

# More UNIX Commands

- **jobs** --- lists your currently active jobs (those that you put in the background) and their job numbers. Useful to determine which one you want to foreground if you have lots of them.
- **bg** --- background a job after suspending it.
- **fg %jobnumber** --- foreground a job
- **!!** --- repeat the previous command (but **CTRL-p**, is safer, because you have hit return in addition)
- **!pattern** --- repeat the last command that starts with *pattern*
- **echo \$VARIABLE** --- shows the value of an environment variable
- **setenv** --- lets you set environment variables. For example, if you typed a wrong value for the **TERM** variable when logging in, you don't have to log out and start over, but you can just do **setenv TERM vt100** (or whatever). To see what all your environment variables are set to, type **env**. The one that you're most likely to have to set is the **DISPLAY** variable, when using an X-display.
- **unset VAR** --- lets you un-set environment variables. Useful, for example, if you've usually set **autologout** but want to stay logged on for a while without typing for some reason, or if you set the **DISPLAY** variable automatically but want to avoid opening windows for some reason.
- **source filename** --- you need to source your dotfiles after making changes for them to take effect (or log off and in again)
- **load** --- will show you the load average graphically
- **ispell filename** --- will check the spelling in your file. If you're running it on a LaTeX file use the **-T** option to tell it to ignore the LaTeX commands. You can create and use your own dictionary to avoid having it tell you that your own name, those of fellow linguists, and linguistics terminology are a typos in every paper you write.
- **weblint** --- checks the syntax of html files
- **latex2html** --- translates LaTeX files into HTML
- **wn word option** --- lets you access the WordNet database and display, for example, synonyms, hypernyms, or hyponyms, depending on the option you select

## Command editing in the tcsh

**These things are the same as in emacs:**

**Backspace** --- delete previous character

**CTRL-d** --- delete next character

**CTRL-k** --- delete rest of line

**CTRL-a** --- go to start of line

**CTRL-e** --- go to end of line

**CTRL-b** --- go backwards without deleting

**CTRL-f** --- go forward without deleting

### Other useful things

**TAB** --- complete filename or command up to the point of uniqueness

**CTRL-u** --- cancel whole line

**CTRL-p** --- show the last command typed, then the one before that, etc.

(you can also use the cursor up key for this)

**CTRL-n** --- go forwards in the history of commands

(you can also use the cursor down key for this)

**CTRL-c** --- cancel the processes after it has started

**CTRL-z** --- suspend a running process (e.g. in order to do something else in between)

you can then put the process in the background with **bg**

**CTRL-l** --- redraws the screen

| (**piping**) --- Lets you execute any number of commands in a sequence.

The second command will be executed once the first is done, and so forth, using the previous command's output as input. You can achieve the same effect by putting the output in a file and giving the filename as an argument to the second command, but that would be much more complicated, and you'd have to remember to remove all the junkfiles afterwards. Some examples that show the usefulness of this:

**ls | more** --- will show you one screenful at a time, which is useful with any command that will produce a lot of output, e.g. also **ps -aux**

**man ls | grep time** --- checks whether the man page for **ls** has something to say about listing files by time - very useful when you have a suspicion some command may be capable of doing what you want, but you aren't sure.

**ls -lR | grep dvi** --- will show you all your dvi files - useful to solve disk space problems, since they're large and usually can be deleted.