

In this assignment you will get familiar with tools and the environment we will use in this course.

Estimate time of completion: 2 hours.

## 1. SSH

Secure Shell (SSH) is a cryptographic network protocol for initiating text-based shell sessions on remote machines in a secure way. This allows a user to run commands on a machine's command prompt without them being physically present near the machine. It also allows a user to establish a secure channel over an insecure network in a client-server architecture, connecting an SSH client application with an SSH server. You can find many tools that support SSH. On Windows, please refer to <http://www.ecs.csun.edu/isgroup/templates/pdfs/Secure%20Shell%20SSH.pdf> On Mac,

Open your Terminal application. You will see a window with a \$ symbol and a blinking cursor. This is your basic command prompt. From here, you may issue the command to establish the SSH connection to your server. The most basic usage of this is as follows.

```
ssh your_CSUN_ID@k200.ecs.csun.edu
```

Immediately after issuing this command, your computer will attempt to establish a connection to your server. If it is your first time connecting to that server, you will see a message asking you to confirm the identity of the machine:

Make sure you entered the address properly, and type yes, followed by the return key, to proceed.

You will then be prompted to enter your password.

## 2. Unix (Linux) Command

After you log in to the CECS K200 Server there is an interactive program that provides an interface between you as a user and the kernel of the server system. The shell interprets commands entered by the user or supplied by a shell script, and passes them to the kernel for execution. Shells available for use on Unix and Unix-like systems include sh (the Bourne shell), bash (the Bourne-again shell), csh (the C shell), tcsh (the TENEX C shell), ksh (the Korn shell), and zsh (the Z shell).

There are some commands you need for basic interaction with the server. Confirm your command input with a return key.

### a. Files

ls --- lists your files

ls -l --- lists your files in 'long format', which contains lots of useful information, e.g. the exact size of the file, who owns the file and who has the right to look at it, and when it was last modified.

ls -a --- lists all files, including the ones whose filenames begin in a dot, which you do not always want to see.

There are many more options, for example to list files by size, by date, recursively etc.

more filename --- shows the first part of a file, just as much as will fit on one screen. Just hit the space bar to see more or q to quit. You can use /pattern to search for a pattern.

emacs filename --- is an editor that lets you create and edit a file. See the emacs page.

mv filename1 filename2 --- moves a file (i.e. gives it a different name, or moves it into a different directory (see below)

cp filename1 filename2 --- copies a file

rm filename --- removes a file. It is wise to use the option rm -i, which will ask you for confirmation before actually deleting anything. You can make this your default by making an alias in your .cshrc file.

#### b. Directories

mkdir dirname --- make a new directory

cd dirname --- change directory. You basically 'go' to another directory, and you will see the files in that directory when you do 'ls'. You always start out in your 'home directory', and you can get back there by typing 'cd' without arguments. 'cd ..' will get you one level up from your current position. You don't have to walk along step by step - you can make big leaps or avoid walking around by specifying pathnames. pwd --- tells you where you currently are.

Please refer to the following Wiki page for more command.

[https://en.wikipedia.org/wiki/List\\_of\\_Unix\\_commands](https://en.wikipedia.org/wiki/List_of_Unix_commands)

### 3. File Transfer

You can exchange files between your local computer and the K200 Server.

Command Line:

Please refer to this document for the detail.

<http://www.ecs.csun.edu/isgroup/templates/pdfs/Secure%20File%20Transfer%20Client%20SCP.pdf>

The command you need to initiate a file transfer from Terminal to K200 server is sftp [your\\_CSUN\\_ID@k200.ecs.csun.edu](mailto:your_CSUN_ID@k200.ecs.csun.edu)

Make sure you exit from previous SSH session before launching a new session.

After the sftp session is established you can use put command to upload the file to your working directory at K200 server.

put YourFile.c

Via ZDrive:

In lab, you can use the Connect To Server feature of the Finder to access your ECS ZDrive. Select Go > Connect To, and enter

Smb://zdrive.ecs.csun.edu/csun\_username

At server address bar. After typing your user name and password you will see your ZDrive in a new finder window. You can use normal GUI operations to exchange files between the K200 server and your local computer.

Other Tools:

If you use other tools, e.g. FileZilla the server address needs to include SSH by using, Sftp://k200.ecs.csun.edu

#### 4. Install Progress Tracking Tool

We use WakaTime to help you track your programming goals and monitor your progress towards them. A tracking log is usually required with your lab reports. In this lab, you need to install the open-source plugins to see personalized coding metrics on your dashboard following this instruction,

<https://wakatime.com/help/plugins/sublime-text>

If you use a different editor following the similar steps to install WakaTime.

#### 5. Compile and Execute C Program

Option 1: Use GCC

The standard way to compile a C program is with the command

```
gcc hello.c -o hello
```

This command compiles hello.C into an executable program named "hello"

Option 2: Use make

make is a Unix tool to simplify building program executables from many modules. make reads in rules (specified as a list of target entries) from a user created Makefile. make will only re-build things that need to be re-built (object or executables that depend on files that have been modified since the last time the objects or executables were built).

COMP 322 Lab does not require you to learn how to write a make script. After editing your program file(s), rebuild the executable by typing make in the shell.

More details about make can be found at,

<http://www.gnu.org/software/make/manual/make.html>

Once your source code is compiled you should see a file named myprog in the same folder. Use this command to run the executable.

```
./myprog
```

**Submission Requirement:**

Please take these screenshots and include all of them in one PDF file.

1. After you log into the K200 Server. (5 pts)
2. After you compile and execute the c program named myprog.c. (10 pts)
3. WakaTime dashboard to show your activity logs of finishing this lab assignment.

The PDF file comprising your screenshots needs to be named following this format  
YourCSUNID\_YourLastName.pdf  
and submitted here in Moodle.

Submission failed to meet the submission requirement will not be graded. Grade may be forfeited.