Essential Linux Commands

by Terry Sergeant

Abstract

This is intended to provide a listing of (and in some cases descriptions of) essential commands for navigating and working in a Linux system.

1 Pre-requisites

- In order to survive the Linux command line you *must* become proficient at using a text-based editor (e.g., vi, emacs, pico).
- If you only remember one command, remember this:

  `man [command]`

  The `man` command looks up the manual pages for the command you specify. In this way you can get information about any of the other commands.

2 Working with Files and Directories

It is important for you to realize that in Linux, *file names are case-sensitive* (along with everything else). Most characters are valid filename characters although some characters (such a spaces) will need to be escaped on the command-line.

Linux organizes files using a hierarchical directory structure and separates directories in a path name using a *forward* slash ("/"). Files with names that begin with a period are designated "hidden" and are immune from some file operations.

When working with path names there are some special characters of note:

. *(single period) means current folder*

.. *(double period) means parent folder*

~ *(tilde) means home folder*

– *(dash) means where I last was*

When working with file names there are two wildcard characters that can be used:

***** *(asterisk) matches 0 or more characters*

? *(question mark) matches exactly one character*

2.1 File Manipulation

pwd *displays present working directory.* Depending on how your prompt is displayed you may not have a reminder as to what directory your are currently working in. This command will let you know where you are.

ls *lists files in a directory.* There are many options that can be supplied to modify the behavior of `ls`. Here are some examples of using `ls`.

`ls`	lists all files in current directory
`ls -F`	modifies file listing to differentiate between types of files (directories, including permissions
`ls ~/progs/`	lists all files in the `progs` subdirectory under your home directory
`ls -a`	shows all files, including hidden ones (whose names begin with a period)
`ls *cpp`	shows all files in the current directory whose names end with `cpp`
`ls -al`	same as doing both `-a` and `-l`

cd *changes to a new working directory.* Two notes of interest: `cd ..` moves one level up the directory tree. Also, `cd` with no directory name changes to the user's home directory (the directory you are in when you first log in).

mv *moves or renames a file.* This command can be used rename a file or to move it from one directory to another. It should be noted that mv can be applied to directories as well as files. Examples:

`mv old.txt new.dat`	renames file from `old.txt` to `new.dat`.
`mv myprogs/cool.cpp .`	moves file named `cool.cpp` in the `myprog` directory into the current directory
`mv myprogs oldprogs`	if we assume that `myprogs` is a directory (as in the previous example) then this would rename the directory to be `oldprogs`

cp *copies a file.* This works pretty much the same way as `mv` with two exceptions. First, the original file is left intact. Secondly, `cp` doesn't copy entire directories (unless you use the `-r` option).

rm *removes a file.*

chmod *changes mode (access permissions) for a file.* Every file and directory has read, write, and executable permissions for three classifications of users: the user (u), the group (g), and others (o).[1] For example, a file can be set to be readable and writeable, but not executable for the user/owner and not readable, writeable, or executable for group members or for anyone else. Likewise, you may want to allow anyone to view a particular file, but not write to it, etc. To view the current settings you can use the `ls -l` command which will produce output similar to this:

```
drwxrwxr-x  3 sergeant sergeant  1024 Aug 14 10:27 office52
-rw------- 1 sergeant sergeant 26907 Jul 30 1999 personal.xls
drwxrwxr-x  2 sergeant sergeant  1024 Jul 24 07:50 rpm
-rw-rw-r-- 1 sergeant webcache 28492 Aug 18 18:29 stats.ps
-rw-rw-rw- 1 sergeant sergeant  5373 Feb 5 2000 questions.tex
```

Each line contains information for a single file. The first ten characters provide information about permissions. The first character has a `d` if the file is a directory and a `-` otherwise (usually). The remaining nine characters specify the read (r), write (w), and executable (x) permissions for user, group, and others. The file named `personal.xls` is readable and writeable by the user (`sergeant`) and not available to anyone else. The file `stats.ps` is available for reading and

[1]That's nine permission settings if you're counting.

writing to the user (`sergeant`) and the group (`webcache`), but only readable to everyone else. You may notice that both directories (`office52` and `rpm`) have executable permissions turned on ... without those permissions a directory is not listable/viewable.

There are two main ways to invoke the `chmod` command. Only the mnemonic method is explained here. The command is followed by a request for new permissions and lastly the filename(s) to which the new permissions should be applied.

`chmod guo+rw fun.txt`	makes the file `fun.txt` readable and writeable by anyone
`chmod go-rwx *cpp`	remove all types of access from group or others for all files whose names end with `cpp`

mkdir *makes a new directory.* Simply follow the command with the desired directory name.

rmdir *removes a directory.* The main thing to be aware of here is that the directory must be empty before it can be removed with `rmdir`.

ln -s *creates a symbolic link* (i.e., shortcut) from one location in the file system to another.

2.2 Other File-Related Commands

tail *show the last several lines of a file.*

find *finds files.* This command has numerous options that allow you to find a file based on all manner of criteria. Example: `find -name henry.txt` will search the current directory and all its subdirectories for a file named `henry.txt`. Example: `find -atime -1` finds all files in current directory and its subdirectories which have been accessed in the last 1 day. Example: `find -name "tmp*" -exec rm {} \;` delete all files that start with `tmp` (in the current directory and any of its subdirectories).

which *identifies which file is executed by a command.*

du / df *displays disk space used / disk space free.*

less *displays file with ability to scroll.* This command is often used as a destination for redirected output.

passwd *changes your password.* NOTE: Use the `yppasswd` command if using a lab computer.

tar *is used to package a group of files into a single file.* Many programs or files are distributed via “tarred and zipped” files. `tar` puts them all into a single file and compression software is used to compress the files. An example of extracting files from an archive:

```
tar xvzf files.tar.gz
```

NOTE: x=extract, v=verify, z=use gzip, f=file name follows

To create an archive you could do something like this:

```
tar cvzf myfiles.tgz mydir
```

NOTE: c=create, f=name of file to create, mydir=name of directory containing files

script *captures output to a file.* This is especially useful for capturing the output of an interactive program (as is often requested by an instructor). The command works as follows:

```
script myprog.out
ls
c++ prog1.cpp
./a.out
exit
```

In this example all of the output generated by the `ls` command and by the program compilation and by the program that was executed are all sent to the file named `myprog.out` (as well as to the screen). The command `exit` tells the script command to stop logging the output to the specified file.

3 Admin Commands

These commands are not all exclusively the domain of the root user. When they are used, though, they are often invoked by an admin.

chown *change the owner of a file/directory.*

chgrp *change the group of a file/directory.*

yum *is used to install software and keep the system up-to-date.* Depending on the Linux distro you may use a different program for this purpose. This command is only available to the root user. Some common options are:

`yum update`	download and install all available updates
`yum install prog`	install program named `prog`
`yum remove prog`	unistall program named `prog`

su *change to a different user’s identity.* With no arguments you become the root user. To become another user just include the name of the user you want to become. To become the root user requires the password, of course. If you are the root user you can become another user without a password.

zip/unzip/gzip/gunzip *used for compressing or uncompressing files* in `.zip` or `.gz` format.

ps *lists processes in the system.* There are *many* options to this command. To see all processes in the OS along with helpful stats on them use:

```
ps guax
```

NOTE: It is not unusual for the above command to fill up more than one screen, so you may want to pipe it to `less` (for scrolling) or to `grep` (for searching)

top *gives a constantly updating display of the processes* that are consuming the most resources last

kill *terminates a process.* This is useful if a program won’t respond. You will need to know the process id of the process you want to kill (i.e., `ps`). If a program doesn’t respond to the kill command then you can add the `-9` option to kill it less politely.

w *lists users currently logged into the system* along with other information about their session

whoami *displays the user you are currently acting as.* This becomes useful when you are an admin who is changing roles.

last *show the last several logins*

history *display commands that have been previously issued*

4 Networking Commands

4.1 Troubleshooting Commands

ping *checks to see if a given server/host is on.*

ifconfig *displays current networking status/settings.*

iptables *work with the firewall.* There are numerous options with this command.

traceroute *learn route from your computer to another host.*

wget *retrieve the specified URL; (this command is actually a program that may need to be installed to be available)*

4.2 Communications

ssh *invokes ssh client for connecting to a remote host.*

sftp *invokes sftp client for connecting to a remote host.*

telnet *opens a terminal to specified host.* This is especially useful for testing a firewall or debugging service on a particular port.

mutt *a text-based email client.*