

INDIAN INSTITUTE OF TECHNOLOGY GOA

OPERATING SYSTEMS LAB (in CS310)

Faculty: Dr. Sharad Sinha
Teaching Assistant: Prachi Kashikar

LAB 01

In this lab, you will be introduced to the Linux operating system. The lab work consists of performing basic system operations such as file management, text editing and permission management. The objective of this lab is to make you familiar with the Linux command-line environment and develop the skills of shell scripting. In this lab, you will also be introduced to Linux system calls. These system calls are used in order to make input-output operations on files, as well as operations to handle files and directories in Linux. This lab exercise will be split in to two parts.

PART A:

In this part, you will be introduced to the Linux environment and commands. You will be made familiar with shell scripting. You will also be introduced to the concept of System calls.

STEPS:

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

- a) echo Hello World

The word 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.

- b) 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 you will use the syntax man < commandname >

- c) Type echo \$ SHELL : This prints /bin/bash bash is the name of the login shell that is currently in use
- d) 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, date, time, kill, history, chmod, chown, finger, pwd, cal, logout, shutdown, uname, netstat, sed, grep and awk.

2) Shell Scripting:

Shell is a UNIX term for the interactive user interface with an operating system. The shell is the layer of programming that understands and executes the commands a user enters. In some systems, the shell is called a command interpreter. A shell usually implies an interface with a command syntax (think of the DOS operating system and its "C:>" prompts and user commands such as "dir" and "edit").

Shell Types:

In Unix, there are two major types of shells –

Bourne shell – If you are using a Bourne-type shell, the \$ character is the default prompt.

C shell – If you are using a C-type shell, the % character is the default prompt.

Usually shells are interactive i.e., they accept command as input from users and execute them. However, some time we want to execute a bunch of commands routinely, so we have to type in all commands each time in terminal. As shell can also take commands as input from file we can write these commands in a file and can execute them in shell to avoid this repetitive work. These files are called Shell Scripts or Shell Programs. Shell scripts are similar to the batch file in MS-DOS. Each shell script is saved with .sh file extension e.g. myscript.sh

a) 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

Some more examples of **shell scripts** are provided as reference material along with this manual.

3) Learning how to use the system calls for opening, reading and writing to files - open, read, write and lseek system calls.

- a) Use the man command to read the manual pages of open, read, write and lseek system calls.
- b) Download and compile the file **system_calls_open_read_write**, which is provided along with this manual. This program intends to read 100 characters from a file and to print these characters on the terminal. It uses open system call

to open a file. The open call returns a file descriptor fd which is then used in the successive read and write system calls to access the file.

- c) Download and compile the file **system_calls_write_lines_of_text** which is provided along with this manual.
- d) Download and compile the file **simulation_ls** which is provided along with this manual. This program performs the simulation of ls command which lists all the folders and subfolders present in the current working directory.

PART B:

Perform the following exercises:

- 1) Use the who command and redirect the result to a file called myfile1. Use the more command to see the contents of myfile1.
- 2) Use the sed command to delete the first character and last character in each line of a file.
- 3) Use the grep command to find how many lines of a file contain a given word. The filename and the word are provided as inputs.
- 4) 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.
- 5) Write a (i) shell script program and (ii) C program to display "HELLO-WORLD"
 - a) Compare the running time of both the programs using time command
- 6) Write a shell script that accepts one or more file name as arguments and converts all of them to uppercase, provided they exist in the current directory.
- 7) Write a shell script to perform the following string operations:
 - a) To extract a substring from a given string
 - b) To find the length of a given string
- 8) Implement in C the following Linux commands using System calls: cat and mv