

The University of Alabama in Huntsville
Electrical and Computer Engineering
Project 1 (10 points)

This assignment is to be handed into your TA by the end of the lab period on:

Mon/Wed Labs: Wednesday January 16, 2013

Tue/Thur Labs: Thursday January 17, 2013

In this first assignment, you will become more familiar with the UNIX operating system and how to perform several basic actions in the terminal windows. **The most powerful and useful way to use UNIX is by typing commands at a prompt in a terminal window.**

Login to the Linux server (falcon, blackhawk or eagle)

Perform the following steps to log onto Blackhawk (A number ends in an even number) or eagle (A number ends in an odd number)

Click on **Options**, drag the mouse pointer over **Remote Login** and select **Enter Host Name**
Type in **falcon, blackhawk or eagle** in the dialog box and press return. After a few seconds, the CentOS login screen appears
Enter your username – press return
Enter your password – press return.

You are now logged into falcon, blackhawk or eagle.

Opening a terminal window

Right click on an empty space on the screen and select terminal.

Laboratory Project #1 Description

This first assignment will go over several of the commands listed in the UNIX guide provided by the TA's. As you go through the procedures in this handout, fill in the answers on the answer sheet where indicated. Once you have finished the worksheet, **turn it in to your TA by the deadline at the top of this page.**

All work for this assignment will be performed in terminal windows. The following pages of this assignment provide all the commands necessary to perform the work involved with this project.

Remember that using the “tab complete” feature can complete all filenames. For example, if you have a file named: this_is_a_very_long_filename.txt and the name is unique after typing in “this”, then to obtain the entire filename all you type in is “this” and press the tab key. An example of this feature will be demonstrated in the assignment.

NOTE: When you remove/delete a file on a UNIX machine, the file is gone and cannot be retrieved. There is no recycle bin or trashcan that holds a deleted file until it is permanently removed. So, if you are going to delete a file, make sure that you absolutely want it deleted.

→→ NOTE: If you stop part way through the assignment, type pwd to find out what directory you need to return to before you pick up where you quit. For example if you stop after completing step 13, pwd gives the result of: **/home/student/username/Example/Prog1**. To start at step 14, you open a terminal window and type **cd Example/Prog1** to make the current working directory the same as when you stopped.

Project #1 Part 1 – Using UNIX Commands

Write your answers to the various questions on this document.

When you complete the project, this document is turned in to your TA.

➔ **NOTE:** If you stop part way through the assignment, type `pwd` to find out what directory you need to return to before you pick up where you quit. For example if you stop after completing step 13, `pwd` gives the result of: `/home/student/username/Example/Prog1`. To start at step 14, you open a terminal window and type `cd Example/Prog1` to make the current working directory the same as when you stopped.

To begin this assignment, you should **log into falcon, blackhawk or eagle** and open a terminal window.

★ *In this handout, all UNIX commands typed at a prompt are shown enclosed in angle brackets – like the following: `<command>`. **The angle brackets are not part of the command.** All sample output results printed to the screen are shown in double quotes like: **`"command result output"`**.*

- ⇒ 1) Make the terminal window opened active by moving the mouse pointer into it and press the left mouse button. Now type the command `<pwd>`. **Write down the response shown on the screen.** It should look like `"/home/student/..."`. The command `pwd` shows you the current directory in which the terminal is active. In this case the directory you are in is your home directory. You can return to your home directory from any other directory by typing the command `<cd>`.
- 2) The path shown in step 1 is called your home directory. Every time you login to the Suns, you start in this directory. You can see what files/directories exist in this directory by typing the command `<ls>` (the first letter is a lower case L). If a directory named `Example` exists, remove it and all of its contents by typing the following command `<rm -r Example>`.
- 3) Create a new directory in your home directory. The name of this new directory is to be called **Example**. To do this, type in the following command: `<mkdir Example>`. Now if you type in the command `<ls>` you should see the directory **Example** listed.
- ⇒ 4) All work for the rest of this handout will be performed in the `Example` directory. To enter into the `Example` directory, type the command `<cd Example>`. Enter in the command `<pwd>` and **write down the result shown on the screen.**
- ⇒ 5) Now type the command `<ls>`. There should be no output to the screen since the newly created directory has no files. However, there are always two slightly hidden files in every directory. Type in the command `<ls -a>`. **Write down the result shown for this command.** Files and directories can start with a period (`.`), and just giving the `ls` command does not show these files.
- ⇒ 6) Now create two new directories named `Prog1` and `Prog2`. Type in the command `<mkdir Prog1 Prog2>` followed by the command `<ls>`. **Write down the result shown.**
- ⇒ 7) There is a way to easily create empty files. Type in the command `<touch file1.txt file2.txt>` followed by the command `<ls>`. **Write down the result shown.**
- ⇒ 8) Now type in the command `<ls -l>`. The `-l` flag (this flag is a lower case L) formats the output in a long format. **Write down the results shown for this command.** You do not need to understand all of the fields shown. If a filename is a directory, then the very first letter on the output line is a `"d"`, if the filename is just a file, then the first character is a `"-"`. The last field shown is the name of the file or directory, and the field before that is the last modification time of the file or directory.

- ⇒ 9) Now let's see how to move and rename files. These actions are performed by using the `mv` command. First, let's see how to rename a file. Type in the command `<mv file1.txt file3.txt>`, then type in the command `<ls file*>` and **write the results shown**. The `"*"` used with the filename acts as a wild card. Therefore, all entries (files and/or directories) in the directory that start with "file" are listed.
- ⇒ 10) Next, move the file `file2.txt` to the directory `Prog2` by typing the command `<mv file2.txt Prog2>`. Now move `file3.txt` to the directory `Prog1` and rename it to `file1.txt`. Both the move and the renaming can be done in one step by typing in the following command: `<mv file3.txt Prog1/file1.txt>`. Type the command `<ls>` and **write down the results of that command**.
- ⇒ 11) What happened to the files `file3.txt` and `file2.txt`? Those two files have been moved to the directories `Prog1` and `Prog2`. Type in the command `<ls Prog1>` and **write down the result**. Note that the file listed is `file1.txt`. Not only did we move a file from one directory to another directory, but we also renamed it in the process (from `file3.txt` to `file1.txt`).
- ⇒ 12) Type in the command `<ls Prog2>` and **write down the result shown**.
- ⇒ 13) Move into the `Prog1` directory by typing the command `<cd Prog1>`. Type in the command `<pwd>` and write out the result shown. Now make a copy of `file1.txt` by typing the command `<cp file1.txt file3.txt>`. Type in the command `<ls>` and **write out the result shown**.
- ⇒ 14) So far, we have seen how to move files to sub-directories of a current directory. What if you want to move a file up to a parent directory? Type the command `<pwd>` again. The path shown lets you know where you are in your directory tree. For the path shown, `Prog1` is a sub-directory of `Example` and `Example` is a sub-directory of your home directory. Furthermore, `Example` is the parent directory of `Prog1`. Type in the command `<ls ../>` and **write down the result shown**. You should see a listing of all files and directories in the parent directory (which is `Example`) of the current directory (which is `Prog1`).
- ⇒ 15) Now type in the command `<mv file3.txt ../file4.txt>`. Then type in the command `<ls ../>` and **write down the result shown**. This command moves `file3.txt` from the current directory to the parent directory and it changes the file name to `file4.txt`.
- 16) To see how the "tab complete" feature works, type in "ls f" followed by pressing the tab key. You should see that the name is completed. Since there is only one file in this directory, you can type "ls " and then press the tab key and the file name is placed on the command line.
- ⇒ 17) Next, let's look at the difference between the commands `rmdir` and `rm`. Type in the command `<rmdir file1.txt>` and **write down the result shown** – you should see an error message. The command `rmdir` is used to remove/delete directories only. To delete files, use the `rm` command, so delete the file `file1.txt` by using the command `<rm file1.txt>`.
- ⇒ 18) Move up to the parent directory of the current directory by typing `<cd ../>`. Type in the command `<pwd>` to verify that you are in the `Example` directory. Type in the command `<ls>` to view the contents of this directory and **write down the result shown**.
- ⇒ 19) Type in the command `<rmdir Prog1>` followed by the command `<rmdir Prog2>`. **Write down the results shown** to the screen after these two commands are given. You should see a warning message about directory `Prog2`.
- 20) If a directory is not empty, it cannot be deleted. All files in a directory must first be deleted before the directory itself can be deleted. Therefore, change into the `Prog2` directory by typing the command `<cd Prog2>` and delete the file present `<rm file2.txt>`.

⇒ 21) Change directories up to the parent directory by typing `<cd ..>` and type in the command `<rm no_file.txt>` and **write down the result shown**. There is no file called “no_file.txt”, so you should see an error message.

22) Now remove the Prog2 directory: `<rmdir Prog2>` and remove the remaining file from the Example directory: `<rm file4.txt>`

23) From the ANGEL page for project 1, download the two C++ programs listed (project1.cpp and project1b.cpp). **Be sure to save these programs in the Example directory**. If you need help with this task ask your TA to show you. **Note: the first time you go to save a file an error message about the download directory will come up. Ignore it and click on OK.** In the download box, click on the browse for other folders button. A list of files and directories should come up on the left side. Double click on your home directory (your username); double click on the Example directory and then click on save. The next time you save a file, the download box will come up in the Example directory, so just click on save.

24) Compile program project1.cpp by typing the command `<g++ project1.cpp -o project1>`. The command here is g++ and it invokes the C++ compiler on eagle. Project1.cpp is the source code of the program being compiled. The flag `-o` indicates that the next name is to be the name of the executable file created by the compiler. **Note the order in which the command is written. That order is important!**

⇒ 25) You can now run the program just compiled by typing the command `<./project1>`. **Write down the results shown** for the complete execution of the program. (Use values of 1, 10.75 and 40 for the values asked for by the program). The prompts for data input are already provided below showing the values entered. **You need to write out the four lines that follow the prompts.**

Enter the employee number: 1
Enter pay rate: 10.75
Enter hours worked: 40

⇒ 26) Now compile the program project1b.cpp by typing `<g++ project1b.cpp -o project1b>`. **Write down the results shown** – there should be an error message.

27) Open the file project1b.cpp in a text editor. For this handout, gedit will be used, so issue the command `<gedit project1b.cpp &>`. In step 26, an error should have been reported for line 23. **The actual mistake that caused this syntax error occurred on line 19.** To correct the error that is reported for line 23, go to line 19 which is “using namespace std”. Place a semi-colon (;) at the end of that line so that it looks like “using namespace std;” and save the file.

28) **Compile the program again** using the command given in step 26, then run the program by typing the command `<./project1b>`. Use values of 1, 10.75 and 40 for the values asked for by the program.

⇒ 29) Type in the command `<ls>` and **write down the results shown**.

30) Steps 24 and 26 show the correct way to compile your program. Typing in the command differently or with typing errors can result in the irreversible loss of your program. Therefore, when compiling your program be sure to use the proper form of: **g++ source_code.cpp -o source_code** . After a successful compile, the file provided after the `-o` flag (i.e. source_code) is the executable program name.

31) The remaining files in the Example directory can be removed at one time by typing the command `<rm proj*>`. This command removes all files starting with “proj”

32) Change directories to your home directory by typing the command `<cd>` followed by the command `<pwd>` to verify that you are in your home directory.

33) Remove the Example directory by typing the command `<rmdir Example>`. If you receive the error message: `rmdir: directory "Example": Directory not empty`, move (change directories) into the Example directory and delete any files/directories that are present. Repeat this step until the Example directory is removed.

34) In this class, there will be 11 projects and one extra credit project assigned throughout the semester. Each project should be worked on in its own directory. This type of organization keeps all the files associated with each project in a separate directory. You just have to remember to work in that directory.

Therefore, the next task is to create the directories required for finishing this project and the next 10 projects. **The following is a short cut way to do it, and no other explanation on how this works is given. It is not recommended that you try similar techniques on your own unless you are willing to risk losing files.**

- First, type in the command `<pwd>` and verify that you are in your home directory. If you are not in your home directory, type in the command `<cd>` to move into your home directory.
- Create a directory called **CPE112_FALL12** by typing the command `<mkdir CPE112_FALL12>` and move into that directory by typing the command `<cd CPE112_FALL12>`

35) Create sub-directories for all the projects (and one for a possible extra credit project) by typing in the following command **exactly as shown**: (Note: The only space occurs between mkdir and Project)

```
<mkdir Project{_01,_02,_03,_04,_05,_06,_07,_08,_09,_10,_11,_12}>
```

Type in the command `<ls -l>` and verify that 12 directories with the names Project_01, Project_02, ... Project_12 have been created.

⇒ Show your TA that the files exist in the proper location and have the TA initial the answer sheet

Now if you are going to work on Project_02, you will open a new terminal window and move into the appropriate subdirectory by typing the command `<cd CPE112_FALL12/Project_02>`. The previous command assumes you are in your home directory – which you should be if you are in a newly opened terminal window.

When you download files from ANGEL be sure to download them into the correct sub-directory for the current project.

Project 1 Part 2: Practice Submitting Programs

For this part of the first project, you are to **perform a practice submission** for a program. The instructions on how to submit a program are in the project description for the **practice project**. The instructions are in the practice project page (and below) under the **Project Tab on ANGEL**. The file Practice.cpp is available for downloading from the practice project page as well.

**Submit the file Practice.cpp to the Practice Program drop box.
Also submit the Practice.cpp file to the project 1 drop box.**

Turn in the answer sheet to your TA for grading.

APPENDIX: Submission of Assignments Via ANGEL

To submit an assignment to an ANGEL Drop Box, follow the steps outlined below.

- 1) The submission deadline is on the assignment handout. **ANGEL strictly enforces the submission deadline**; therefore, it is in your best interest to finish the assignment well ahead of the deadline.
- 2) To submit a completed assignment, go to the drop box for the assignment:
(Typical location is: Projects Tab → Project # → Project # Drop Box).
 - **On this page, enter a title for the submission – NOTE: This is required.**
 - Enter in a message if desired.
 - Attach the file by clicking on the **Attachments** button.
 - a. Click on the **Browse** button to look for the file on the computer.
 - b. After selecting the file to submit, click on the **Upload File** button.
 - c. The file appears in the area labeled **Uploaded Files**. When all files have been uploaded, click on the **Finished** button. (For multiple files, you can set the window up for drag-and-drop so that all files can be selected and dropped in to the Uploaded Files area at one time).
 - You are returned to the assignment drop box page, and the file you uploaded is now listed. **Click on the Submit button below the file to submit the file.**
 - A page with a status message is displayed. Click **OK** to return to the assignment drop box page. At the bottom of the page is a list of all submissions you have made for this particular drop box.
- 3) Once a submission has been made, students cannot retrieve, modify or delete it.
- 4) **If a submitted file needs to be changed to a newer version, just submit the newer version. ←**