

Important Linux Commands

(Jacques Lagnel 2012)

Help on any Unix command

<code>man {command}</code>	Type man ls to read the manual for the ls command.
<code>whatis {command}</code>	Give short description of command.
<code>apropos {keyword}</code> or <code>man -k {keyword}</code>	Search for all Unix commands that match keyword, eg apropos file .

List a directory/files

<code>ls {directory}</code>	It's ok to combine attributes, eg ls -laF gets a long listing of all files with types.
<code>ls {directory_1} {directory_2}</code>	List both <code>{path_1}</code> and <code>{path_2}</code> .
<code>ls -l {directory}</code>	Long listing, with date, size and permissions.

Change to directory

<code>cd {directory}</code>	There must be a space between.
<code>cd ~</code>	Go back to home directory, useful if you're lost.
<code>cd ..</code>	Go back one directory.

Make a new directory

<code>mkdir {dirname}</code>	
------------------------------	--

Remove a directory

<code>rmdir {dirname}</code>	Only works if <code>{dirname}</code> is empty.
<code>rm -r {dirname}</code>	Remove all files and subdirs. Careful!

Print working directory

<code>pwd</code>	Show where you are as full path. Useful if you're
------------------	---

	lost or exploring.
--	--------------------

Copy a file or directory

<code>cp {file1} {file2}</code>	
<code>cp -r {dir1} {dir2}</code>	Recursive, copy directory and all subdirs.
<code>cat {newfile} >> {oldfile}</code>	Append newfile to end of oldfile.
<code>scp {from machine} {to machine}</code>	Secure copy file/s over network {from} to {to}

Move (or rename) a file

<code>mv {oldfile} {newfile}</code>	Moving a file and renaming it are the same thing.
<code>mv {oldname} {newname}</code>	Rename file
<code>rename '.texte' '.txt' *.texte</code>	Rename files with extension .texte to .txt
<code>perl-rename</code>	A more powerful rename tool

Delete a file

<code>rm {filespec}</code>	? and * wildcards. "?" is any 1 character; "*" is any string of character(s).
----------------------------	---

View a text file

<code>more {filename}</code>	View file one screen at a time.
<code>less {filename}</code>	Like more , with extra features.
<code>cat {filename}</code>	View file, but it scrolls.
<code>head {filename}</code>	Output the first 10 lines of filename
<code>tail {filename}</code>	Output the last 10 lines of filename
<code>tail -f {filename}</code>	Output the contents of file as it grows, starting with the last 10 lines
<code>cat {filename}</code>	Display the full content of filename

Edit/create a text file.

<code>nano {filename}</code>	Nano is a small text
------------------------------	----------------------

	editor.
vi or vim {filename}	vi/vim is a powerful one.
sed 's/spam/bar/' {file}	Command line editor replace spam by bar in file
csplit {filename} {pattern}	Split filename with pattern

Create a empty text file.

touch {filename}	Create create empty file.
------------------	---------------------------

Compare two files

diff {file1} {file2}	Show the differences.
----------------------	-----------------------

Other text commands

grep '{pattern}' {file}	Find regular expression in file.
awk	text pattern scanning and scripting processing language
sort {file1} > {file2}	Sort file1 and save as file2.
sort -o {file} {file}	Replace file with sorted version.
wc {file}	Count words in file.
ln -s {file} {slink}	create symbolic link slink to file
uniq {filename}	Find unique/duplicate line in filename
wget	download file(s) from http or ftp
Dos2unix {filename}	Convert line ending from dos to unix format

Find files on system

find {filespec}	Works with wildcards. Very powerful.
-----------------	---

File compression

tar -xvzf {filename.tar.gz}	Uncompress the file filename.tar.gz
tar -cvzf {filename.tar.gz} {file(s) or	Compress {file(s) or

directory}	directory} in filename.tar.gz
------------	----------------------------------

Wildcards and Shortcuts

*	Match any string of characters, eg page* gets page1, page10, and page.txt.
?	Match any single character, eg page? gets page1 and page2, but not page10.
[...]	Match any characters in a range, eg page[1-3] gets page1, page2, and page3.
~	Short for your home directory, eg cd ~ will take you home, and rm -r ~ will destroy it.
.	The current directory.
..	One directory up the tree, eg ls ..

Redirection and Pipes

(You **pipe** a command to another command, and **redirect** it to a file.)

{command} > {file}	Redirect output to a file, eg ls > list.txt writes directory to file.
{command} >> {file}	Append output to an existing file, eg cat update >> archive adds update to end of archive.
{command} < {file}	Get input from a file, eg sort < file.txt
{command} < {file1} > {file2}	Get input from file1, and write to file2, eg sort < old.txt > new.txt sorts old.txt and saves as new.txt.
{command} {command}	Pipe one command to another, eg ls more gets directory and sends it to more to show it one page at a time.

Permissions, important and tricky!

Unix permissions concern who can **read** a file or directory, **write** to it, and **execute** it. Permissions are granted or withheld with a magic 3-digit number. The three digits correspond to the **owner** (you); the **group** (?); and the **world** (everyone else).

Think of each digit as a sum:

execute permission= 1 write permission= 2 write and execute (1+2)= 3

read permission= 4 read and execute (4+1)= 5 read and write (4+2)= 6

read, write and execute (4+2+1)= 7

Add the number value of the permissions you want to grant each group to make a three digit number, one digit each for the owner, the group, and the world. Here are some useful combinations. Try to figure them out!

<code>chmod 600 {filespec}</code>	You can read and write; the world can't. Good for files.
<code>chmod 700 {filespec}</code>	You can read, write, and execute; the world can't. Good for scripts.
<code>chmod 644 {filespec}</code>	You can read and write; the world can only read. Good for web pages.
<code>chmod 755 {filespec}</code>	You can read, write, and execute; the world can read and execute. Good for programs you want to share, and your <code>public_html</code> directory.

Permissions, another way

You can also change file permissions with letters:

u = user (yourself) **g** = group **o** = other **a**=all **r** = read **w** = write **x** = execute

<code>chmod u+rw {filespec}</code>	Give yourself read and write permission
<code>chmod u+x {filespec}</code>	Give yourself execute permission.
<code>chmod ugo+rw {filespec}</code>	Give read and write for all .

System tools

<code>date</code>	Show date and time.
<code>df</code>	Check system disk

	capacity.
du	Check your disk usage and show bytes in each directory.
env	Show all environmental variables.
w	Who's online and what are they doing?
Kill -9 PID	Kill process (force with -9) with a given PID number
Killall blastall	Kill all processes named blastall
top	Display all running processes
ps	Display your currently active processes