# BASIC UNIX COMMANDS

## COMMANDS

Commands are what you type at the prompt. Commands have arguments on which they operate. For example, in `rm temp`, the command is `rm` and the argument is `temp`; this command removes the file called `temp`. Here I put arguments in UPPER CASE. Thus, words such as FILE are taken to stand for some other word, such as `temp`. In the following list, I use [] for optional arguments that are not typicaly used.

Commands have options that are controlled with switches, which are usually letters following a single dash. Usually you can write several letters after one dash. For example `ls -l` lists files in a long format, with more information. `ls -a` lists all the files, including those that begin with `.`, which are usually files used by various programs. `ls -al` lists all the files in the long format. The following list omits most options. Many programs will give you a list of their options if you type the name of the program followed by `-h` or `-?`.

Other options are controlled in special files usually beginning with `.`, such as `.emacs, .mailrc`, and `.newsrc`. You can edit these. The file `.cshrc` contains general options, plus aliases that you make up for commands you use often.

`|`, the pipe symbol, takes the output of one command and gives it to the second command as input, for example, `ls -l | more` allows you to view a long file list through "more".

`>` directs the output to a file, e.g., `ls -l > listing.tmp` puts the listing into a file named listing.tmp. `>` overwrites an old file of the same name. `>>` appends to a file.

`*` stands for any string of letters. For example, `ls t*` lists all the files beginning with t.

↑ recalls previous commands, and you can edit these commands using emacs editing.

The tab key completes commands if you type the first few letters.

ctrl-z suspends most programs. `fg` resumes. ctrl-c stops most programs.

`alias` abbreviates a series of commands, separated by semicolons. Useful in `.cshrc`.

## GETTING HELP

`apropos KEYWORD`: Looks for commands related to KEYWORD.

`man COMMAND`: Shows manual pages on COMMAND. This is the authoritative source on the items described here. The commands can do much more than is listed here.

`whatis COMMAND`: Tells what COMMAND does.

## FILES AND DIRECTORIES

All information is stored in files. File names and commands are case sensitive. Case matters. Files are contained in directories. You start out in your own home directory, and your prompt usually tells its name. At any given time, one of these directories is your working directory, the one you are in.

You can refer to files in your working directory by just their names. You can refer to a file that is in a subdirectory by giving a subdirectory name, a slash, and the file name, e.g., Mail/baron. You can refer to any file on the computer by giving its full name, starting with a slash, such as /home7/b/baron/mbox.

If the file is a program, typing its name will run it. (That is what commands do.) If the program is something you have just written and is in the director you are in, put `./` before the name. If the file is not a program, typing its name will give you an error message. If you want to see its contents, for example, you must use a command such as "more" before the file name.

`ls [DIRECTORY]`: Lists files. (Also try: ls -f, ls -s, ls -a.)

`rm FILE`: Removes FILE.

`more FILE` or `less FILE`: View FILE. (? or h for help.)

`cd DIRECTORY`: Change the directory you are in to DIRECTORY.

`cd`: Change to your home directory.

`cd ..`: Change to the next directory up in the hierarchy.

`mkdir DIRECTORY`: Make DIRECTORY.

`rmdir DIRECTORY`: Remove DIRECTORY.

`rm -rf`: Recursively remove a directory and anything in it.

`mv FILE1 FILE2`: Moves or renames FILE1 to FILE2.

`cp FILE1 FILE2`: Makes a copy of FILE1, called FILE2.

`cat FILE1 FILE2 > FILE3`: Concatenate FILE1 and FILE2, calling the result FILE3.

`chmod 644 FILE`: Unprotect FILE for others to read or copy.

`chmod 755 FILE`: Unprotect program or "script."

`chmod 755 DIRECTORY`: Unprotect DIRECTORY, needed for web page directories.

`head` and `tail`: Print the top and bottom of a text file.

## TEXT FILE MANIPULATION

These commands operate on files. The output goes to the "standard output," which is your terminal display. If you want to "redirect" the output to a file,

use `> FILENAME` at the end of the command (with `FILENAME` being the name of the file). Use `>>` instead of `>` if you want to append to the file rather than write it from scratch.

`diff`: Find the differences between two text files.

`grep LETTER-STRING FILE1 [> FILE2]`: Prints (or puts in FILE2) all lines of FILE1 that contain LETTER-STRING. Use the `-v` switch to get lines not containing the string.

`sed s/STRING1/STRING2/g`: replace `STRING1` with `STRING2` throughout a file. See also the `y` switch for replacing characters.

`cut -d" " -f2 FILE`: Extract the second column from a file, where the columns are delimited by spaces. Use `paste` to put such cuttings back together.

`sort FILE`: Sort the lines alphabetically.

`uniq FILE`: Remove adjacent duplicate lines. Typically used with the output of sort, e.g., `sort FILE | uniq`. Use the `-c` to count the number of adjacent examples of each line.

`wc FILE`: Count characters, words, and lines.

## WHAT'S GOING ON?

`w [USER]`: Who is using the computer. This is useful to see whether you are logged on twice. (See ps and kill, below, in case this happens.) Also try `who` and `finger`.

`finger USER`: Gives information about the user, including the files .project and .plan, if you have these files. You can sometimes use this for people on other computers. You can keep useful information in your .plan, such as your schedule, your phone number, etc.

`quota -v`: Tells you how much of your quota for files is used up.

`ps -fu USERNAME`: Lists the processes that you are running, if you put in your username. You can use this to find the number of processes that you want to kill, such as those left over when you did not log out properly. It is the first number listed.

`kill PROCESS-NUMBER`: Kills the process you don't want. If this doesn't work, try `kill -9 PROCESS-NUMBER`.

`last -22`: Shows you the last 22 users who logged in.

## THE INTERNET

`mutt` and `elm`: Read and write electronic mail. Each has its own "help". See the discussion of these in the psychology web page computer section.

`slrn` and `tin`: Read news (and respond to postings).

`Pnews`: Post to newsgroups. (You can also say `mutt psych-general@psych` or `elm psych-general@psych` to post to upenn.psych.general.)

`lynx [URL]`: Reads web pages as text files.

`ssh HOST`: Connects to a remote computer.

`talk USERNAME`: Allows you to talk with someone logged on. The full username must be specified for remote computers, and you must use `ntalk` instead of `talk`. It is not a good idea to use this casually unless you know that the other person will not be annoyed.

## TRANSFERRING FILES

Many computers on the internet have ftp or ssh, but all of the programs for transferring files depend on having the relevant software on both computers.

`ftp and scp`: Fast way of uploading and downloading, or moving any file from one computer to another. For example, `scp myfile baron@psych.upenn.edu:` - you need the colon at the end.

`rsync`: synchronize files or directories on two different computers. Good for backing up.

## EDITING

`pico FILE` or `emacs FILE` `xemacs FILE` or `vi FILE`: Edit FILE. One of these editors may be specified as the default for mailing and news programs. Pico is easiest because it has all the commands listed at the bottom of the screen, but it is the least useful because it has few commands.

## COMPRESSING, ENCRYPTING, and SECURITY

`tar cvf FILE.tar` and `tar xvf FILE.tar`: Create and extract an archive file. Useful for backing up directories. On Linux `xvfz` will extract and unzip in one step.

`gzip FILE`: Reduce the size of files for storage. use `gunzip FILE` to unzip them. Files "zipped" with gzip have the suffix .gz.

`crypt < FILE1 > FILE2`: Encrypts or decodes file1 so that you need a password to read it. But if you want to be absolutely sure beyond any doubt that nobody will read your files, do not leave them on cattell.

`antiword FILE1 > FILE2`: Decodes a Word file into text. (That is all most of them are.) Save storage space. You can use this with Mutt or Elm to read doc files sent in email messages.

`passwd`: Change your password.

Jon Baron, September, 2003