

Lab 1 : Working with Linux

Objective

Lab 1 is intended to provide an introduction to Linux. The lab consists of performing basic system operations such as file management, text editing and permission management. The objective of this lab is to make students familiar with the Linux command-line environment and develop the skills of shell scripting. This lab serves as a platform for the subsequent labs related to system calls, process management, file management and memory management.

Recommended Systems/Software Requirements

Any flavor of Linux. In our case, we are using lubuntu as it is light weight and thus suitable for using in vm.

Preparation

Download and Install Oracle Virtualbox from <https://www.virtualbox.org/wiki/Downloads> in your computer. (Use older version 5.2 if you need support for 32-bit hosts).

Download lubuntu from <https://lubuntu.net/downloads>.

Create a virtual machine in Virtualbox and install lubuntu in it. You may follow the tutorial <https://dwheeler.au.com/2014/01/25/setting-up-a-lubuntu-virtual-machine-with-virtual-box/>

Tutorial

1. Login to the system, open the Terminal and type the following on the login prompt:

```
echo hello world
```

The words hello world would get displayed as output on the terminal. The *echo* command displays a line of text. Type *man echo* to read what an echo command does.

2. Linux provides on-line manuals for different commands through an interface called *man*. To know about *man* type the following on the terminal: *man man* and read the description that is displayed.

For all commands we will use the syntax *man < commandname >*. Type

```
echo $SHELL
```

This prints */bin/bash*. Bash is the name of the login shell that is currently in use.

3. Study the following Unix/Linux general purpose utility commands using their *man* pages and execute them on the terminal: *man*, *who*, *cat*, *cd*, *cp*, *ps*, *ls*, *mv*, *rm*, *mkdir*, *rmdir*, *echo*, *more*, *less*, *date*, *time*, *kill*, *history*, *chmod*, *chown*, *finger*, *pwd*, *cal*, *logout*, *shutdown*. Show at least an example of each command usage and explain what happened after the command.

4. Study the following Linux commands: *sed*, *grep* and *awk*. Use the *sed* command to delete the first character and last character in each line of a file. Use the *grep* command to find how many lines of a file contain a given word. The file name and the word are provided as inputs. Show the complete command line instruction for each case.

Shell Scripting

You are advised to refer to the text book mentioned bellow or any suitable online resource on shell scripting.

Write a shell script to display "HELLO WORLD" on the terminal:

- Open an editor
- Type *echo HELLO WORLD*
- Save the file with .sh extension (say test.sh)
- Close the editor
- At the terminal, type *sh test.sh*

Expected output at the prompt : HELLO WORLD

Write the full command line instructions for each of the following tasks.

5. Use the *who* command and redirect the result to a file called *myfile1*. Use the *more* command to see the contents of *myfile1*.

6. Use the *date* and *who* commands in sequence (in one line) such that the output of *date* will display on the screen and the output of *who* will be redirected to a file called *myfile2*. Use the *more* command to check the contents of *myfile2*.

7. Write a *sed* command that swaps the first and second words in each line in a file.

8. Write

- a shell script program and
- a C program to display "HELLO WORLD"

Compare the running time of both the programs using *time* command.

9. Write a shell script that takes a command line argument and reports on whether it is directory, a file, or something else.

10. Write a shell script that accepts one or more file name(s) as arguments and converts all of them to uppercase, provided they exist in the current directory.

11. Write a shell script that determines the period for which a specified user is working on the system.

12. Write a shell script that accepts a file name, starting and ending line numbers as arguments and displays all the lines between the given line numbers.

13. Write a shell script that deletes all lines containing a specified word in one or more files supplied as arguments to it.

14. Write a shell script to perform the following string operations:

- To extract a sub-string from a given string
- To find the length of a given string

References

1. Unix concepts and applications, Fourth Edition, Sumitabha Das, TMH.
2. Unix and shell Programming, B.A. Forouzan & R.F. Giberg, Thomson.
3. Beginning shell scripting, E. Foster, Johnson & other, WileY India