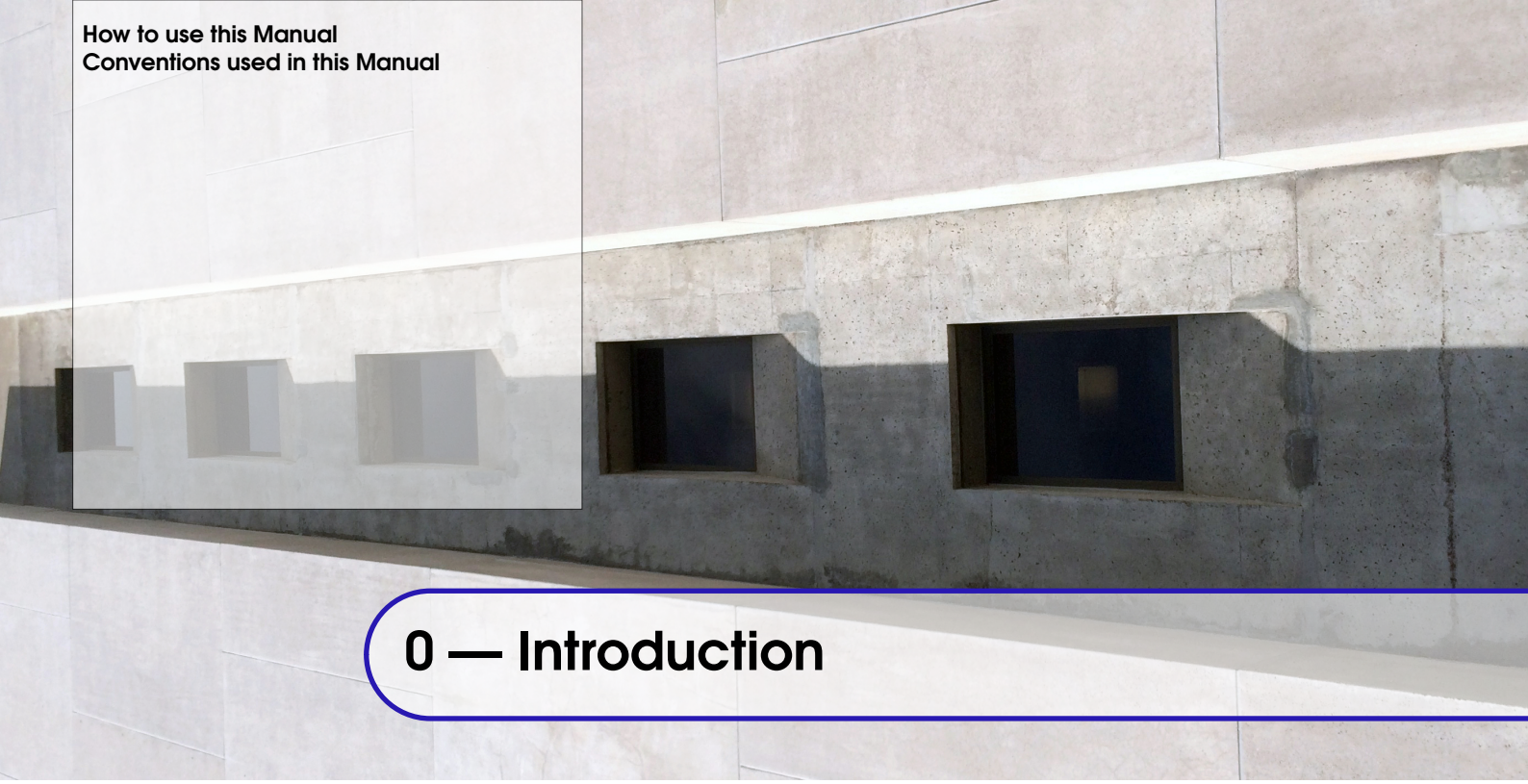
Contents



0 — Introduction

Welcome to CSC412, the Advanced Software Lab. This lab course is designed to give you hands-on experience with technologies and techniques that you may not have seen in other computer sciences courses taught at San Francisco State University. As this is a lab course, you will experience these subjects through practical, hands-on work.

This class covers a wide range of software, tools, and technologies, so the treatment of each subject must necessarily be brief. You are encouraged to learn more about all the topics covered in this course. References to printed and web resources will be provided with each lab to help direct you to more information about the topic covered.

All of the technologies covered in this course are applicable to web design to some degree. In fact, by doing the coursework you will implement a website that introduces yourself professionally, a useful thing to have when job seeking. The work that you do in this course will help prepare you for CSC 648, the Software Engineering capstone course required of all majors in the program.

The current version of the lab manual has been written with examples for Apple's OS X and Microsoft Windows 7. If you are using a UNIX-based operating system (such as Ubuntu), you will most likely be able to follow the instructions for OS X.

Your feedback is always welcome and encouraged. Feel free to ask questions during class, talk to the instructor during office hours, or email the instructor. Your feedback helps the instructor and department create a better class.

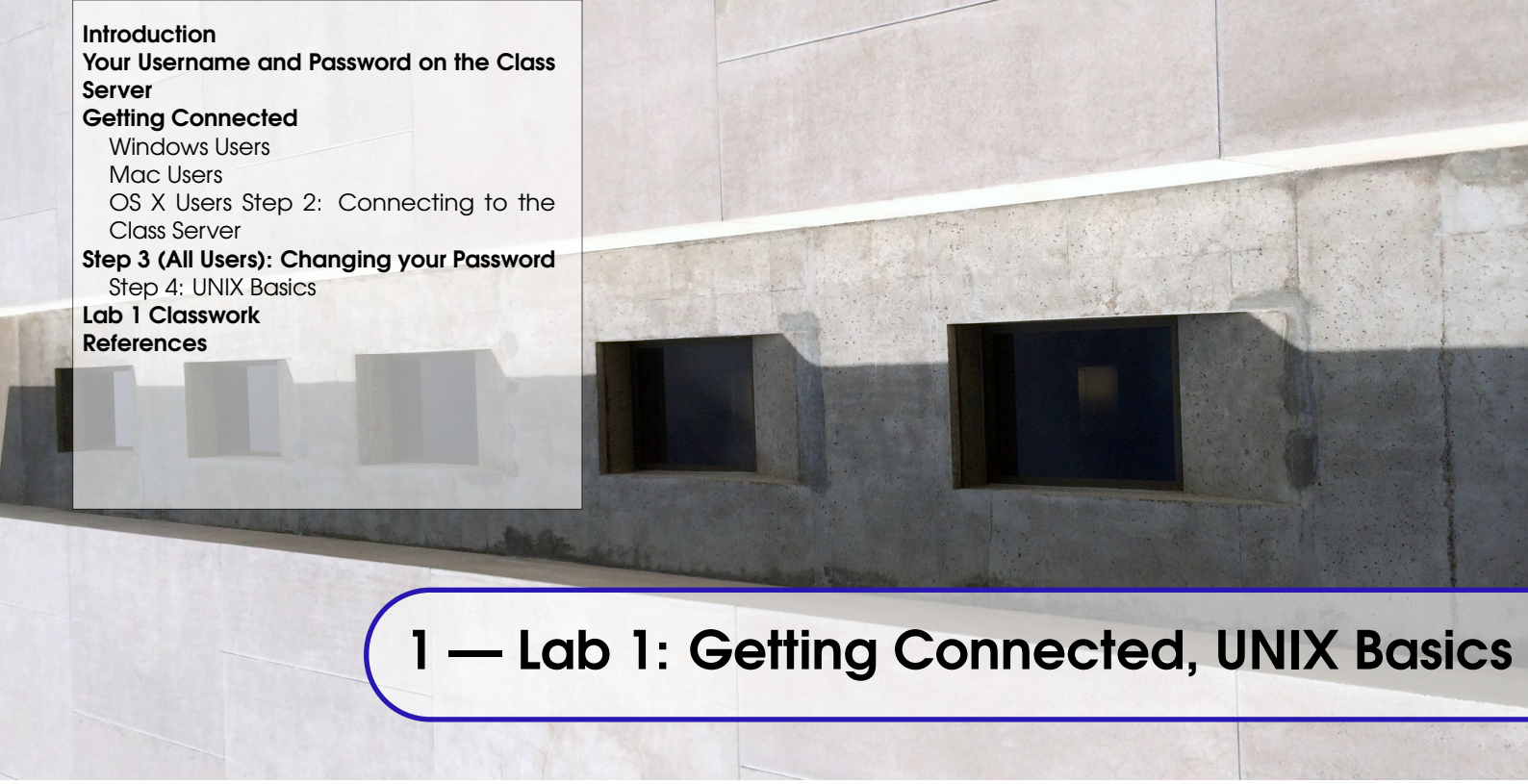
0.1 How to use this Manual

Each week you will be assigned a lab. Each lab is broken into an introductory section at the beginning, and practical work at the end. Though you are tempted to just jump ahead right to the practical work, it is important to read through the introductory material as well. Your instructor may include worksheets that you are required to fill out for lab credit, and these will involve this reading.

0.2 Conventions used in this Manual

A number of typographical conventions are used in this manual:

- Console text is written in `black fixed-width sans-serif font`.
- Keyboard entry by you, the user, is written in a `blue fixed-width san-serif font`. This is text that you are expected to type into the computer.
- Directories are represented `with‣small‣arrows‣between‣them`.
- Menu paths are indicated by boxes › with › arrows
- The ⏎ character represents the return or enter key being pressed by the user.
- A shadowed box will surround single keys that need to be described with words, for example Space.
- URLs and email addresses are written in a clickable fixed-width serif font, e.g. `http://cs.sfsu.edu`.
- ⌘+key (on Macs only) means you press and hold the key marked "cmd" or "command", then simultaneously press the key indicated by key, then release both keys. For example ⌘+M is read "command m" and is executed by first pressing and holding the command key, then pressing the letter m while the command key is still pressed, then releasing both keys.
- ctrl+key means you press and hold the key marked "ctrl" or "control", then simultaneously press the key indicated by key, then release both keys. For example ctrl-m is read "control m" and is executed by first pressing and holding the control key, then pressing the letter m while the command key is still pressed, then releasing both keys.
- ⇧ means press the shift key while performing the other keystrokes. For example ⇧+⌘+M is read as "shift command m" and is executed by holding the shift key down while pressing and holding the command key down, then pressing the m key, and then releasing all keys.



1 — Lab 1: Getting Connected, UNIX Basics

1.1 Introduction

During the semester you will be doing work using both your own individual computer and on a class computer. In fact, we all will be working on the same class computer simultaneously. To allow more than one of us to use the computer simultaneously, our class computer is configured as a *server*, and we will refer to this computer as the *class server*.

A *server* is a computer system (consisting of hardware and/or software) that provides services to *clients*. Clients interact with the server using a *protocol*. A protocol is an agreed upon set of rules that define how communications between a server and a client are carried out. This arrangement is known as the *client/server model*, or *client/server architecture*.

We will be running client software on our individual computers that communicates with the class server via the SSH (Secure Shell) protocol. This protocol allows us to connect to the class server, and interact with the class server in a *UNIX shell*.

A shell is a *command line interpreter* that provides a text-based interface to the operating system. The user may type text commands via the keyboard, and the shell executes commands via the operating system in which the shell is running. The commands the user enters at the command line are actually programs that are run in the operating system. We will be interacting with the class server through a shell account.

In this lab we will be installing the necessary client software on our own individual computers to allow us to communicate with the class server. We will then connect to the class server, and look around our account. We will also change our password from the default class server password.

1.2 Your Username and Password on the Class Server

The class server is located at `http://csc412sfsu.com`. It is a Amazon Web Server. Your username on the server is the part of your SFSU email address before the @ sign. So, if your

SFSU email address is jdoe@mail.sfsu.edu, your username on Amazon Web Services(AWS) Server will be jdoe. Your initial password on the AWS-Server is your SFSU student ID number.

Your account might not yet be setup. If by following the next steps you find you cannot login, please contact the instructor immediately

1.3 Getting Connected

Before you begin, you must make sure that your computer is connected to the Internet. If you are on campus, be sure to log into the campus network before you begin.

The following instructions are initially different for OS X and Windows users. After logging into the AWS-Server on either operating system the instructions are then the same for both systems.

> Mac users please skip ahead to OS X Users Step 1: Accessing the Command Line.
> Windows users continue at Windows Users Step 1: Installing Client Software.

1.3.1 Windows Users

Step 1: Installing Client Software

To communicate with the class server on a Windows machine, you need to install client software. The client software that you will use on Windows is called PuTTY. PuTTY is actually a suite of programs providing terminal emulation and file transfer functionality. You will need to download two programs: PuTTY (the SSH client) and PSCP (a secure file copy program).

These instructions have been written using the Chrome browser. Specific details may vary depending on the browser that you use.

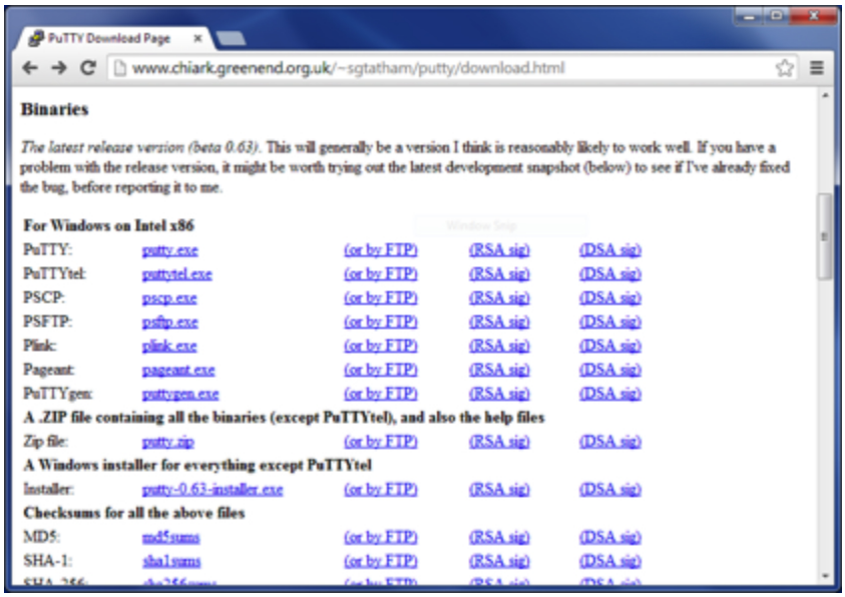


Figure 1.1: PuTTY download site.

1. Using an internet browser of your choice, browse to `http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html` (Figure 1.1).

2. Scroll down until you can see the programs under the “For Windows on Intel x86” heading.
3. Right-click on the “putty.exe” link.
4. Select “Save link as...”

Figure 1.2: PuTTY download site.

5. Save it as putty.exe (as type Application) on your desktop. The putty icon should appear as in Figure 1.2.
6. Save it as pscp.exe (as type Application) on your desktop.
7. You should now have two new program icons on your desktop, for putty and for pscp, which should look like Figure 1.2.

Windows Users Step 2: Connecting to the Class Server

Before you begin using PuTTY, if your system is running a firewall, be sure that you open port 22 outbound for the PuTTY programs.

1. Run the putty program by double-clicking on the putty icon on your desktop.

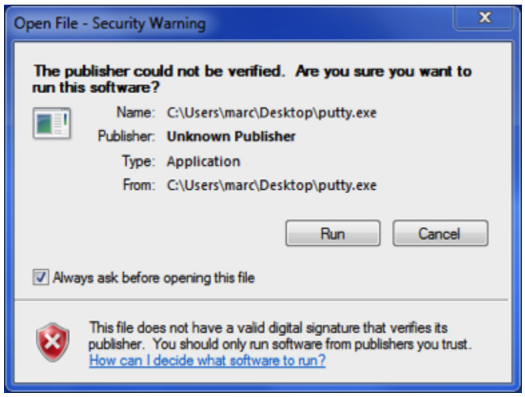


Figure 1.3: Open File Security Warning window.

2. An "Open File - Security Warning" may appear saying that the publisher could not be verified (Figure 1.3). If the security warning appears:
   (a) Uncheck the "Always ask before opening this file" checkbox.
   (b) Click on Run to continue.
3. You will then see the “PuTTY configuration” window (Figure 1.4).
4. In the “Host Name” field, type csc412sfsu.com↵ (Recall the ↵ character represents the return, or enter key).
5. You may see a “PuTTY Security Alert” window indicating that the server’s host key is not cached (Figure 1.5). If you see this alert, click on Yes to continue.
6. A terminal window labeled “csc412sfsu.com- PuTTY” will appear, with the words “login as:” at the top of the window Figure 1.6). You are now connected to the class server. Select this window by clicking on it.
7. Type your username (see Your Username and Password on the Class Server) and press enter.

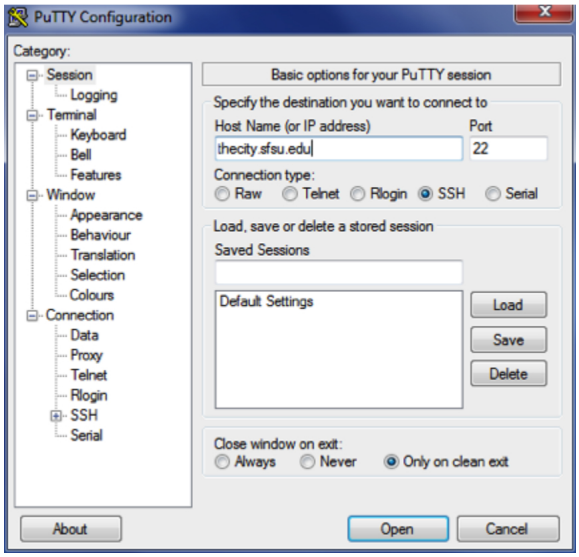


Figure 1.4: PuTTY Configuration window.

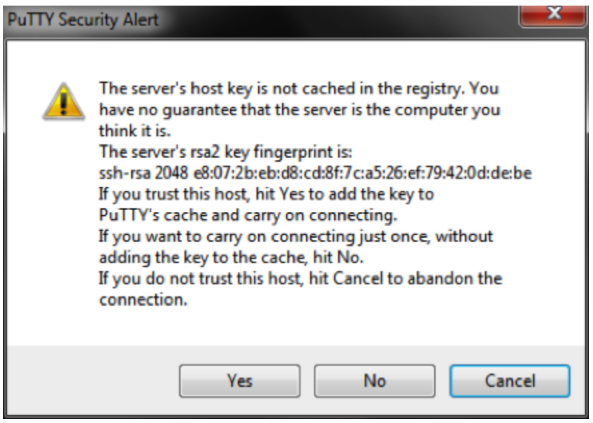


Figure 1.5: PuTTY Security Alert window.

8. You will then be prompted for your password (see Your Username and Password on the Class Server). Type your password and press enter.

Now that you're connected to the class server, the first thing to do is to change your default password to something more secure. Skip over the next two sections, and continue at Step 3 (All Users): Changing your Password.

1.3.2 Mac Users

Step 1: Accessing the Command Line

OS X is built upon a unix base. We are able to open a local shell to our individual computer's operating system using the Terminal application on our local machine.

1. Run the Terminal application using one of the two following methods:
   (a) The Terminal app is located in the `Applications▸Utilities` folder. You may use finder to browse to this location and double-click on the Terminal.app icon (Figure 1.7).
   (b) You may also get quick access to the program via spotlight (Figure 1.8). Press ⌘+Space, and then type `terminal`↵ (Remember, the ↵ character represents the

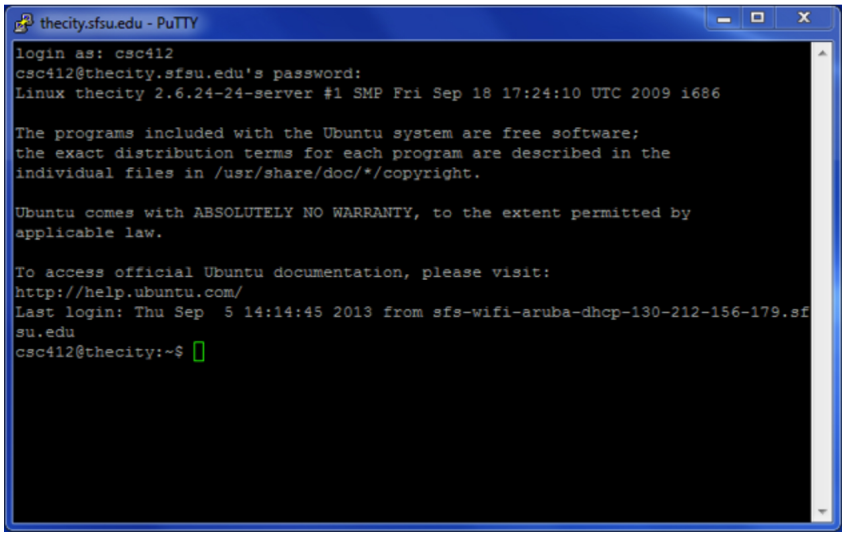


Figure 1.6: PuTTY Terminal window connecting to csc412sfsu.com

Figure 1.7: Terminal application icon.

return, or enter key).

Configuring Terminal Emulation

2. Following the release of OS X 8.5, there are some problems with terminal emulations in Terminal. To make sure that the correct type of terminal is being emulated, from Terminal's menu bar, select Terminal > Preferences (or type ⌘+,), then select the Advanced tab (Figure 1.9). In the "Declare terminal as:" field, select "xterm".

Your Home Directory

3. After using either method a or b from Step 1, you will see a terminal window appear. In the case of Figure 1.10, the terminal window has a blue background, white foreground, and the prompt is " $"; your terminal window may appear somewhat different. Note the prompt in the terminal window in Figure 1.10 is indicating that current directory is the user's home directory (more on this later).

Now that you are at a command prompt on your local computer, you are ready to connect to the remote class server.

1.3.3 OS X Users Step 2: Connecting to the Class Server

Since OS X is UNIX-based, it means that certain useful software is already installed with the operating system, including an SSH client and SCP (secure copy) client. We will be using these clients to connect to the class server.

To connect to the class server:

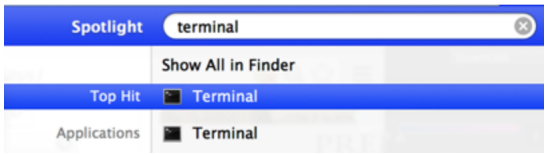


Figure 1.8: Using Spotlight to start the terminal app.

Figure 1.9: Mac terminal emulation setting.

1. At the command prompt type: `ssh username@csc412sfsu.com` ↵, where *username* is your class server username (see Your Username and Password on the Class Server).
2. You will then be prompted for your password. Type your password, and press return. Note: as a security feature you may not see anything as you type your password.

You are now connected to the class server (Figure 1.11). The first thing to do is to change your default password to something more secure. Continue to the next step, Step 3 (All Users): Changing your Password.

1.4 Step 3 (All Users): Changing your Password

Now that you're connected to the remote server, the first thing you must do is change your default password. To do this, you will use the `passwd` command.

1. At the command line, type `passwd` ↵.
2. When prompted for your current password, enter your current password and press ↵.
3. When prompted a new password, enter your new password and press ↵.
4. When prompted to "retype your new UNIX password", enter your new password again and press ↵. If your new password is valid and matches, the system will let you know that your password was updated successfully.
5. Log off the remote server by typing `exit` ↵.

1.4.1 Step 4: UNIX Basics

Now that you can get into your UNIX account, it is time to familiarize yourself with some UNIX concepts and commands.

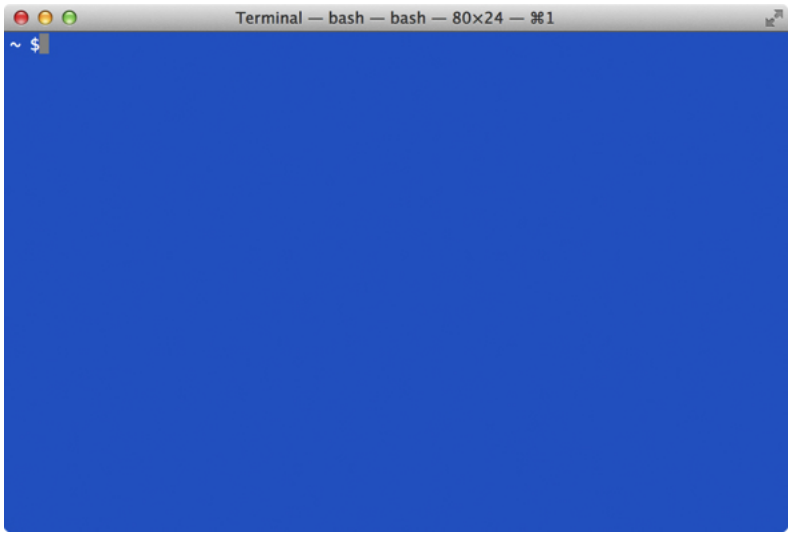


Figure 1.10: Terminal window on a Mac. Your colors may vary.

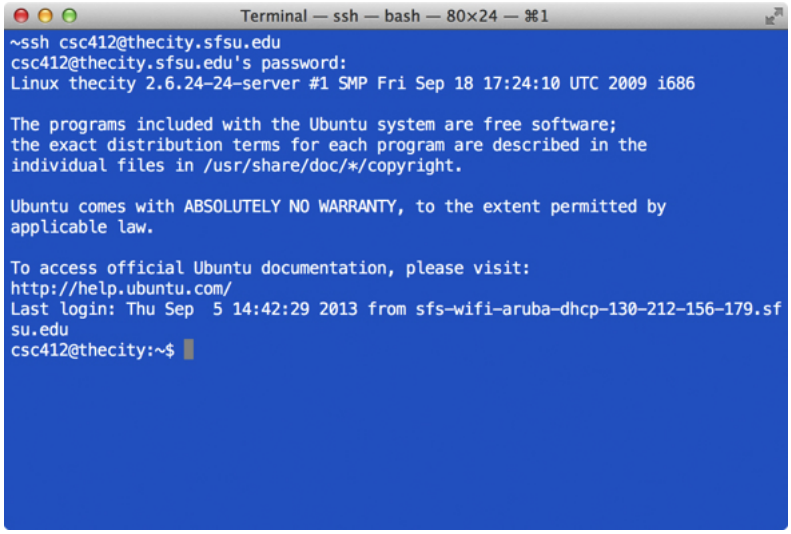


Figure 1.11: Connecting to the class server on OS X.

UNIX File Structure

The UNIX file structure is organized like most modern operating systems into a hierarchical tree-like structure. This tree contains directories, which are roughly equivalent to folders in Windows or OS X. Directories may contain files and sub-directories. Files may be executable, which means they can be run by typing their name. A portion of AWS-Server's directory structure is depicted in Figure 1.12.

The directory at the base of the tree is called the *root directory*. It is denoted by a forward slash (/). A *path* is the list (in order) of the directories you traverse to get to a file.

So, for example, the path to the index.html file at the bottom of the tree in Figure 1.12 is `/home/csc412/public_html/index.html`. When a path begins with a / (i.e. when you specify the path all the way from the root), it is known as an absolute path. Notice that each directory in the path is separated from the next by a forward slash.

When you are at the command line, you are in a *working directory*. It is much like working with files in one particular folder at a time, that folder being the default where actions will take

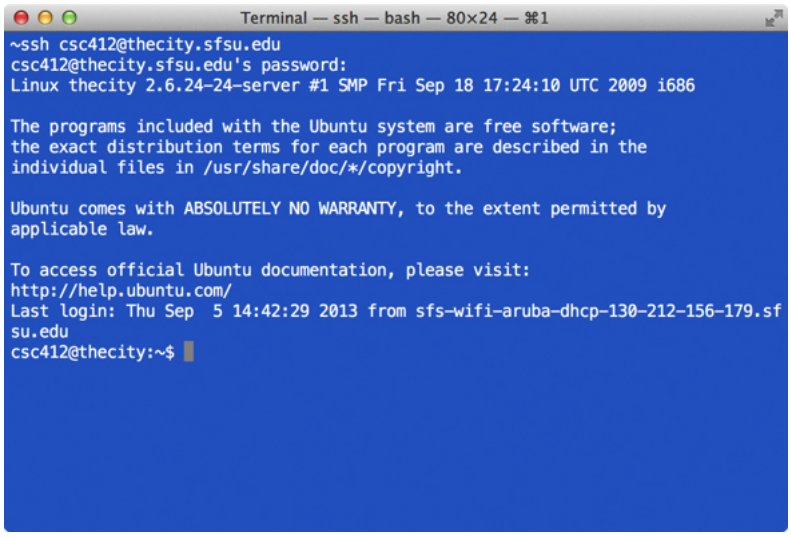


Figure 1.12: Connecting to the class server on OS X.

place. If you do not specify a path when using UNIX commands, it is assumed you are referring to a file in your working directory. You can move between directories, changing your working directory.

You may also specify directories using relative paths. Relative paths do not begin with a /. For example, if the current working directory was `/home/csc412` (Figure 1.12), and we wanted to see what was in the `index.html` file located in the `public_html` directory below it, you could use the `cat` command, followed by the file name. Therefore, you have two ways of specifying this filename. The absolute path to `index.html` (as mentioned above) is `/home/csc412/public_html/index.html`. The relative path from the current working directory of `csc412` is `public_html/index.html`. Therefore, with csc412 as the current or working directory, you could type either:

```
cat /home/csc412/public_html/index.html↵
```

or

```
cat public_html/index.html↵
```

with the same result.

There are two other conventions that you should know when specifying paths: a single period represents your current directory, and two periods represent the parent directory (the directory above the current directory).

When you first log into your account on the class computer, your working directory is your *home directory*. Your home directory is referred to in a path by the tilde symbol ~. This is the directory assigned to you when your account was created on the server. Your home directory is a subdirectory located off of /home. Table 1.1 contains some directories that are commonly located in home directories on UNIX systems. On AWS-Server, you most likely only have the public_html directory. On other servers, you may see the mail and cgi-bin directories. These directories are associated with important system functionality and should not be removed.

The public_html directory

The public_html directory is a special directory that the web server looks in to find web pages to serve. The URL `http://csc412sfsu.com/~username`, refers to the `public_html`

Directory	Description
mail	This directory contains files important to the correct operation of your email account on the server. It is recommended that you do not modify these files.
public_html	This is the root of your website located on the server. Files placed in this directory can be seen by the public using a web browser, so do not place any sensitive information in this directory.
cgi-bin	This is the directory that contains executable scripts to be run by the web server.

Table 1.1: Common UNIX directory name conventions.

directory off of *username*'s home directory, which we will call your account URL. The web server looks first for index.html file, and if it is there and has the correct permissions, serves the contents up to the web. If index.html is not present (or certain other specially named files), an error will appear in the browser when the user's account URL is accessed.

UNIX Commands

Now let's review some essential UNIX commands. Commands are typed at the prompt, with the command the first (or leftmost) thing you type. Commands may take arguments, which are much like arguments to a method. Arguments are passed to commands by typing a space after the command and then the argument; multiple arguments are separated from each other with spaces. Remember, the shell does not process the command line (including the arguments) until the ↵ key is pressed.

There is a (mostly) standardized way to describe UNIX commands, which is shown in the following table:

Behavior	Description
Required	Required arguments are listed without square brackets
Optional	Optional arguments in square brackets; do not include these square brackets when you are using the command in the shell
Repeated	Arguments can be used more than once in a single command is specified with an elispsis (. . .); do not type the three dots, but rather you may repeat the argument, separated by a space.

Table 1.2: UNIX command description syntax.

Each of the following commands is actually quite powerful. Only the most basic usage of each command is described below. It is recommended that you use one of the references at the end of the lab to explore other aspects of these commands.

ls [option]... [file]...

Lists the contents of a directory. With no arguments specified, ls will list the contents of the working directory. To see all files, including hidden files, use the -a option (type `ls -a` ↵ at the command prompt). To see the files in another directory, you can specify which file or directory you wish to see. For example, to see the contents in the 412 directory from my home directory, you can type `ls /home/csc412/` ↵.

cd [directory]

Change your working directory. `cd` ↵ will change your working directory your home directory. `cd ..` ↵ will change your directory to the parent directory (one directory level above). `cd ~` ↵ will change your current directory to your home directory. If the present working directory is csc412 (Figure 1.12), to move to the public_html sub-directory type `cd public_html` ↵. You may use an absolute pathname. From anywhere in the file system you could get to the public_html directory of the previous example by typing `cd /home/csc412/public_html` ↵.

man [manual page]

Displays online help pages about a command. For example, to see information about the ls command, you would type `man ls` ↵

cat [option] [file]...

Prints the contents of the file to the console. Use this command to see what a file contains. For example, to see the contents of Hello.java, you would type `cat Hello.java` ↵.

cp [option].. source dest

Copies *source* file to *dest*. You may change the name when you copy to *dest* by providing a new file name. For example, from the csc412 directory, to copy public_html/index.html to ./index_copy.html you would type: `cp public_html/index.html ~/index_copy.html` ↵. This will make a copy of index.html in your home directory. This command can also be used to copy entire directories.

mv [option[.. source dest

Moves a file from source location to dest location. This command can be used to change the name of a file. To move a file from public_html subdirectory to the current directory you would type `mv public_html/index.html .` ↵ (notice the user of the period to mean the current directory). To change the name of index.html in the current directory to index1.html, type `mv index.html index1.html` ↵.

pwd [option]

Prints the current working directory.

1.5 Lab 1 Classwork

1. Login to your account on the class server.
2. Find out what your current working directory is.
3. Change your current working directory to your account's public_html directory
4. Find out what your current working directory is. How does the path of this directory differ from the path in step 2?
5. Find out what files are in your current working directory (you should still be in your public_html directory)
6. Copy the file hello_index.html, which is located in /home/csc412/ to your current directory.
7. Using a browser, browse to http://csc412sfsu.com/~ username,where username is your

username on the class server.

8. Going back to the terminal and rename hello_index.html to index.html.
9. Go back to the browser, and reload the page from step 7. What has changed? Notice that index.html is a special page name.
10. Go back to the terminal and try using the nano editor on your index.html file. Type: `nano index.html` ↵.
11. Under the line that prints Hello World! to the browser window, add a line that says "This is the page of username" where username is your username.
12. Add something of your own choosing to the webpage, such as a message or a quote.
13. Save the page. From nano you must use ctrl- X and be sure to say yes to saving the file!
14. Let me know that you have finished your assignment by emailing me from the command line, and sending a copy to yourself. To do this, you use the sendmail command. Type: `sendmail csc412sfsu@gmail.com your_mail` ↵ from the command line, where `your_mail` is your email address. Note: After you hit return all you will see is sendmail waiting for input; it will appear that sendmail is hung, but it is actually waiting for you to type something. Now you will compose and email to csc412sfsu@gmail.com and to yourself.
15. The first you should do is give the email a subject. To do this you type: `Subject: whatever` ↵ where `whatever` is your subject line. For this mail. please make your subject "csc412.01 username done!", where username is your username.
16. Once you have hit return after the subject line in step 15, you are now typing in the body of the email message. This one should just contain the url to the page that you created. For example, "http://csc412sfsu.com/~csc412".
17. Email messages may contain more than one line, sendmail will not quit if you just hit return, or type quit. To finish the email you are currently working on and send it, type a period on a line by itself and hit return.

1.6 References

1. An interesting general-purpose introduction to web servers.
2. Sever Definition
3. Secure Shell (SSH)
4. How to spell "unix"
5. UNIX Introduction
6. UNIX Commands
7. Bash Shell
8. Pico, nano
9. Bash Shell