BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI (RAJ.)
CS F111 Computer Programming

LAB SESSION #2

(Unix Commands and Sublime Text Editor)

As discussed in Lab 1, to do any task in Unix, certain commands need to be executed. We have studied a few Unix commands in the last Lab and today we will be discussing some other Unix commands. We will also discuss how to use the sublime text editor for creating text files in both ubuntu and WSL.

Grep Command:

The ***grep*** command searches a named input file (or the standard input, if no input is mentioned) for a particular pattern of characters and displays all lines that contain that pattern. By default, it prints the matching lines.

Several options can be used with grep. For instance, grep –i "hello" file1.txt prints out all lines of file1.txt containing the pattern "hello", regardless of whether uppercase or lowercase letters. Find out the meaning of the following three other options to grep using the manual page: –v, –c, and –o. Try out using some examples, and then note down the meaning of these options.

Example: Consider that there is a file named demo.txt stored in your home directory

```
THIS LINE IS THE 1ST UPPER CASE LINE IN THIS FILE.
this line is the 1st lower case line in this file.
This Line Has All Its First Character Of The Word
With Upper Case.


Two lines above this line is empty.
And this is the last line.
```

Demo.txt

Command1: grep "this" demo.txt
Output: this line is the 1st lower case line in this file.
Two lines above this line are empty.
And this is the last line.
Observation: Search is case-sensitive

Command 2: grep -i "this" demo.txt
Output: THIS LINE IS THE 1ST UPPER CASE LINE IN THIS FILE.
This line is the 1st lower case line in this file.
This Line Has All Its First Character Of The Word With Upper Case.
And this is the last line
Observation: All lines containing the keyword "this" (not case-sensitive) are displayed.

Try executing the grep command with other options.

Pipe command:

The Pipe is a command in Linux that lets you use two or more commands such that the output of one command serves as input to the next. In short, the output of each is processed directly as input to the next one like a pipeline. The symbol '|' denotes a pipe.

Example:

ls -l | grep ".txt" The output generated by the ls –l command is provided as input to the grep command and the grep command searches for the presence of the keyword ".txt" in the input (thus fetching all files which end with .txt)

Input and Output Redirection

Most Unix system commands take input from your keyboard and send the resulting output to your terminal. *A command normally reads its input from the standard input, which happens to be your keyboard by default. Similarly, a command normally writes its output to standard output, which is again your terminal by default.* However, these default streams can be changed by using input/output redirection so that the input is taken from a file, and the output goes to a file, for instance.

The output from a command normally intended for standard output can be easily diverted to a file instead. This capability is known as output redirection. If the notation **> f1** is appended to any command that normally writes its output to standard output, the output of that command will be written to file **f1** instead of your terminal. This is ***output redirection***. Just as the output of a command can be redirected to a file, so can the input of a command be redirected *from* a file. As the greater-than character **>** is used for output redirection, the less-than character **<** is used to redirect the input of a command. This is ***input redirection***.

Example:

wc –l test.c **>** text.txt
Writes the output of wc –l command in text.txt instead of terminal. The content of the text.txt gets overwritten.
To avoid overwriting text.txt, >> is used instead of >. Use of >> append the new text at the end of the file.

Similarly, you can observe the use of input redirection by redirecting a file to wc –l:
wc –l < test.c
The file is given as an input to wc –l, and the no. of lines in the file is printed on the screen. Can you observe some difference in the output of the command above and "wc –l test.c"?

Attempt each of the following tasks on the Linux machine, and write down the corresponding Linux commands you used for each task in your notebook:

1. How do you make the system print out the name of <u>your home</u> directory?

You notice that a long string starting with the / character is printed, with words separated by the / character again. The starting / is called the root directory under which all other directories and sub-directories reside.

2. What is the <u>parent directory</u> of your home directory?

Just as the home directory is denoted by the ~ symbol, the current directory is denoted by the symbol . (period), while the parent is denoted by .. (two periods).

3. Make a directory called **dir1** under your home directory.
    a) Change to this directory.
    b) Make three empty files: **file1 file2** and **file3** under the current directory.
    (You can either use the **touch** command or the cat > *filename* operator. Try both options.)
    c) Make a sub-directory called **dir1-1** under the current directory.
4. Draw the file structure of the existent files and directories under your home directory.
5. Remove the subdirectory **dir1-1**.
6. Try now removing **dir1**

7. Store the list of all the files and directories (using ls long listing) in a file called **dirfile**.

8. Display the contents of **dirfile** on the screen using the **cat** command

9. Store in a file called **userlist** the list of users who are currently logged into the system

10. Combine the contents of **dirfile** and **userlist** and store in the file called **file1**

11. Print the number of lines, words and characters in **file1** using the **wc** command.

12. Copy the contents of **file1** into **file2**.

13. Append the following two lines into **file2**:

    This is file2.

    And I am using Linux!

Using sublime text editor in Linux:

Please follow the following videos that instruct about editing text files using sublime text and accessing/editing them on the Linux shell prompt.

For Students who are using Ubuntu (installed independently or through Virtual Machine, here are the instructions:

https://drive.google.com/drive/folders/1RgAYUlrgXTCU2mcWUe4WriCOj792F1C_?usp=sharing

For Students who have installed WSL over Windows, here are the instructions:

https://drive.google.com/drive/folders/1ZMM-L-gtVzfiTleUhbMeiR1LT05LqAFz?usp=sharing