**DUE DATE:** **Thursday, May 28, 2020 Before 11:59 PM (Midnight)**

# CSE-278: Introduction to Systems Programming

# Homework #1

Max Points: 50

**Note: If you are using your personal machine then prior to commencing work on this exercise, you may need to enable Windows `OpenSSH` client (see: https://websiteforstudents.com/enable-windows-10-built-in-ssh-client-server/)**

**NB: ssh command is enabled by default in Windows 10**

**You should save/rename this document using the naming convention MUid.docx (example: ahmede`.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 off Canvas under `Files→Handouts`. You may discuss the questions with your TA/instructor. However, the key part of the exercise is to try the different commands and explore them via trial-and-error process.

**Name:** **William Mechler**

**PART 1**

We will use only textual/terminal interface to Linux
– On Mac use built in Terminal & SSH
– For Windows install PuTTY & PSCP from:
https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html

Linux server for this course:

- Host/server name: **os1.csi.miamioh.edu**
- You will need VPN for off-campus access to the server
- https://miamioh.teamdynamix.com/TDClient/Requests/ServiceDet?ID=9101

This course will use the following Linux server throughout this course. You should memorize the name of this server: **`os1.csi.miamiOH.edu`**. For off-campus access, you will need to use Virtual Private Network (VPN) due to security concerns. You can get VPN software for free from: http://miamioh.edu/vpn

1. What is the server-name or hostname of the Linux server being used for this course?

   The hostname of the Linux server is: `os1.csi.miamiOH.edu`

2. Assume you would like to access the server when you are not on-campus. What extra software do you first need to run before trying to access the server while off-campus?

   You would need to install Any Connect vpn to connect to the server when not on-campus.

3. What was the full `ssh` command you used (from Windows-powershell or a Mac-Terminal) to log onto the Linux server:

```
ssh mechlewg@os1.csi.miamiOH.edu
```

4. 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 in 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/mechlewg`

   Note: The home directory will be similar to `/home/ahmede`. 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/ahmede` 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.

5. Now let's practice creating a directory (aka a "folder") and navigating directories using relative paths via:

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

   Use this command to create a directory named `cse278` that is specified as a relative path (you may also use an absolute path with this command). 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. Here we will use a relative path:

   ```
   $ cd cse278
   $ pwd
   ```

   **Note**: The `pwd` command prints the absolute path to the Present Working Directory (pwd). You can use `pwd` to remind you of your working directory. It is handy for troubleshooting issues, particularly with "File not found" type errors.

6. 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/mechlewg

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

   cd ~/ takes you back to your home directory. And pwd shows what directory you are currently in.

7. 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 Google for `man ls` .

8. 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 (typically the word with numerical values).

OS Name: Linux    Kernel Version: 4.15.0-65-generic

9. Now let's practice how to write output from a Linux 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 > ~/cse278/dump.txt
```

10. Now let's practice how to copy files to/from the Linux server using the `scp` (secure copy) Linux command. Assume we want to copy the `dump.txt` file from `~/cse278` directory to the `Downloads` folder on your local computer.

i. Start a new/different powershell window (or a `Terminal` window on your Mac). **Note**: You should be running `scp` on your local computer!

ii. From the powershell prompt run the following two commands (change `MUid` to your ID). Don't forget the `./` at the end of `scp` command to signify the current folder.

```
cd ~/Downloads

scp MUid@os1.csi.miamioh.edu:~/cse278/dump.txt ./
```

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)

> i 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 MUid@os.csi:~/cse278/exercise1`

## `scp` Practice

The scp command is a frequently used command. You will use it a lot in this course. In the past students have struggled to memorize the command and you will most likely struggle with `scp` if you don't form a strong mental model. So let's get some practice with `scp`.

**Note**: Just type the `scp` command in this MS-Word document for practice. Do not try to run the `scp` commands as it will generate errors (because the files you will be using do not actually exist).

11. Write the `scp` commands for each one of the following scenarios. The first one is completed for you to illustrate an example. For all questions assume server is `os1.csi.miamioh.edu`.

   i. When you download files from Canvas they are typically placed in your `Downloads` folder. Copy a file named `Exercise1.java` from the `Downloads` folder to a directory named `cse278/exercise1` on the server.

```
cd ~/Downloads
scp Exercise1.java ahmede@os1.csi.miamioh.edu:cse278/exercise1
```

   ii. Copy a file named `dump.txt` from the `Desktop` folder to a directory named `cse278/ex1/tests` on the server.

```
cd ~/Desktop
scp dump.txt mechlewg@os1.csi.miamioh.edu:cse278/ex1/tests./
```

   iii. Copy a file named `ex1.cpp` from the `Downloads` folder to a directory named `NetBeansProjects/ex1` on the server.

```
cd ~/Downloads
scp ex1.cpp mechlewg@os1.csi.miamioh.edu:cse278/NetBeansProjects/ex1./
```

12. Write the `scp` commands for each one of the following scenarios. The first one is completed for you to illustrate an example. For all questions assume server is `os1.csi.miamioh.edu`.

   i. Copy a file named `ex1.cpp` in a folder named `NetBeansProjects/ex1` from the server to your `Downloads` folder.

```
cd ~/Downloads
scp mechlewg@os1.csi.miamioh.edu:NetBeansProjects/ex1/ex1.cpp ./
```

   ii. Copy a file named `out.txt` in a directory named `NetBeansProjects/hw1` on the server to the `Desktop` on your local machine

```
cd ~/hw1
scp out.txt C:/Users/willm/Desktop
```

iii. Copy a file named `ex1.html` in a directory named `cse278/testing` on the server to your `Downloads` folder

```
 cd ~/testing
scp ex1.html C:/Users/willm/Downloads
```

13. Let's do an exercise of copying a Java program to the Linux server (helps to double check if you are learning anything about `scp`) and running a Java program on the Linux sever (you will perform similar process for running C++ programs as well) –

    i. Download solution for `Exercise1.java` and input text file `para1.txt` `para2.txt` from Canvas and `scp` it to the Linux server to your home directory (*i.e.*, `~/`). Remember `scp` is always run (in a powershell/terminal) on your local computer (and not on the server).

    More specifically, the java code should be able to handle:

    A. Input redirection ( < ) in the Linux shell prompt (without any modification of the supplied starter code)

    B. Command Line Parameter as args[0] (modify the code a little bit to do this)

    As a test for the both approaches, use para1.txt and para2.txt as the suitable input filename.

    ii. Now, on the Linux server terminal, ensure your `pwd` is your home directory (use `cd` command to change directory if needed) and ensure `Exercise1.java` is there (of course, using the `ls` command).

    iii. Compile the Java program using the Java compiler

    ```
    $ javac Exercise1.java
    ```

    iv. Run the java program using the JVM (without any modification of the starter code)

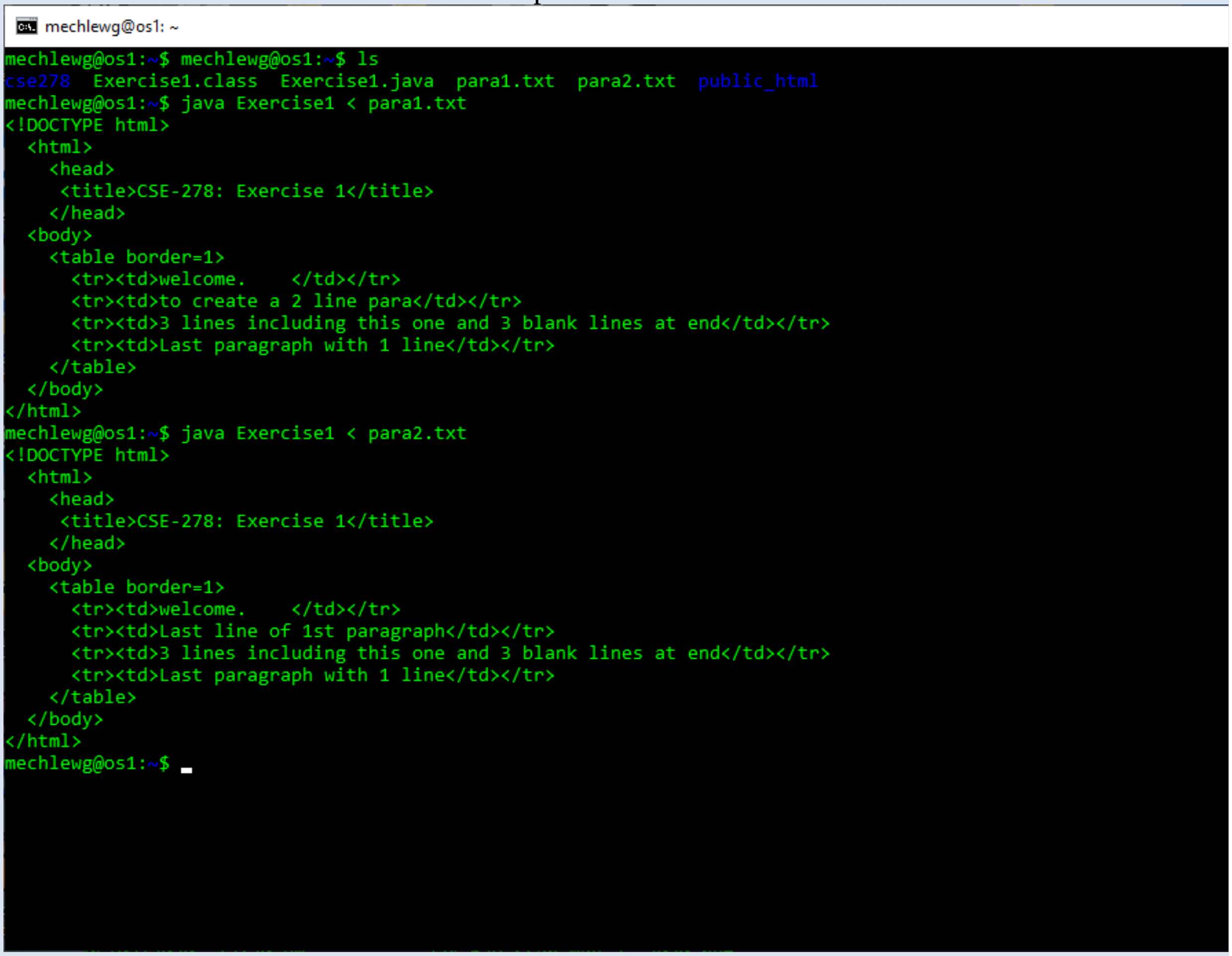
    ```
    $ java Exercise1 < para1.txt
    ```

    ```
    $ java Exercise1 < para2.txt
    ```

v. Copy-paste the `para1.txt` input to the program and observe its operations. In powershell click on the icon (top-left corner) to get menu and select `Edit`→`Paste`. You will need to press Control+D to stop the program.

vi. Copy-paste a screenshot of the powershell/terminal (showing output from the program) in the space below:

Place a screenshot of powershell/terminal window here



```
mechlewg@os1:~$ mechlewg@os1:~$ ls
cse278  Exercise1.class  Exercise1.java  para1.txt  para2.txt  public_html
mechlewg@os1:~$ java Exercise1 < para1.txt
<!DOCTYPE html>
  <html>
    <head>
     <title>CSE-278: Exercise 1</title>
    </head>
  <body>
    <table border=1>
      <tr><td>welcome.    </td></tr>
      <tr><td>to create a 2 line para</td></tr>
      <tr><td>3 lines including this one and 3 blank lines at end</td></tr>
      <tr><td>Last paragraph with 1 line</td></tr>
    </table>
  </body>
</html>
mechlewg@os1:~$ java Exercise1 < para2.txt
<!DOCTYPE html>
  <html>
    <head>
     <title>CSE-278: Exercise 1</title>
    </head>
  <body>
    <table border=1>
      <tr><td>welcome.    </td></tr>
      <tr><td>Last line of 1st paragraph</td></tr>
      <tr><td>3 lines including this one and 3 blank lines at end</td></tr>
      <tr><td>Last paragraph with 1 line</td></tr>
    </table>
  </body>
</html>
mechlewg@os1:~$ _
```

vii. Now modify the java code so that the above can be run as:

```
$ java Exercise1Modified para1.txt
$ java Exercise1Modified para2.txt
```

viii. Copy-paste a screenshot of the powershell/terminal (showing output from the modified program) in the space below:

Place a screenshot of powershell/terminal window here

# Additional Linux command to practice:

14. Now back to the Linux server. On the Linux machine copy `dump.txt` to your home directory (indicated by `~/`) but with a different name using the command below:

```
$ cp dump.txt ~/dump1.txt
```

Using the `ls` command ensure the file has been copied to your local home directory correctly.

> Note the difference between `cp` and `scp`:
> 1. `cp` is local copy – copy file from one directory to another on the same machine.
> 2. `scp` is for remote copy – copy a file from one machine to another!

15. Now, on the Linux machine, practice moving a file from one directory to another. You can also use mv to change the name of a file. Unlike copy (`cp`), move (`mv`) command removes the original file.

```
$ mv ~/dump1.txt ~/dump2.txt
```

Using the `ls` command ensure the file has been moved/renamed correctly.

16. Now let's practice how to delete or remove files using the `rm` (remove) Linux command. **Note**: Deleting files cannot be undone ☠ you delete a file it is gone for good ☠ consider yourself duly notified of this consequence ☠

i. Remove dump.txt from your home directory using the following command:

```
$ rm ~/dump2.txt
```

Using the `ls` command ensure the file has been removed from your home directory correctly.

> 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).

17. 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.

# PART2: UNIX AND LINUX COMMANDS (FILE AND DIRECTORY MANIPULATION)

## OBJECTIVES:

The objectives of this part of the laboratory exercise are to:

- Explore *script* command for recording user activity
- Review *file* manipulation
- Gain familiarity with *most frequently used* Linux/UNIX commands
- Explore BASH (Bourne Again Shell) environment

## PRELIMINARIES

Perform and record the following Linux operations, *in the given order*, in a log file using Linux "*script*" command.

### STEP 1: EXPERIMENT WITH EACH OF THE COMMANDS IN THE GIVEN ORDER

1. Start recording
2. Display the current date and time
3. Display who is logged in
4. Create **hw1** directory in your root directory
5. Change the access rights of this directory as follows:
    a. User: read-write-execute
    b. Group: read-write

   c. Others: read-execute

6. Direct the *detailed listing* of your root directory to the file **root.txt**
7. Copy **root.txt** file to the current directory with the name **root2.txt**
8. Sort the **root2.txt** from the file size in *descending order* and store the result in **sorted.txt**
9. Concatenate files **root.txt** and **root2.txt** into the file **joint.txt**
10. Display the content of the file **root2.txt**
11. Rename **root2.txt** as **newroot.txt**
12. Copy **newroot.txt** file to the **hw1** directory with the name **root3.txt**
13. Change current directory to **hw1** directory
14. Create the **hold** subdirectory in **hw1**
15. Create the **temp** subdirectory in **hw1**
16. Copy **root3.txt** into **temp** directory with the name **root4.txt**
17. Move **root3.txt** into hold directory with name **root5.txt**
18. Delete **root5.txt** while you are in **hw1** directory
19. Delete **hold** directory
20. List the detailed contents of **hw1** directory
21. Clear the screen
22. Go back to your root directory
23. Display the current working directory
24. Stop recording

## STEP 2: ONCE YOU ARE SURE THAT ALL THE COMMANDS ARE OK

**Directions**

- No late assignments will be accepted!
- This work is to be done *individually*
- The script file will have to be saved with the name ***hw1_yourMUID.log***
- Make sure that the log file content is no more than 1-2 pages (by issuing the command `wc -l` and check the number of lines, typically 50 lines are there in a page)
- Please make sure that the log file is pure text, there should NOT be any weird characters
- In case the output of a command will be very lengthy, you may use output redirection > to temp text file so that the log file content will be less
- Assignment is due before Thursday, May 28 11:59pm Midnight
- On or before the due time, drop the *electronic copy* of your work in the *canvas*

Don’t forget to Turn in the two files!  hw1_MUID.pdf & hw1_MUID.log