COLBY COLLEGE
Department of Mathematics
and Computer Science

CS357 **Allen B. Downey**
Spring 1998

Assignment 1: UNIX intro

Due: Tuesday, February 17

The purpose of this assignment is to introduce you to the UNIX environment we will be using, to get you to practice using some of the common UNIX commands, and to get you started on learning some C++.

Log in

I am writing these instructions with the assumption that you are sitting at one of the SGIs in the Mudd Lab or the Schupf Lab. You can also log in from a Macintosh, Wintel or Linux machine, but I will assume if you are using one of those systems that you know how to create windows and connect remotely.

1. Get a piece of paper from me that has your account name and password on it. The account names are in the form os1 through os25. These accounts exist on the SGI machines with the names iris4 and iris5 (the machines in the Mudd Lab), iris6 through iris11 (the workstations in the Schupf Lab) and iris12 (the Power Challenge in the Schupf Lab).
2. Log in to the machine. The first time you log in, the window manager may tell you that you have no environment set up, and it will ask you if you want to create one. You do.
3. The default environment will create a window with lots of icons in it. If you figure out what it's good for, let me know. Otherwise, just close it.
4. From the desktop menu, select "UNIX Shell." It will create a window with the standard UNIX command-line interface. Commands you execute in this window will run locally, on the machine you are sitting at.
5. Also from the desktop menu, select "iris12." This will create a remote login window, and automatically log you into iris12. You should not have to type your password; if you do, there is something wrong with the setup and you should let me know. Commands you run in this window will run on the Power Challenge, which means that they will run much faster, unless iris12 is very heavily loaded.
6. Your home directory, which is where you will do all your work, is available from all the SGI machines. Thus, most of the time it does not matter whether you execute a command locally or remotely.

7. In the local window, type `netscape &` to start a web browser in the background. After a few seconds, the browser window will appear. Load the following URL: `http://rocky.colby.edu/cs357`.

8. Read the course information there, if you haven't already, and then go to the UNIX help index. You will probably want to bookmark this page.

9. Find the documentation of the `pwd` command and read all about it. Run it to find out the complete path of your home directory. Make a note of it.

Get the C++ manual

1. In the netscape window, go back to the cs357 page and select `C++ information` and then select the compressed Postscript version of the `Quick introduction to C++`.

2. A window will appear asking you where you would like to put the file. Check to make sure that it is planning to put it in your home directory, and hit OK.

3. In the local window, run `ls` to see if the new file `c++.ps.gz` has appeared in your home directory.

4. Type `gunzip c++.ps.gz` to uncompress the file. The suffix `.gz` will disappear. I will assume from here on in that any time you use a UNIX command you are not familiar with, you will consult the unixhelp web page and read all about it.

   I am not sure if all the SGIs have `gunzip` installed. If yours does not, you can run this command on iris12, or you can download the uncompressed Postscript version.

5. Print the file. On iris{4,5}, use `lpr`; for example `lpr -Pmuddlab c++.ps`, where `muddlab` is the name of the printer in (you guessed it) the Mudd Lab. On iris{6-12}, use `lp` instead, as in `lp -PHP c++.ps`. HP is the name of the printer in the Schupf Lab. Capitalization matters.

Set up the C++ example

1. Using netscape again, pick up the compressed version of the `stack example from the handout`.

2. Uncompress the resulting file, then unpack it using `tar`. The command is `tar -xvf c++.tar.gz`. It will print a list of the files it is unpacking and create directory named `c++example`.

3. Use `cd` to move into the c++example directory, and type `ls` to see what's there. Now type `make stack` to try to compile the program. It will fail, and report that the command `CC` was not found. `CC` is the C++ compiler, but it does not exist on all the SGI machines. For now, it only exists on iris12.

4. Move into your iris12 window and `cd` into the c++examples directory. Type `make stack` again. This time it should succeed.

5. Run the newly-compiled executable named `stack`. It should print a bunch of numbers from 17 to 26 and back to 17.

6. Stop here and read the C++ handout, referring to the examples you just set up. The UNIX commands to read the contents of a file are `cat` and `more`.

Check out your environment

1. When you log in to your UNIX account, the system creates an environment for you and initializes a set of environment variables. These variables are used by various programs to control their behavior. For example, the PRINTER environment variable tells programs the name of the default printer.
2. Type `printenv` to see the environment variables the system has set for you. To change an environment variable, use `setenv`. For example, if you are in the Mudd Lab, you probably want to `setenv PRINTER muddlab`. In the Schupf Lab, `setenv PRINTER HP`.
3. If you set the PRINTER environment variable, you can omit the `-P` option when you run `lp` or `lpr`.

Teach yourself a little emacs

1. Emacs is the text editor we will be using to compose source code. If you already use vi and would rather use it instead of emacs, you may, but keep in mind that I don't know vi at all, so if you have a problem I can't help you.
2. If you run emacs in a local window, you have the option of running it in the background (`emacs &`), in which case it will create a new window for itself.
3. If you run emacs on a remote host (like iris12), you will want it to run in the existing window. To make sure that happens, check that the environment variable DISPLAY is not set. If it is, type `unsetenv DISPLAY`.
4. Once you have started emacs, type Control-h followed by a t. A document will appear that will teach you how to use emacs.
5. There is an emacs quick-reference sheet on the cs357 web page. Pick up and print a copy if you want.

Something to hand in

1. Use emacs to compose an answer to questions 2.1 and 2.9 from the textbook. Your answer to each question should be within a factor of two of 250 words.

   This is not just an exercise in using emacs, it is also a chance to think about some of the issues in the chapter and practice technical writing. Grammar and spelling count, but clarity and conciseness are critical. Think carefully about what you want to say before you write.

   When evaluating alternative designs, it is standard to explain the pros and cons of each alternative and then suggest under what circumstances each alternative would be better.

   Please print your answers, make sure your name is on all pages, staple the pages together, and hand them in.