

National University of Computer and Emerging Sciences



Laboratory Manual

for

Operating Systems Lab

(CL-220)

Course Instructor	Mr. Saifullah Tanvir
Lab Instructor(s)	Ms. Saba Traiq Mr. Irtaza
Section	CS-4J
Semester	Spring 2023

Department of Computer Science

FAST-NU, Lahore, Pakistan

Tentative Course Outline:

<i>a</i>	<i>Instructor</i>	<i>Lab</i>	<i>Topic</i>	<i>Contents</i>
1		lab 1	Introduction to operating system (ubuntu) and writing first program	<ul style="list-style-type: none"> • Introduction to operating system • Writing first program in C++ in Ubuntu
2		lab 2	Linux system calls,Fork system call and	1 Working with Linux system calls 2 Creating process using Fork system call 3 MakeFile Utility .
3		lab 3	MakeFile Utility	Introduction to MakeFileUtility
4		lab 4	pipes	Information sharing between processes using unnamed pipes
5		lab 5	Shared Memory	Shared Memory through POSIX API
6		lab 6	Threads and Multithreading	<ul style="list-style-type: none"> • Understanding difference between threads and processes • Concurrency in threads. • Using pthread library.
8			Mid Term	
7		lab 7	synchronization through Semaphores and mutexes	synchronization through Semaphores between threads
9		lab 8	synchronization between two Processes	• Memory Based(Unnamed) Semaphore Shared between two Processes
10		lab 9	Mid solution	Mid solution and queries entertained
11		lab 10	File System	Implementatation of basic file system
12		lab 11	Memory	Memory numeriacals.
14			Final Exam	

Tools:

We will be working in Linux OS. As most of our systems are windows based, we will be using Ubuntu as guest OS. Tools required for lab tasks are:

1. VMware/Virtualbox
2. Ubuntu/fedora(any version)
3. Or WSL can be used too

Installation:

You can download virtualbox from the following link:

<https://www.virtualbox.org/wiki/Downloads/WindowsHosts>

You can download execution file of Ubuntu from the following link:

<https://ubuntu.com/download/desktop/download/22.04LTS>

After downloading both, installation processes starts. You have to view following video to do that:

<https://youtu.be/v1JVqd8M3Yc>

You need to share folder between hosts and guest OS. To do that watch following video:

<https://www.youtube.com/watch?v=GZBiyKfSTA4>

Your installation is done now. Let's start working!!

Objectives

In this lab, students will:

1. Practice Basic commands on terminal
2. Develop a small program in C for reading/writing files

Basic Commands

- Clear the console: **clear**
- Changing working Directory: **cd Desktop**
cd Home
- List all files in directory: **ls**
- Copy all files of a directory within the current work directory: **cp dir/***
- Copy a directory within the current work directory: **cp -a tmp/dir1**
- Look what these commands do
cp -a dir1 dir2
cp filename1 filename2

Compiling C and C++ Programs on the Terminal:

For C++:

Command: `g++ source_files... -o output_file`

For C:

Command: `gcc source_files... -o outputfiles`

Example:

`gcc main.c lib.c -o run.exe`

Passing Command Line Arguments to a C/C++ Program

- Command line argument is a parameter supplied to the program when it is invoked. Command line argument is an important concept in C/C++ programming. It is mostly used when you need to control your program from outside. Command line arguments are passed to the `main()` method.

- To pass command line arguments, we typically define `main()` with two arguments: **first argument counts the number of arguments** on the command line and **the second is a pointer array which holds pointers of type char which points to the arguments** passed to the program. The syntax to define the main method is **`int main (int argc, char *argv[])`**.
- Here, **`argc`** variable will hold the number of arguments pass to the program while the `argv` will contain pointers to those variables. **`argv[0]` holds the name of the program while `argv[1]` to `argv[argc]` hold the arguments.**
- Command-line arguments are given after the name of the program in command-line shell of Operating Systems. Each argument separated by a space. If a space is included in the argument, then it is written in `""`.

In Lab Tasks

Question 1: (1 marks)

See the usage of the following commands online. Also, run them on the terminal.

- | | |
|------------------|---------------------------------------------------------------------------|
| 1. pwd | Lists current directory |
| 2. ls | Lists all files/folders in the current directory |
| 3. cd | Enters a directory. SYNTAX: cd xxxx xxxx is the path |
| 4. cp | Copies from one directory to another |
| 5. mkdir & rmdir | Makes a new directory |
| 6. man | displays the user manual of any command that we can run on the terminal |
| 7. sudo | sudo allows a permitted user to execute a command as the superuser |
| 8. apt-get | Installs other operations on other softwares |

Question 2: (2 marks)

- Create a file named **main.c** and write a code to print "Welcome to BSBS Operating System Lab Course" on terminal.
- main.c** file contains the main function receiving **command-line arguments**.
- You will pass the name of Course via these arguments.

Question 3: (3 marks)

- Write a program that takes multiple numbers from the user through **command line arguments**.
- Print the **sum and average** of these numbers on the terminal.

Question 4: (4 marks)

- Write a program to copy numbers from one file to another.
- Create a function `removeNonAlphabets(char * inputFileName, char * outputFileName)` in C.
- This function reads the content of input file and writes only the numbers to the output file.
- The names of input and output files are passed through **command line arguments**.
- You can write any alphabets and numbers in the input file.

Note: You can use any mechanism for file-handling, in this task.