$ last (this command lists all of the users who have recently logged into or out of the system.)

**less** – Less is a program similar to more, but which allows backward movement in the file as well as forward movement. Also, less does not have to read the entire input file before starting, so with large input files it starts up faster than text editors like vi. Less uses termcap (or terminfo on some systems), so it can run on a variety of

terminals. There is even limited support for hardcopy terminals. (on a hardcopy terminal, lines which should be printed at the top of the screen are prefixed with a caret.)

example: user1@foo]\$ less -p CPU /var/log/dmesg (this command searches /var/log/dmesg for any instance of the string 'CPU' and displays those lines with 'CPU' highlighted.)

**locate** – Locate, in combination with **updatedb** provides a secure way to index and quickly search for files on your system. It uses incremental encoding to compress its database to make searching faster, but it will also store file permissions and ownership so that users will not see files they do not have access to.

example: user1@foo]\$ locate gimp (this command searches a secure database starting at / for the given string. note: you must first run the command updatedb before using locate.)

**ls** – Lists files, directories and their respective permissions (depending on the switches and options used.)

example: user1@foo]\$ ls -als (this command lists all of the files and subdirectories in the current working directory (cwd) and

their respective permissions. to slow down long directory listings use: ls -als | less use q to exit.)

**lspci** – lspci is a utility for displaying information about all PCI buses in the system and all devices connected to them.)

example: user1@foo]\$ lspci -v (this command lists all of the pci devices detected by the system in verbose mode.)

**lsusb** – lsusb is a utility for displaying information about USB buses in the system and the devices connected to them. To use lsusb make sure you have usb-utils package installed.)

example: user1@foo]\$ lsusb -v (this command lists all of the usb devices detected by the system in verbose mode.)

## | M |

**man** – Man formats and displays the on-line manual pages. If you specify section, man only looks in that section of the manual. name is normally the name of the manual page, which is typically the name of a command, function, or file. However, if name contains a slash (/) then man interprets it as a file specification, so that you can do man ./foo.5 or even man /cd/foo/bar.1.gz.

example: user1@foo ]\$ man sudo (this command will display the sudo command man page and a verbose list of options and usage.)

**more** – File perusal filter for crt viewing.

example: user1@foo ]\$ more +3 myfile.txt (in this example we would see the results of myfile.txt displayed starting at line 3)

**mount** – Mount a filesystem. see also: umount.

example: root@foo ]# mount -t ext3 /dev/hdk3 /mnt/foo (this command mounts partition 3 on the hard disk /dev/hdk on /mnt/foo where the user can then access it with the proper rwx permissions. automounting filesystems can be taken care of in /etc/fstab)

**mv** – Move files from one directory to another. (similar to cut) this option deletes the file from the original location and moves it to another, it is also used to rename files.)

example: user1@foo ]\$ mv foo.rpm foo2.rpm (this command renames “foo.rpm” to “foo2.rpm” within the same directory.)

## | N |

**nano** – Nano is a small, free and friendly editor which aims to replace pico, the default editor included in the non-free pine package. Rather than just copying pico’s look and feel, nano also implements some missing (or disabled by default) features in pico, such as “search and replace” and “go to line number”.

example: root@foo ]# nano /var/log/dmesg (this command displays the text file /var/log/dmesg in a terminal window and presents it for editing or reading.)

## | O |

**octave** – A high-level interactive language for numerical computations.

example: root@foo ]# octave –help (this command will display the options for the octave command.)

## | P |

**passwd** – Passwd is used to update a user’s authentication token(s). Passwd is configured to work through the Linux-PAM API. Essentially, it initializes itself as a “passwd” service with Linux-PAM and utilizes configured password modules to authenticate and then update a user’s password..

example: user1@foo ]\$ passwd newuser (this command prompts the issuer to provide a new password for the given user and then updates the password database accordingly replacing the previous password with the new one.)

**pkill** – Look up or signal processes based on name and other attributes.

example: user1@foo ]\$ pkill -HUP syslogd (this command makes syslog re-read its configuration file.)

**ps** – Prints a list of currently running processes, their respective owners and process id’s.

example: user1@foo ]\$ ps aux (this command displays the currently running processes among other important data about the respective processes.)

**pwd** – Print the name of the working directory.

example: user1@foo ]\$ pwd (this command prints the current path/working directory in your shell, such as: /home/user1/foo)

## | Q |

**quota** – Displays a list of users’ disk usage and limits. By default only the user quotas are printed. Quota reports the quotas of all the filesystems listed in /etc/mtab. For filesystems that are NFS-mounted a call to

the `rpc.rquotad` on the server machine is performed to get the information.

example: `root@foo ]# quota user1` (this command displays the requested users disk quota usage and limits.)

## | R |

**rm** – Remove files and/or directories.

example: `user1@foo ]$ rm -rf ~/temp` (this command removes the directory `/home/user1/temp`)

**rsync** – A faster more flexible replacement for `rcp` (remote file copy)

example: `user1@foo ]$ rsync -avz -e ssh rsync@remote.acme.com:/home/rsync/out/  
/home/rsync/from_remote` (this command copies the contents of `/home/rsync/out/`  
`/home/rsync/from_remote` on a remote system.)

(An important thing here, is that the presence or absence of a trailing “/” in the source directory determines whether the directory itself is copied, or simply the contents of this source directory. Leaving out the trailing slash creates the new directory “out” on the remote machine.)

## | S |

**service** – Run a system V init script on the system.

example: `user1@foo ]$ service httpd restart` (this command restarts a running `httpd` daemon or starts an idle/stopped `httpd` daemon.)

**shutdown** – Bring the system down.

example: `user1@foo ]$ shutdown -r now` (this command shuts the system down immediately and reboots.)

**ssh** – Starts the secure shell connection with a system running the `sshd` (secure shell daemon).

example: `user1@foo ]$ ssh 192.168.0.10` (this command initiates a secure shell connection between the localhost computer and the internal local network computer at `192.168.0.10`.)

**stat** – Display file or filesystem status.

example: `user1@foo ]$ stat /home` (this command displays the status of the `/home` directory such as, block size, inodes, links, access and last modified date(s), among other user definable variables.)

**strace** – Trace system calls and signals.

example: `user1@foo ]$ strace -p 3284` (this command will start a trace on all system calls and signals associated with process # 3284.)

**su** – Substitute user command. most commonly used to run commands / daemons which require root privileges, view / alter certain directories, files and log files.

example: `user1@foo ]$ su -` (this command will prompt the user for the root password and, if supplied correctly, logs the user into roots profile. the `-` denotes that the user should inherit roots profile and path. this is quite helpful when issuing commands from many directories without typing the entire path to the command. system administrators should see the “`sudo`” command for extending certain privileges to users who do not need or cannot be trusted with the root password.)

**sudo** – Super user”do” allows a user to issue some commands which require elevated privileges as defined in `/etc/sudoers` but does not require knowledge of the root password.

example: `user1@foo ]$ sudo mount -t ext3 /dev/hdk3 /mnt/foo` (this command allows a normal user with `sudo` privileges to issue a command that requires elevated privileges to run. the user will be required to enter their own password to complete the command by default, this option can be changed to not require a password. see: `man sudoers`)

## | T |

**tail** – Output the last part of a file.

example: user1@foo ]\$ tail -n 5 vhosts.conf (this command outputs the last 5 lines of the file vhosts.conf to the terminal.)

**tar** – The tar command compresses or uncompresses tar or tar.gz archived files and directory structures.

example: user1@foo ]\$ tar -xvfz foo.tar.gz (this command uncompresses the file foo.tar.gz into the cwd.)

**top** – The top program provides a dynamic real-time view of a running system. It can display system summary information as well as a list of tasks currently being managed by the Linux kernel. The types of system summary information shown and the types, order and size of information displayed for tasks are all user configurable and that configuration can be made persistent across restarts. The program provides a limited interactive interface for process manipulation as well as a much more extensive interface for personal configuration — encompassing every aspect of its operation. And while top is referred to throughout this document, you are free to name the program anything you wish. That new name, possibly an alias, will then be reflected on top's display and used when reading and writing a configuration file.

example: user1@foo ]\$ top -u user1 (this command shows all the processes owned by user1 in realtime in the terminal.)

## | U |

**uname** – Print certain system information such as: currently running kernel, hostname, etcetera.

example: user1@foo ]\$ uname -a (this command prints information about the currently running kernel as well as other information.)

**useradd** – When invoked without the -D option, the useradd command creates a new user account using the values specified on the command line and the default values from the system. Depending on command line options, the useradd command will update system files and may also create the new user's home directory and copy initial files. The version provided with Red Hat Linux will create a group for each user added to the system by default.

example: root@foo ]# useradd -d -g 500 newuser (this command adds a new user to the system with a home directory of /home/newuser and a group id of 500. Alternatively, you can also use something like 'ftp' instead of '500' to specify what group(s) you want to add the new user to. see also: man adduser and userdel / deluser.)

**usermod** – Modify a user account.

example: user1@foo ]\$ usermod -a -G sshusers user1 (this command adds user1 to the group sshusers.)

## | V |

**vi** – vi is, quite simply, a programmers terminal-based text editor. vi is on every Unix / Linux system you will ever encounter and is very powerful. It can be used for high level programming or simply to edit a text file, save and close it.

example: user1@foo ]\$ vi /var/log/dmesg (this command displays the text file /var/log/dmesg in a terminal window and presents it for editing or reading.)

## | W |

**wall** – This command allows root or other users allowed to issue commands as root to send a message to everyones terminal.

example: user1@foo ]\$ wall how's everyone doing today? (this command prints the message "how's everyone doing today?" to the terminal.)

**wc** – The wc command prints newline, word, and byte counts for each file, and a total line if more than one file is specified. With no file, or when file is -, read standard input.

example: user1@foo ]\$ wc -l /var/log/dmesg (this command displays the total number of lines in the text file /var/log/dmesg.)

**who** – Shows who is currently logged on to the system.

example: user1@foo ]\$ who -last (this command displays verbose output of the users and processes logged into the system.)

**whois** – (see **jwhois**.)

## | X |

**xrandr** – Xrandr is used to set the screen size, orientation and/or reflection. The -s option is a small integer index used to specify which size the screen should be set to. To find out what sizes are available, use the -q option, which reports the sizes available, the current rotation, and the possible rotations and reflections. The default size is the first size specified in the list. The -o option is used to specify the orientation of the screen, and can be one of “normal inverted left right 0 1 2 3”. The -x option instructs the server to reflect the screen on the X axis. The -y option instructs the server to reflect the screen on the Y axis. Reflection is applied after rotation. The -help option prints out a usage summary. The -verbose option tells you what xrandr is doing, selects for events, and tells you when events are received to enable debugging.

example: user1@foo ]\$ xrandr -verbose (this command shows the current reflection, rotation and other information.)

**xwininfo** – Xwininfo is a utility for displaying information about windows. Various information is displayed depending on which options are selected. If no options are chosen, -stats is assumed. The user has the option of selecting the target window with the mouse (by clicking any mouse button in the desired window) or by specifying its window id on the command line with the -id option. Or instead of specifying the window by its id number, the -name option may be used to specify which window is desired by name. There is also a special -root option to quickly obtain information on the screen's root window.

example: user1@foo ]\$ xwininfo (this command displays the geometry for the specified window.)

## | Y |

**yum** – Yum is a powerful, interactive, and automated package update / query / install program which can be used for maintaining systems using rpm (Redhat Package Manager).  
example: root@foo ]# yum -exclude=kernel\* update (this common command initiates a full system update, while telling yum to exclude / ignore any kernel packages.)

## | Z |

**zip** – Zip is a compression and file packaging utility for Unix, VMS, MSDOS, OS/2, Windows NT, Minix, Atari and Macintosh, Amiga and Acorn RISC OS. It is analogous to a combination of the UNIX commands tar(1) and compress(1) and is compatible with PKZIP (Phil Katz's ZIP for MSDOS systems). A companion program (unzip), unpacks zip archives. The zip and unzip programs can work with archives produced by PKZIP, and PKZIP and PKUNZIP can work with archives produced by zip. Zip version 2.3 is compatible with PKZIP 2.04. Note that PKUNZIP 1.10 cannot extract files produced by PKZIP 2.04 or zip 2.3. You must use PKUNZIP 2.04g or unzip 5.0p1 (or later versions) to extract them.

example: user1@foo ]\$ zip -r foo1 foo2 (this command creates the archive foo2.zip, containing all the files and directories in the directory foo1 that is contained within the current directory.)