

CSE-278: Introduction to Systems Programming

Lab Exercise #1:

You should save/rename this document using the naming convention **MUId.docx (example: guerinf.docx).**

Objective: The objective of this exercise is to review the process of accessing a Linux server and working on a Linux-based computer. Review some basic Linux commands.

Fill in answers to all of the questions. For almost all the questions you can simply copy-paste appropriate text from the Terminal window into this document. You should refer to [ShellCommands.pdf](#) document available on the server. To download it, see last exercise. You may discuss the questions with your neighbor or instructor. However, the key part of the exercise is to try the different commands and explore them via trial-and-error process.

Name: **Noah Dunn**

This course will use the following Linux server throughout this course. You should memorize the name of this server: **database.csi.miamioh.edu**. Alternatively, we are going to use the following **ceclnx01.cec.miamioh.edu**.



.Wait for your instructor to guide the class through starting a terminal and running some of the basic commands in a Linux terminal. The process is relatively straightforward. If you already have some experience with working in a terminal, you may proceed with this exercise. A video demonstration of using a Terminal to access the Linux server is available off [Canvas → Pages](#).

Step 0: Log onto the Linux server

1. Open a Mac terminal (use MacKey Spacebar to start spotlight and search for terminal)
2. At the terminal prompt (\$) type the following command to log onto the Linux sever (you need to use your MUId in the command below):

```
$ ssh MUId@ceclnx01.cec.miamiOH.edu
```

3. Type in your password when prompted and press Enter. Note that when you type your password you will not see it or see any special characters

1. When you log onto the Linux machine, you will start off in a default directory called your **home** directory. You should create all your files and save your work off sub-directories under your home directory. To figure out what your home directory is, you need to use the `pwd` (present working directory) command (that is, type `pwd` at the shell (\$) prompt and press `ENTER` key, which is indicated by ↵) as shown below:

```
$ pwd ↵
```

What is your home directory:

```
/home/dunnnm2
```

Note: The home directory will be similar to `/home/guerinf`. It is also referred to as path, which indicates the series of directories to be navigated to reach your home directory. The home directory changes for every user. You must always remember your home directory, as it is handy. In addition, note the following important terminology associated with paths:

Absolute path: In Linux, paths always start with a `/` (forward slash or just slash, i.e., the division sign) indicating the root directory. Example: `/home/raodm` or `/usr/bin/ls` etc.

Relative path: Paths that **do not start** with a `/` are relative paths. Relative paths indicate directory and file structures with respect to `pwd` (present working directory). Example: `../cse278` or `../` or `../../courses/csex43/exercises` or `cse278/exercises` etc.

2. Now practice creating a directory and navigating directories using relative and absolute paths do:

- i. **mkdir (make directory):**

This command is used to create a directory that is specified either as a relative or absolute path. Example:

```
$ mkdir cse278
```

- ii. **cd (change directory):**

This is a very frequently used command that is used to change your present working directory (`pwd`) and for navigating directory structures. The new directory to change to can be specified as a relative or absolute path

```
$ cd cse278
$ pwd
```

3. Change your `pwd` to the `cse278` directory you created earlier. What is the output you get when the following sequence of commands are typed and executed at the shell (\$) prompt?

```
$ cd ~/
$ pwd
```

Output: `/home/dunnnm2`

- i. In our own words, explain what you think the above commands are achieving?

The "`cd ~`" command changes present working directory to the user's home directory.

4. In Linux, the `ls` (list) command is used to list files in a directory and `ls` has lots of features/options. Try the following 2 versions of `ls` (the `-l` "dash ell" option lists details of files)

```
$ ls
$ ls -l
```

The "`-l`" option enables long listing which provides more details on each of the files in the current working directory. Later on we will spend more time on outputs from the `ls` command.

The `ls` command has many options. To look up information on `ls` (or other Linux commands) you can look-up information from the `man(ual)` page for the `ls` command (type `man ls` at the shell prompt, use arrow keys to navigate, and **press q to quit reading** the `man` page) . Needless to add, you can always Google for the commands and find out details.

5. Determine some basic information about the operating system by typing the command “`uname -rs`” and noting the name of the operating system (first word) and version of the kernel (second word).

OS Name: **Linux**

Kernel Version: **4.4.0-112-generic**

You might want to do the same with the other server. Open a window and login (ssh) into **ceclnx01.cse** , repeat the commands in 3,4,5 and logout (with *exit*).

6. Now let’s practice how to write output from a command (or could be output from your C++ program later on) . For now create a simple text file using the following command:

```
$ ls -l /usr/share > dump.txt
```

7. Now let’s practice how to copy files using the `scp` (secure copy) Linux command.
- Copy `dump.txt` from the Linux server to your local machine (overwriting any file with the same name if it exists in the destination directory) using the command below:

Start a new Terminal window on your Mac. The Mac also runs a shell similar to Linux. At the Mac shell prompt use the `scp` command.



Since Mac and Linux are pretty close it is important to ensure you are able to distinguish the terminal between Linux and Mac. For this you could change the color (or profile) of a terminal.

The syntax for `scp` is: `scp <SourceFile> <DestinationFile>`

```
$ scp Muid@database.csi.miamiOH.edu:dump.txt .
```

Using the `ls` command ensure the file has been copied to your Mac correctly. If you get errors such as “No such file or directory”, then check to ensure your source file path is correct (using `ls` and `pwd` commands on your Linux machine)



By switching the order of parameters you can copy files from a local machine to a Linux machine. Example (copy `hello.txt` to Linux machine):
`scp hello.txt raodm@ceclnx01.cec.miamiOH.edu:~/cse278/exercise1`



You must memorize and practice the commands covered in this laboratory exercise: `pwd` (present working directory), `cd` (change directory), `mkdir` (make directory), `ls` (list), `cp` (copy), `mv` (move/rename), `scp` (remote secure copy) `less` (view a file).

8. **(Optional. It may not be available for the first lab).** Respond to the online Canvas discussion titled “[The most weird bug I have encountered in my program](#)” by completing the following tasks:
 - i. Describe a bug (aka semantic error) that you have encountered in one of your programs that you found hard to troubleshoot or the bug was really weird (by responding to this initial posting).
 - ii. Respond / Reply to at least one (you can respond to as many as you would like) posting from another individual.
9. Once you successfully completed the aforementioned exercises, save this MS-Word document as a PDF file. Next upload the following to Canvas:
 - i. This MS-Word document (duly filled-in) saved as a PDF document.

Ensure you actually **submit** the files after uploading them to Canvas.