

# Using Unix

Tom Kelliher, CS 318

Jan. 25, 2002

The purpose of this document is to get you up to speed quickly on the basics of Unix — enough to get started. Becoming a Unix guru requires years of devotion to the “one true way.” For more information, refer to the plethora of links on the class home page. I can also give recommendations for Unix books to those interested.

## SSH'ing to Phoenix

On the lab machines, look under the Programs menu for the Telnet category and choose Secure Telnet (which is really TTSSH). In the new connection dialog box, set host to phoenix.goucher.edu and choose the SSH service. Within the following dialog box, enter your username and phoenix password.

To access phoenix from your own computer, you'll need to install an SSH client. Follow the SSH link on my home page for the TTSSH client and instructions.

## Changing your password

The password I've given you is one only a computer could love. Well, it was generated by a computer, so what do you expect? You'll probably want to change it. Use this command from a command prompt: `passwd`. (Uh, that period ends the sentence, it's not a part of the command. Watch out for this in the following examples, too. OK?)

## Editing Files

Three editors are available: `pico`, `vi`, and `emacs`. `Pico` is the easiest to use. It will remind you of `wordpad`. It's the least powerful of the three, meaning you'll quickly outgrow it and you will then curse it the rest of your miserable days. Discriminating users use either `vi` or `emacs`. The enlightened use `emacs`.

## Changing Your Environment

1. Edit the file `.login` in your home directory (for instance, type `pico .login`). Find the line `setenv PRINTER pclab` and change `pclab` to `black`. Save the file and exit the editor. Files you print will now be printed to the printer in the X Lab.
2. Edit the file `.cshrc` in your home directory. Start `pico` with the `-w` switch to disable line wrapping (`pico -w .cshrc`). Find the line which begins `set path` and add `/usr/local/pgsql/bin` immediately after `/usr/local/bin` to the list of directories. Make sure this line remains a single line! Once you make this change you'll be able to access various PostgreSQL utilities.

Log out then log back in for your changes to take effect. To log out use the `logout` command.

## Files, Directories and Navigating the Filesystem

You always start out in your home directory when you log in. The command `ls` is used to list the files in your current directory. `ls -l` will give you details. The `cd` command is used to change your current directory. For instance, the web server looks in your `public_html` directory for web-related files. To enter that directory, type `cd public_html`. System configuration files are in `/etc`. To go there just type `cd /etc`. To go back home from anywhere (yes, you can go home again) just type `cd`.

If you create a web-related file and the web server gives you “Permission denied” errors, you’ll need to change the access permissions on the file using `chmod`: `chmod go+r <file>` (where `<file>` is replaced with the actual file name, but you already knew that).

To create a new directory, use `mkdir <directory_name>` Use `chmod go+rx <directory_name>` to allow others access to a directory you just created. Use `rmdir <directory_name>` to delete a directory.

Use `rm <file>` to delete a file. Once a file is deleted, it can’t be brought back again, so be careful!

## Printing Files

Type `lpr <file>`, then run to the printer in the X Lab. If you want to print to your own printer from phoenix first install Linux and then come see me.

## Running PHP from the Command Line

If your PHP script isn’t getting along with the web server you can run your script from the command line: `php -q <script>`. Any error messages will be displayed for your viewing pleasure and debugging nightmare. If you care to see the error messages produced by the web server, use the `less` command: `less /var/log/httpd/error_log`. Type `G` to get to the end of the file once `less` is running.

## Exercise

Write a pair of web pages with embedded PHP scripts. The first PHP script should generate a form which allows the user to input 10 numbers. When the form’s submit button is pressed, the 10 numbers should be sent to the second PHP script as an array. The second script should display the largest number in the array and the average of the 10 numbers.