



Computer Science 507 Software Engineering

The College of Saint Rose
Spring 2013

Lab 1: Unix Introduction/Refresher

Due: 6:00 PM, Monday, January 28, 2013

In this assignment, you will learn or refresh your knowledge of some Unix commands. You may work alone or with a partner on this assignment.

None of these tasks is intended to be hard, but if you don't have much Unix experience (and it is reasonable if you don't), they could be. So...ask if you are not sure what to do! Plenty of questions are expected.

The Unix Command Line

To get started, log into `mogul.strose.edu`. If you are on a College laptop, you should see the program “putty” on your desktop which you can use for this purpose. Mac users can open a Terminal window and use `ssh` to connect.

In your favorite editor, create a document in which you will answer the questions you find in this section. Start by putting your name at the top of this document.

GUIs are nice, but they can be slow to navigate and too restrictive for some purposes. For some of our examples and assignments, you will be working in a Unix environment and interacting with the system by typing commands at the Unix *shell*, or *command line*. When you log in, you will be presented with a prompt. This is your direct interface to issue commands to the operating system. When you type a command here, the shell will execute the command on your behalf, print out any results, then reissue the prompt.

Of course, the command line is useless if you don't know what commands it understands. You will learn about several important commands in this lab and many more throughout the semester. One of the most important is `man` – the Unix manual. Every Unix command has a manual page, including `man`. To see the manual page about `man`, type the command:

```
man man
```

Navigating the Directory Structure

You have very likely used systems where files can be organized into *folders*. When accessing these kinds of structures from the command line, we usually refer to them as *directories*. Each program in a Unix system, including your shell, maintains the notion of a *working directory*. That is where the program will look for files unless instructed to do otherwise. You'll hear Unix users asking a question like “What directory are you in?” and the answer to this is your working directory.

When you first open a shell, your *home directory* is your working directory. The command `pwd` will instruct the shell to print your working directory.

Question 1: What is your home directory on `mogul.strose.edu`? (use `pwd`)

You can also list the contents of your working directory with the command `ls`.

Question 2: What output do you see when you issue the `ls` command on `mogul.strose.edu`?

Other important operations to navigate and modify the directory structure are changing your working directory (`cd`), creating a new directory (`mkdir`), and removing a directory (`rmdir`).

Create a directory in your account for your work for this course (`cs507` might be a good name), and a directory within that directory for this assignment (`lab1` might be a good name).

Question 3: Change your working directory to the one you just created and issue the `pwd` command. What does this show as your working directory?

In your shell window and in your home directory (note: you can always reset your working directory to be your home directory by issuing the command `cd` with no parameters), issue this command:

```
uname -a > linux.txt
```

This will execute the command `uname -a`, which prints a variety of information about the system you are on, and “redirects” the output, which would normally be printed in your terminal window, to the file `linux.txt`.

Look at the contents of the file `linux.txt` with the command:

```
cat linux.txt
```

Question 4: What do you think this information means?

Unix Commands

Identify the function of and experiment with these Unix commands (a few of which you have already used):

```
ls      cd      cp      mv      rm      mkdir   pwd
man     chmod   cat     more    grep    head    tail
ln      find    rmdir   wc      diff    tar     touch
```

Question 5: Give a one sentence description of each command.

Using appropriate commands from the above list, move the `linux.txt` file you created in your home directory into the directory you created for your work for this assignment.

Show that this has worked by issuing the following command from inside of your course directory (but not inside the directory for this assignment):

```
ls -laR > ls.out
```

Then move the file `ls.out` into the directory for this assignment.

Using the Unix manual, your favorite search engine, or in discussion with your classmates, determine the answers to these questions:

Question 6: How do you change your working directory to be “one level up” from the current working directory? (Give the command.)

Question 7: Give three different ways to change your working directory to be your home directory. All likely involve the `cd` command, but will take different parameters.

The Emacs Editor

Emacs (emacS from the Unix command line) is a powerful text editor. You will want to become familiar with it. Identify the function of and experiment with these Emacs commands:

```
C-x C-s    C-x C-c    C-x C-f    C-x C-w    C-g    C-a    C-e
C-d        C-_        C-v        M-v        C-s    C-r    M-%
```

`C-` before a key means hold down `Ctrl` and hit that key. `M-` indicates the “Meta” key, which on most systems is `Esc`. To issue a Meta command, hit the `Esc` key, release it, then hit the key for the command you wish to issue. Use the keystrokes rather than the menus. It will save you time in the long run!

Include a brief description of each Emacs keystroke in a file `emacs.txt`. Use Emacs to create and edit this file.

Learn these commands – you will use them often. Hints can be found in the Unix and Emacs web pages linked from the course website. Ask others if you’re not sure how to do something in the Unix environment. Share your tips and tricks with each other.

Submission

Transfer the files you have created for this lab to the computer you’re working on with a secure copy program like WinSCP or FileZilla.

To submit the assignment, send all necessary files (your document with the answers to the questions, your files `ls.out`, `linux.txt`, and `emacs.txt`) as attachments to terescoj@strose.edu by 6:00 PM, Monday, January 28, 2013.

Please include a meaningful subject line (something like “CS507 Lab 1 Submission”). Please do not include any additional files, such as emacs backup files, object files, or executable programs.

Grading

This lab will be graded out of 15 points.

Grading Breakdown	
linux.txt file	1 point
ls.out with correct directory structure	1 point
Unix command descriptions	4 points
Emacs command descriptions	4 points
other question responses	5 points