

Guide - Basic Commands

Linux has so many commands that it's easy for the uninitiated to feel overwhelmed. Don't panic. Linux's commands are all quite simple and easy to learn. As for those commands that appear bewilderingly complex, it's just an illusion. Long, seemingly complicated commands are actually just made up of multiple, simpler commands. They achieve their power by being able to work together. This is the way of the UNIX world in general.

- Let's look at an example: **cat /var/log/messages | grep failed**

This command will sort through your log file and list all of the failed login attempts since the system was last started. The entire line is actually composed of three small commands:

- **cat** prints out the contents of a text file. Here, we are telling it to display the contents of the file `/var/log/messages` (this is your main log file.)
- `|` This symbol is called the pipe (it is located to the left of your backspace key.) It is used to take the output of one command (in this case `cat`) and pass it to a second (`grep`, in our example.)
- **grep** will sort through all of the lines it is given (in this case the lines from the messages file), grab any lines with a word matching the one you specify (I specified the word 'failed'), and display them.

There is no end to the possibilities of combining simple commands to achieve complicated tasks. In this way, the command line becomes much more powerful and time-saving than the point-and-click wizards of Windows. (I once had to change the ownership and access rights of ~150 users' home directories and files on an NT server. It took over 3 hours of clicking. With proper command line tools, it could have been done in minutes.)

I list only a few commands and options (the ones I find myself most commonly using). You will quickly outgrow this list. When you do, use the `--help` switch to get a full listing of the commands options (e.g. `tar --help`.) You can find more comprehensive lists of commands from other newbie guides. Also, do yourself a favor and buy the book "Linux in a Nutshell" published by O'Reilly. Just browsing through it is an indispensable way to learn new Linux commands.

Some Other Notes and Tips:

1. Save yourself a ton of typing by using command completion (the `<Tab>` key.) See explanation below.
2. Your present working directory is not in your path by default, nor should it be (to keep trojans and viruses from executing themselves.) To run an executable or script that is not your path, you must specify that the program is in your current directory by prefacing it with `./`. For example, to start Quake3 from the `quake3` directory, you would type **`./quake3`**
3. You can cycle through previously entered commands with the up/down arrow keys.
4. Remember that Linux is case sensitive.

A Few Notes on the Syntax of `--help`

You can get a description of a command's use by typing `<command> --help` at the command prompt. If the description is so long that it runs off the top of the screen, try piping it to `more` (or less depending on your preference.) For example, here is the first section of the help file for the command `tar`: (I cut off a couple of lines so this page will print properly.)

```
[hoffman@buffy hoffman]$ tar --help | more
GNU 'tar' saves many files together into a single tape or disk
archive, and can restore individual files from the archive.
```

```
Usage: tar [OPTION]... [FILE]...
```

```
If a long option shows an argument as mandatory, then it is
mandatory for the equivalent short option also. Similarly for
optional arguments.
```

```

Main operation mode:
-t, --list          list the contents of an archive
-x, --extract, --get extract files from an archive
-c, --create        create a new archive
-d, --diff, --compare find differences between archive and
-r, --append        append files to the end of an archive
-u, --update        only append files newer than copy in
-A, --catenate      append tar files to an archive
                   --concatenate    same as -A
                   --delete         delete from the archive (not on mag

Operation modifiers:
-W, --verify        attempt to verify the archive after
--remove-files      remove files after adding them to
--More--

```

Note the usage description in the example. Optional elements are enclosed in brackets: []. The brackets themselves should never be typed. Sometimes, you may run into a syntax that looks like this: **<command> [option A | option B]**. When options are divided by the pipe symbol, you may choose one or the other but not both. In the example using tar above, we are told to type tar, followed by the options we want, followed by the name of the file(s) to act upon. Then we are given a listing of the options we can use with tar. If you want to use multiple options, you can usually stick them all together. For example, I could use tar's options to unzip, extract, and be verbose: **tar zxvf filename.tar.gz**. Note here that I did not use a dash to preface my options. Some commands require a dash, others do not. You will get used to it.

Some General Commands

login	Log into the system as specified user. I'm not sure what distributions require you to use this. Usually you can just enter your user name at the login prompt. Usage: login [username]
logout	Log out of the system.
startx	Start the X window system. Use this to bring up KDE, Gnome, etc. You can press Ctrl-Alt-Backspace to shutdown X if something goes wrong.
<Alt>- [F1-F6]	Switches to the specified virtual terminal. At system start, virtual terminal 1 is displayed. You can log into this terminal as one user and switch to a second terminal (by pressing Alt-F2) and login as another user. You can be logged into multiple terminals at the same time.
grep	Use grep to grab lines with text matching your criteria. Example: ps ax grep inetd The above example will list any process with 'inetd' in its name
<Tab>	Command completion. Linux will try and guess the name of the file or directory that you are typing and fill in the rest of the file or directory name for you. Type in the first few letters of the file or directory name and press the <Tab> key. If there are similar existing filenames, Linux will fill in as much as it can. This will save you a lot of typing.
script	Script will record everything displayed in a terminal and save it as a file. This is very useful for recording a terminal session. Most of the examples found on this website were recorded using script. Usage: script [filename] Press Ctrl-d to stop script.
 	The pipe. Lets you pass the output of your first argument to a second command. Usage: <command 1> <command 2> Example: ps -aux less This example will list all running processes on your computer and pipe then to less so that you can scroll up and down the list to view them. Press q to quit less.

>	<p>Takes the data you send to it and saves it to a text file.</p> <p>Usage: <input> > [filename]</p> <p>Example: echo Hello World! > testfile This example will create a file called 'testfile' containing a single line reading "Hello World!" Be careful with this. If 'testfile' had already existed, it would have been overwritten.</p>
>>	<p>Appends text to the specified file.</p> <p>Usage: <input> >> [filename]</p> <p>Example: echo Another line of text >> testfile This example will add a line reading "Another line of text" to the end of the file 'testfile'.</p>

Basic Filesystem Commands

pwd	Shows your present working directory. Has nothing to do with your password (try passwd instead.)
ls	<p>Lists the contents of your present directory (The same as <i>dir</i> in DOS.) You will know when you have become comfortable with Linux because you will find yourself typing <i>ls</i> instead of <i>dir</i> on Windows machines.</p> <p>ls -l will give you a long listing of a directory showing the attributes of its files and subdirectories.</p>
cd	<p>Change directory.</p> <p>Usage: cd [directory] If you don't specify a directory, cd will take you to your home directory.</p>
cp	<p>Copy.</p> <p>Usage: cp <source> <destination></p>
mv	<p>Move. Can also be used to rename a file.</p> <p>Usage: mv <source> <destination></p>
rm	<p>Remove. Deletes a file or directory.</p> <p>Usage: rm <filename></p> <p>rm -f <filename> Will force the deletion without asking for confirmation. rm -R <directory> Will delete the directory and all of its subdirectories (the 'R' stands for 'recursive'.)</p>
mount	<p>Mounts a drive. Root privileges are normally required to mount devices.</p> <p>Usage: mount <device> <mountpoint></p> <p>Example: mount -t vfat /dev/fd0 /mnt/floppy</p> <p>The above example will mount a DOS diskette as /mnt/floppy.</p>
umount	<p>Unmounts a previously mounted volume.</p> <p>Usage: umount <mountpoint></p>
cat	<p>displays the contents of a file.</p> <p>Usage: cat <filename></p>
vi	<p>Powerful text editor built on alien technology. See the vi guide for more information.</p> <p>Usage: vi <filename></p>

less	Lists the contents of a file or its input so that you can scroll up or down through it. Hit q to quit. Usage: less <filename>
tar	A very powerful utility used for combining multiple files into one and vice versa. It can also be used to compress and decompress files. Usage: tar [options] [filenames] Note that you should always list the option 'f' last or you will get nasty output to your monitor. Don't ask me how I know this. Example: tar zxvf foo.tar.gz The above example will decompress, and extract the archive called foo.tar.gz.
chown	Change the owner of a file. Usage: chown <username> [filename directory] You can also use chown to change the group name by separating the owner and group names with a period like so: chown owner.group file1
chmod	Change the read, write, and executable settings for a file or directory. Use this to make that batch file you just made to run Quake3 executable. Example: chmod 755 runq3 The above example would give read, write, and execute permissions to the file 'runq3' to its owner and make it read-writable to everyone else.
df	Disk Free. Shows Hard Disk usage statistics for your system.
tail	tail will display the last ten lines of a text file. Actually, it will display as many lines as you specify but ten is the default. Usage: tail <filename> Example: tail -50 /var/log/messages The above example will display the last 50 lines of your log file.
mkdir	Make a new directory. Usage: mkdir <directory name>
rmdir	Remove a directory Usage: rmdir <directory name>
du	Display statistics about your Hard Disk usage.
find	Find is an exceptionally powerful utility because it allows you to execute commands on the items you find. See find --help for a full listing of options. Usage: find [options] [path] [matching criteria] Example: find / -iname quake* The above example will search all subdirectories of root (/) for any files having names that begin with 'quake' regardless of case (-iname).
chgrp	Change the Group ownership on a file or directory. Usage: chgrp <group owner> [filename directory]

Basic Administrative Commands

su	Super User! Use this to switch from and ordinary user to root. Type exit to return to your mild-mannered self.
adduser	Adds a new user to the system. Usage: adduser <username>

passwd	Can be used to change your password or to assign a password to a new user. Usage: passwd [username] If you don't specify a user, the present user's password will be changed.
shutdown	Shutdown or reboot the computer. Usage: shutdown [option] [time] Example: shutdown -h now The above example will immediately shutdown and halt the computer. Example: shutdown -r 0 The above example will reboot the computer in zero minutes.
w	Displays uptime as well as a list of current users and what they are doing.
free	Displays the memory statistics for your system. Works like DOS's 'mem' command.

Basic Process Control

ps	Lists all running processes. Example: ps ax The above example will list all running processes.
top	Displays a chart of running processes and the resources they are using. The list is ordered by the most cpu intensive programs and updated every few seconds.
kill	Kill is used to shutdown or restart processes. Use ps to find the process id number of the app you want to kill. Usage: kill [option] <process ID number> Example: kill -1 <process ID> The above example would cause a daemon with the specified process ID number to reread its configuration file, essentially re-starting it. Be very careful not to omit the negative symbol...killing process #1 will kill your system. Example: kill -9 <process ID> The above example will kill an app with extreme prejudice. This is very useful if you run Netscape under Gnome.
xkill	Typing this in a xterm window will prompt you to click on a window that you want to shut down. Use this to close misbehaving applications in X.
<Ctrl>-c	This will stop whatever process you have running in your terminal.

Basic Network Commands

ifconfig	Displays information about your network devices. This command produces results similar to winipcfg in Win 95/98 and ipconfig /all in Win NT. You can also use ifconfig to configure your network adapters.
ifdown	Takes down the specified network adapter. This is very useful if you want to enact a setting change for a particular adapter. Just take it down then back up. No rebooting necessary. Usage: ifdown [interface] Example: ifdown eth0 The above example would take down your first Ethernet card.

ifup	<p>Brings up the specified network adapter.</p> <p>Usage: ifup [interface]</p> <p>Example: ifup eth0 The above example would bring up your first Ethernet card.</p>
ping	<p>Ping sends ip packets to a remote computer and looks for a response. This is very useful for checking on whether a host is alive on the network or not.</p> <p>Usage: ping [ipaddress hostname]</p> <p>Example: ping www.frankenlinux.com Press Ctrl-c to stop pinging.</p>
telnet	<p>Telnet is used to connect to and work on a machine remotely. See the Telnet Guide for a full description.</p>
ftp	<p>File Transfer Protocol. Use this to transfer files across a network. See the FTP Guide for a full description.</p>