Homepage of Dr Phil Hasnip

Research

CMT Seminars

Teaching

Publications

# Top Unix Commands

In Unix most operations are carried out by typing commands at a command prompt, usually called a *terminal* or *shell*. Many of these commands have optional features, and these are usually activated by *switches*. A switch is specified by a hyphen "-" (or occasionally two hyphens "--") followed by a letter or word. Every command has its own set of switches, but we've included the most common in this list.

1. Listing files

| | |
|---|---|
| **ls** | list files in current directory |
| **ls -l** | list files in a long format |
| **ls -a** | list all files (including hidden files) in current directory |
| **ls -F** | adds indicators to the list output to identify directories and different types of files. |

These switches can be combined, for example **ls -al** would list all files in the current directory, including hidden files, in a long format. You can also give a directory or file name, e.g. **ls /home/pjh503** lists files in the directory "/home/pjh503".

2. Special characters

In Unix some characters have special meanings that you can use. For example the asterisk "*" means "any string", so the command **ls a*isk** will list all the files in the current directory that start with "a" and end "isk". These special characters are not allowed to be used in the actual names of files or directories.

| | |
|---|---|
| **.** | the current working directory |
| **..** | the parent directory to working directory |
| **~** | your home directory |
| **/** | the root (top-level) directory. This is also the separator for directories |
| ***** | a wildcard meaning any string of characters |
| **?** | a wildcard meaning any single character |

3. Files

| | |
|---|---|
| **cp** file1 file2 | makes a copy of file1 and calls it file2 |
| **mv** file1 file2 | moves (renames) file1 to file2 |
| **rm** file1 | removes (deletes) file1 |
| **rm -i** file1 | asks for confirmation that you want to delete file1 |

4. Directories

| | |
|---|---|
| **pwd** | print the working directory |

| | |
|---|---|
| **cd** dirname | change directory to the one called "dirname" |
| **cd ..** | change to the parent directory of the current directory |
| **cd ~** | change to your home directory |
| **mkdir** dirname | makes a new directory with name "dirname" |
| **rmdir** dirname | removes the directory with name "dirname". The directory must be empty |
| **rmdir -r** dirname | recursively removes directories and subdirectories |

5. Text files

| | |
|---|---|
| **cat** file1 | writes the whole of file "file1" to the terminal, also useful for concatenating files |
| **more** file1 | displays the file "file1" a page at a time |
| **less** file1 | a more versatile version of "more", but less common |
| **head -30** file1 | show the first 30 lines |
| **tail -25** file1 | show the last 25 lines |
| **tail -f** file1 | show the last few lines and keep updating as the file grows |
| **wc** file1 | counts lines, words and characters in a file |

6. Help

| | |
|---|---|
| **man** utilityname | manual pages for the command "utilityname" |

7. Useful keypresses

| | |
|---|---|
| **ctrl C** | interrupts whatever is currently running.<br>(It can get you out of trouble at embarrassing moments) |
| **ctrl Z** | puts a foreground process into the background. |
| **ctrl S** | suspends current terminal |
| **ctrl Q** | resumes current terminal |

8. Permissions and ownership

In Unix only the administrator (called "root") can modify everything. Each file and directory is owned by one of the users, and (like users) also belongs to a "group". As a user you can set whether your files can be read, written or executed by just you, only members of your group, or everyone. You can see the permissions of a file using **ls -l**. E.g.,

```
-rw-r-----  1 pjh503 phys  3360 Sep 26 15:40 fortran.html
drwxr-xr-x  2 pjh503 phys  4096 Sep 27 23:56 Linux
```

The important part is the string of 10 characters at the start. The first character says what type of thing this entry is, usually either "-" for a normal file or "d" for a directory. You can see in the example that "fortran.html" is a file, and "Linux" is a directory.

The next nine characters are actually grouped as three sets of three. The first set of three refers to the permissions the owner, in this case "pjh503" (me). The "r" means I can read the file, the "w" means I can write to the file, and the "-" means that I *cannot* execute it. If I could execute it, it would have an "x" in the third permission. For directories "execute permission" has a different meaning -- it means I can list its contents.

The next set of three refers to users who are not the owner (pjh503 in this case) but are in the same group as the owner (phys in the example above). In the example above users in the group phys can read the file "fortran.html" but not write to it.

The final three characters refer to anyone who is neither the owner, nor in the same group as the file.

| | |
|---|---|
| **chmod** | changes file and directory permissions |
| **chmod u+r** file1 | change permissions of "file1" so user ("u") gains ("+") read permission ("r"). |
| **chgrp** group1 file1 | change "file1" to belong to group "group1" |

In general the permissions use the following format:

| | |
|---|---|
| **u** | user |
| **g** | group |
| **o** | others |
| **a** | all (equivalent to **ugo**) |
| **+** | gain |
| **-** | lose |
| **r** | read permission |
| **w** | write permission |
| **x** | execute permission |

These can be combined, so for example to grant read and write permission to the user who owns a file and anyone in the same group as the file you could use **chmod ug+rw**.

9. Using printers

| | |
|---|---|
| **lp -P** printername file1 | prints file1 to printer "printername" |
| **lpq -P** printername | enquiry, print queue |
| **lprm -P** printername jobnumber | removes print job "jobnumber" from print queue |

10. Calculator

| | |
|---|---|
| **bc -l** | command-line calculator |

11. Processes

| | |
|---|---|
| **ps** | list my processes (programs) that are running, along with their process ID (PID) |
| **kill** pid | kill (stop) my process with the given PID |
| **top** | show the top few processes sorted according to CPU usage<br>Once top is running, type M to sort by memory usage instead, and q to quit |

12. Searching for things

| | |
|---|---|
| **sort** files | sort the specified files |
| **grep** pattern files | search files for particular patterns |
| **find** dirname **-name** file1 | search directory "dirname" and subdirectories for files called "file1" |

13. Previous Commands

You can use the up and down arrows on the keyboard to scroll through previous commands. Alternatively:

| | |
|---|---|
| **history 15** | lists your last 15 (variable) commands and numbers them |
| **!!** | repeats your last unix command |
| **!**23 | repeats the command numbered 23 |
| **!**f90 | repeats the last command beginning with e.g. "f90" |
| **!!**addtext | appends "addtext" to previous command line |
| **^**string1**^**string2 | substitutes "string2" for "string1" in previous command |

14. Redirection

Many commands take their input from the keyboard and write their output to the terminal window, but this isn't always what you want. You can redirect the input and output of commands.

| | |
|---|---|
| command **>** file1 | redirects the output of "command" to "file1" instead of to standard output (screen) |
| command **>>** file1 | appends the output of "command" to "file1" instead of to standard output (screen) |
| command **<** file1 | takes input for "command" from file1 |
| command1 **\|** command2 | pipe standard output of command1 to standard input of command2 |

15. Remote access

| | |
|---|---|
| **ssh** user1@machine1 | login securely as user "user1" into machine "machine1". |

| | |
|---|---|
| **scp** file1 user1@machine1: | copy file "file1" to the home directory of user "user1" on machine "machine1". Note the colon! |
| **sftp** machine1 | interactive secure ftp (file transfer program) |

16. Compressing files

| | |
|---|---|
| **gzip** file1 | compress file "file1". The compressed file will be called "file1.gz" |
| **gunzip** file1.gz | uncompress file "file1.gz". The uncompressed file will be called "file1" |