

# Easy Unix for Gov1000

Tao Li

August 18, 2003

## Thanks

The author thanks Dan Epstein and Kosuke Imai for extensive comments. Please send your comment directly to [li7@fas.harvard.edu](mailto:li7@fas.harvard.edu). Thanks a lot!

## Logging In

Once you find a **login:** prompt, type your *username*, followed of course by hitting the “return” key. Wait for the **password:** prompt, and type your *password*. Notice that to keep it secret, your password does not appear on the screen. To change the password, type **passwd** and follow the instructions.

In UNIX, “quit” does not have the connotation of an ungraceful end that it has in normal English; it is the normal way out. When a hasty retreat is desired, some programs (such as mail) offer an **exit** command.

## The Unix File System

The UNIX file system is tree-structured, meaning that any directory can contain other directories, as well as ordinary files, programs and links (pointers) to files in another part of the directory tree. Sub-directories can be nested to any depth.

Each process has a *working directory*, which is where unix looks by default for any files you command it to work with. When you log in, the working directory is simply your *login directory*. The command to display the path from the root directory to your working directory is:

```
pwd
```

Try this. You should get a response along the lines of `/gov1000/students/username`, which is the path from the root of the directory tree to the login directory for your student account.

## File and Directory Names

To refer a file in the working directory, just give its name. Names can be any number of characters up to some (ridiculously large) limit. Uppercase and lowercase are *distinct* in file and directory names. Almost any character can be used in file names, although characters with a special meaning to the command interpreter may have to be quoted.

Appending a / and another name, which can be done to any number of levels, means that the name before the / is expected to be a directory, and the name after the / is to be sought in that directory. The following example might help you to understand what I mean. For example, suppose under your login directory, there exists a directory called “homework”, under “homework” there exists a subdirectory called “set1” and there is a file called “UNNAMED.tex” under it. To refer the file “UNNAMED.tex” under your login directory, you need to specify the whole path leading to the file.

```
homework/set1/UNNAMED.tex
```

means that a directory named `homework` will be sought in your working directory (now is your login directory), and it will contain a directory named `set1`, which in turn contains a file named `UNNAMED.tex`. Basically you tell the computer how to find a file since otherwise the computer will search it fruitlessly under your working directory only.

Each directory contains two special entries:

```
“.” is that directory itself,  
“..” is that directory’s parent directory.
```

As a special case, to refer to a file using the entire path, from the root of the file tree all the way to the file and directory in which it’s located, begin the name by placing a backslash before the first directory name. For example:

```
/gov1000/students/student1/homework/set1.UNNAMED.tex
```

To use such file specifications would require that you know the path from the root of the file system. Instead, if the first character is `~/`, the name following is sought as a login directory. The following filename:

```
~/student1/homework/set1.UNNAMED.tex
```

is equivalent to the preceding one.

## Directories Commands

The all-purpose command for displaying the contents of a directory is `ls`. `ls` with no arguments lists the contents of your working directory. `ls` with a directory name argument lists the content of that directory.

Many UNIX commands come with *options*, which are often distinguished by being preceded by a `-`. For example:

```
ls -a
```

will show otherwise invisible files and directories such as `."`, `."`, and others.

```
ls -F
```

displays files and directories so you can distinguish them and their types. A `/` after the name indicates a directory, a `*` an executable file (program).

```
ls -l
```

provides a long (detailed) form of directory listings that includes the sizes, ownership, dates, and protection modes of the files and directories. Options typically precede file or directory names in UNIX commands. Options can usually be combined, for example `ls -al`. You could also type this as `ls -a -l`.

You might try what happens when you type `ls`, `ls -aF`, `ls -al`, `ls ..`, `ls /.`. You can probably explain why when you type `ls ..` from your login directory, you get a list of the usernames of all the other students with accounts.

To see all the process you are currently running, use the command `ps -u username`.

The commands for creating and removing directories are:

```
mkdir name      create empty directory
rmdir name      remove empty directory
```

You might try creating and removing some directories. What happens if you try to create a directory with a name that already exists?

You can change the working directory with the command

```
cd name
```

where **name** is a directory name. `cd` with no arguments sets the working directory back to your login directory.

You can create as many directories within your login directory as you like, and delete any file within it. To minimize clutter, the files and directories that you use for different courses, projects, and other activities should have their own directories.

## Copying, Renaming, and Deleting Files

If you want a separate copy to modify, use:

```
cp fromfile tofile
```

which copies `fromfile` to `tofile`. To rename a file or directory use:

```
mv oldname newname
```

Notice that the arguments of all of these commands first specify the existing file(s), and then the destination. If the destination is a directory, the file(s) will be copied/moved into that directory. For example, if there is a file called `xyz.tex` under your working directory, you could:

```
mkdir save
cp xyz.tex save
```

which would make a copy of your `xyz.tex` file in a directory called `save`, and the new file would be `save/xyz.tex`. That is, the effect is just like:

```
mkdir save
cp xyz.tex save/xyz.tex
```

You can remove (delete) files with:

```
rm filename
```

If you add `-r` after `rm`, it will recursively remove a directory, first removing the files and subdirectories beneath it. If you add `-f` after `rm`, the system will not ask for your confirmation of removal. This is the default setting for some system.

When you remove a file it's really gone, and the only way to get back a deleted file is to ask us to retrieve an old version from backup tapes (which are made every day, and kept for a few months). Nevertheless, to forestall disk space exhaustion, you should remove files that you no longer need.

## Filename Wildcard

The asterisk `*` in unix is used as a wildcard, representing any string of characters (except "Red Sox," apparently). To refer to all files beginning with, for example, the three letters 'buf', type `buf*`. For example,

```
ls buf*
```

generates a list of all filenames starting with the string `buf`. The list is substituted for `buf*` in the command line, and the `ls` command then displays the size, date, protection, and names of those files. What we (purposely) neglected to mention above is that many commands, such as `ls` and `rm` will accept a list of one or more filenames as arguments.

Of course, powerful commands with wildcards must be used with care. Suppose we were meant to type:

```
rm proj/buf*
```

but instead slipped and typed:

```
rm proj/buf *
```

The result would be to remove `proj/buf`, and then to remove `*`, all of the files in the working directory! They who hesitate before hitting return are often saved.

## Finding Documentation

At this point it gets a bit tedious to describe all the variants of the more complicated programs. To find out more about the relatively more complicated programs mentioned below, such as `ftp` (file transfer program), `chmod` (change mode), and others, you should read through the corresponding manual entries.

```
man latex
```

Type `q` to quit the help page. To get a the “one-liner” of a program, say, `LATEX`, you can type:

```
whatis latex
```

## File Protection

Usually only the file’s owner (the user who created it) can modify it, but anyone can read it. The protection can be changed for individual files with `chmod`, or the `umask` command can be used to change the default protection for all files created by that process. The default allows read access to all so that you might have more opportunities to learn and share, this being a university. However, if anyone complains that you are snooping on them, you will be fully responsible for your actions. Use `man` to learn more about them. This command is relatively advanced for a beginner.

## Text Editor

The text editor for Gov 1000 is Emacs. Emacs, like many other unix commands that we will use, opens up a separate window for itself when you type the command “`emacs`”. For such programs, if you just type the command name at the shell prompt, then the shell prompt is locked until you close that other window. To avoid this, type “`emacs &`”. Use the `&` sign after all such commands that open a separate window.”

## Displaying and Printing Files

Once you can edit files, you will likely want to be able to display them or parts of them, and to print them. The usual programs for displaying files on a terminal or workstation window are `more` or `less`, followed by the file name. Use `man` to find the appropriate options.

To print a file use:

```
lpr file
```

Most time you need to specify the printer you are going to use. This is made available through the option `-P printer’s address` following the command `lpr`. You need to ask a local expert for the printer’s address.

There are also a number of commands related to `lpr` for checking the print queue (`lpq`) or for removing items from the print queues (`lprm`). `ipstat` is another text-printing program used to show the what print jobs are waiting.

The command `grep` can be used to search for the specific word in a file. For example, under the working directory you have a file called `latex.tex` and you want to search the word “verb” in the file. Type:

```
grep verb latex.tex
```

## Compressing and Combining Files

The `gzip` command is used to reduce the amount of disk space utilized by a file. For example,

```
gzip latex.ps
```

will produce a file called `latex.ps.gz`, which uses much smaller space than the original file. You can get your original file back by typing

```
gunzip latex.ps.gz
```

The `tar` command combines files into one big file for archiving purposes. It does not compress the files; it merely makes a large quantity of files more manageable. For example, if your working directory contains two files, `test1.ps`, `test2.ps` and you type,

```
tar -cvf test.tar *
```

you will create a file called `test.tar` which is a combination of `test1.ps`, `test2.ps`. To get the original files from `test.tar`, type

```
tar -xvf test.tar
```

## X Window System

The X Window System, generally known as “X” or “X11”, is the windowing interface we now use. X pretty much speaks for itself. The main thing you have to do to get used to it is to press the different mouse buttons when the cursor is in different parts of the screen, i.e., the background, in a window, in

the window's control bar, or its corners. Pressing the control key while pressing a mouse button within a window provides extra interesting alternatives.

You can run any program in the background by appending a `&` to the end of the command and its parameters. This is most useful when the program pops up in its own window and you still want to use your original one. Type "`netscape &`" at a prompt to open Netscape. Type "`xterm &`" to open a new xterm window.

## Remote Connecting and File Transfer

For example, you want to connect to your `fas.harvard.edu` account `xxx`. Type

```
ssh xxx@fas.harvard.edu
```

and then input your password. If you want to transfer a file called `test.ps` from your course working directory to your fas account, you can do the following

```
sftp xxx@fas.harvard.edu
```

and then input your fas password. You will then see a prompt like this

```
ftp>
```

Type `put test.ps`. The file `test.ps` will be transferred from your course account to your fas account. On the other hand, if there is a file in your fas account which you want to transfer to your course account working directory, do the same procedure and use the command `get filename` for the last step.

To transfer a file/directory from your PC to the unix system (or the other way around), you need to use the program called SecureFTP. It is installed in the lab machine.

## Unix commands for L<sup>A</sup>T<sub>E</sub>X

Please refer to my introduction to L<sup>A</sup>T<sub>E</sub>X programming for Gov 1000 first. Once you have created a file called, say, `minimalism.tex` using Emacs editor and saved it. Type under the prompt

```
latex minimalism.tex
```

If everything is fine the prompt will eventually return. Now your code has been compiled and a file `minimalism.dvi` has been created. You can view this file by typing `xdvi minimalism.dvi` or you can create a “.ps” file by typing

```
dvips -o minimalism.ps minimalism.dvi
```

You can create a “.pdf” by typing

```
ps2pdf minimalism.ps minimalism.pdf
```

Or you can type `pdflatex minimalism.tex` in the very beginning instead of typing `latex minimalism.tex`.

To view these files, type `gv minimalism.ps` or `ghostview minimalism.ps`. PDF file can be viewed similarly.