Work Integrated Learning Programmes

BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Department of Computer Science and Information Technology
NETWORK PROGRAMMING: CSIW ZC462
Laboratory Work Sheet - 1

1. Pre-requisite:

a) Knowledge of Linux and command line usage
b) Knowledge of C/C++ Programming
c) Understanding of Networking concepts (TCP/IP Model, Sockets)
d) Knowledge of Operating System Concepts (Process, Threads, File Descriptors)

2. Important Instructions:

a. Experiment should be conducted on the BITS OSHA Cloud laboratory only. **Experiments conducted outside this laboratory platform will not be evaluated.**

b. Save all your programs in home directory **(/home/cloud)** with proper program name.

c. Completed worksheets should be **uploaded in Taxila (e-learn) portal only**.
Start Date: 02-07-2025
End Date: 14-07-2025

d. This laboratory activity carries **5 Marks** weightage.
   - **i.** Part-I (1.5 Marks)
   - **ii.** Part-II (1.5 Marks)
   - **iii.** Part-III (2 Marks)

e. **Refrain from copying or sharing documents with others. Any evidence of such practice will attract severe penalty**.

f. **Copying source code from any webtools (ChatGPT,Google Gemini etc.) will result in 0 marks.**

g. Any queries related to LAB Activities should be sent to support team or to post in the discussion forum of the respective course.

h. **Attach the screenshot of the output in line with the question or at the end of the document mentioning proper question number**.

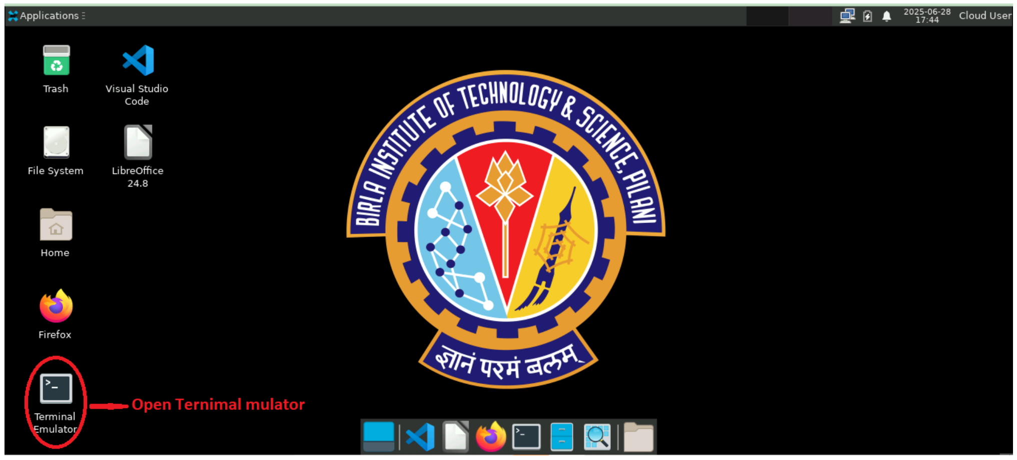
i. **Capture your observation in full screen covering your username.**

**************************FILL IN THE DETAILS GIVEN BELOW****************

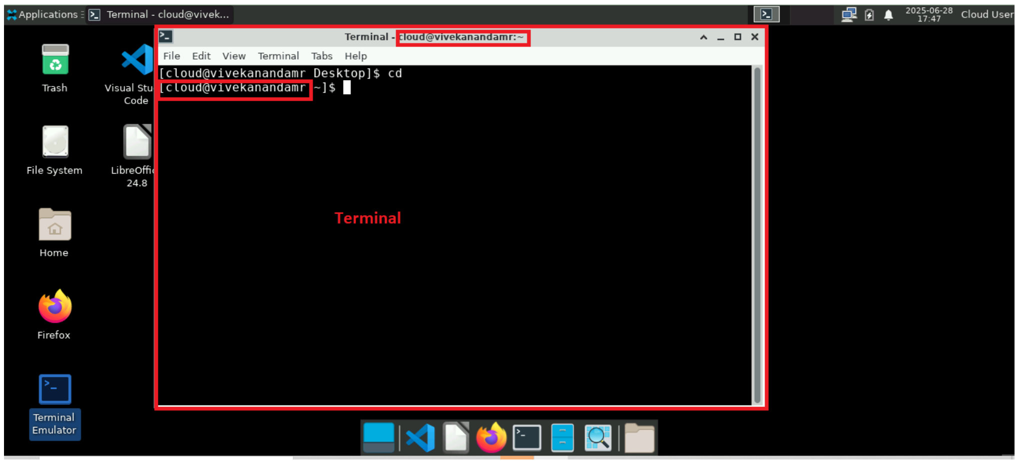
Sl. No	**Name as appeared in Taxila**	**BITS ID No.**
1.		

Steps to Open the vi Editor→ Enter Source Code → Compile and Execute the code

- After successful login to virtual labs , Open the **Terminal Emulator** application as shown in picture

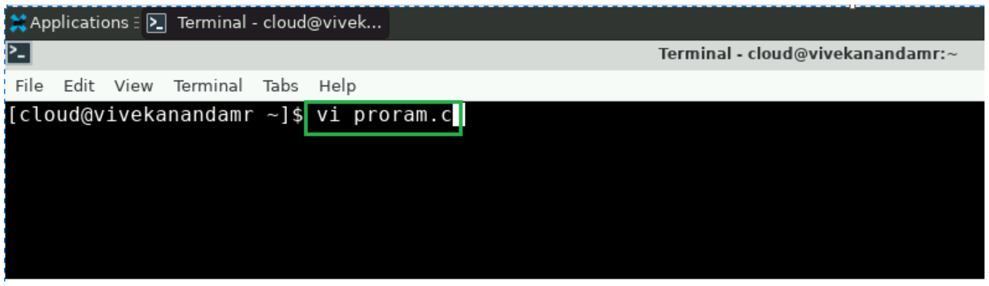


- You will be able to see the application window as shown below.



```
[cloud@vivekanandamr Desktop]$ cd
[cloud@vivekanandamr ~]$
```

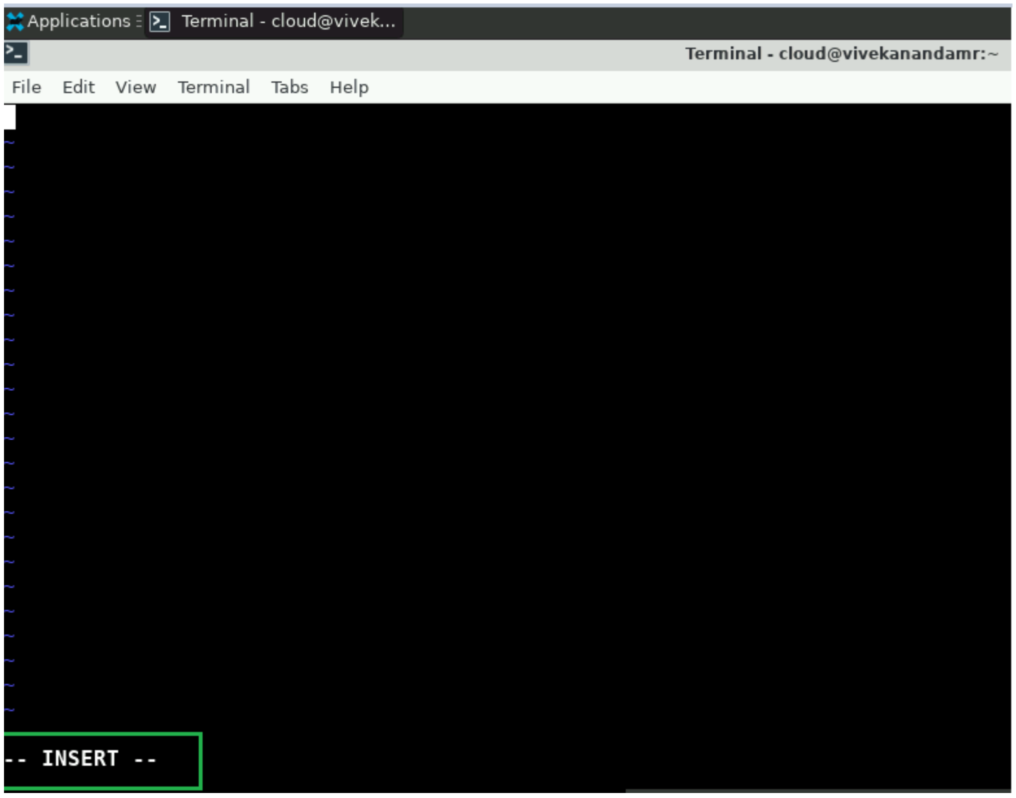
- Run the below commands to enter the source code in vi-editor
- Open vi-editor : **`vi program.c`** (Give the program name as per convenient) and


Applications
Terminal - cloud@vivek...
Terminal - cloud@vivekanandamr:~
File Edit View Terminal Tabs Help

```
[cloud@vivekanandamr ~]$ vi proram.c
```

then press Enter key to open the Text Editor.

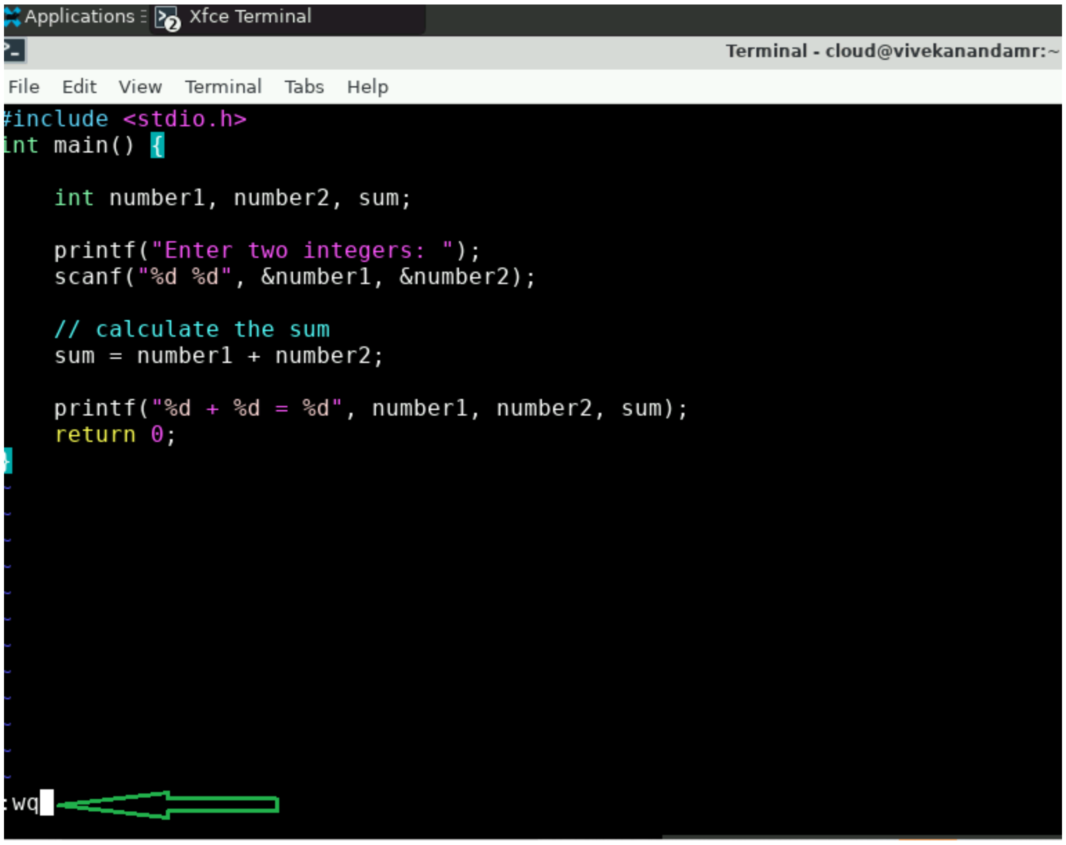
- Once Text Editor is opened press the key **I (Insert)** to enter the source code. You should be able to see **Insert** option in the editor once you press key **I.**


Applications
Terminal - cloud@vivek...
Terminal - cloud@vivekanandamr:~
File Edit View Terminal Tabs Help

```
-- INSERT --
```

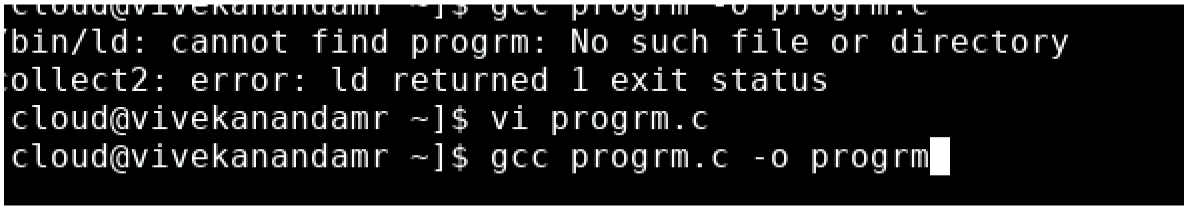
- Enter the program/source code in **vi-editor** shown in above picture

- Save the program by entering the command in editor **:wq (Escape+Shift+:)**


Applications
Xfce Terminal
Terminal - cloud@vivekanandamr:~
File Edit View Terminal Tabs Help

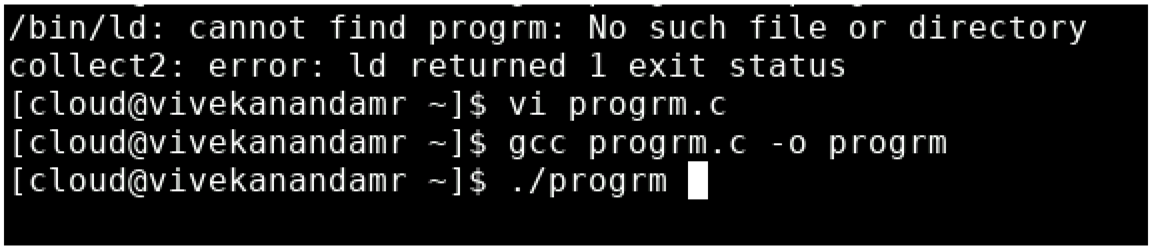
```
#include <stdio.h>
int main() {

    int number1, number2, sum;

    printf("Enter two integers: ");
    scanf("%d %d", &number1, &number2);

    // calculate the sum
    sum = number1 + number2;

    printf("%d + %d = %d", number1, number2, sum);
    return 0;
}
~
~
:wq
```

- **After saving the program, compile and run the program**
- **To compile the program : gcc program.c -o program**



```
bin/ld: cannot find progrm: No such file or directory
ollect2: error: ld returned 1 exit status
cloud@vivekanandamr ~]$ vi progrm.c
cloud@vivekanandamr ~]$ gcc progrm.c -o progrm
```

- **To Execute the program: ./program**



```
/bin/ld: cannot find progrm: No such file or directory
collect2: error: ld returned 1 exit status
[cloud@vivekanandamr ~]$ vi progrm.c
[cloud@vivekanandamr ~]$ gcc progrm.c -o progrm
[cloud@vivekanandamr ~]$ ./progrm
```

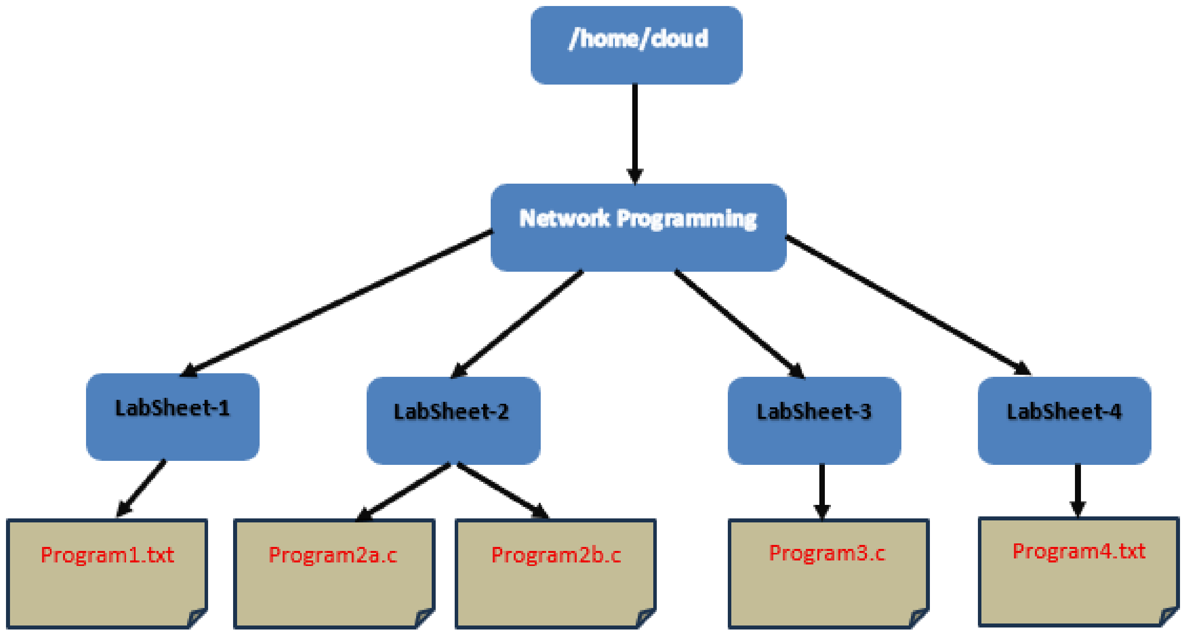
- **If it displays any errors correct the errors and compile an run the program**

Part-I (1.5 Marks)

Objective:

- **To understand the usage of linux commands**

1a) Use proper linux commands, implement a directory structure as per the tree given below. All those rounded rectangles are directories, and the rest are files



- Make use of **mkdir** command to create directories mentioned in the above tree
    - You can create the directory **Network Programming** under the **home** (/home/cloud) directory.
- Create text file(**.txt**) and c-program(**.c**) file using **vi** editor
- Create files as shown in above tree under respective directories.
- Make use of **cp** command to copy the files from one directory to another
    - You need to copy all the contents in the directory **LabSheet-3/Program3.c** to the folder **LabSheet1** by utilizing **cp** command.
- Explore the various options of command **ls**
    - Go to the directory **LabSheet-2** and list the files under this folder using the command **ls**
    - Explore the following options of ls command
        - ls -a
        - ls -ltr
        - ls -t

1b) Use proper commands to view Linux file permissions.

Tasks to be carried out:

i) Crate directory with **your name** under **/home/cloud**
ii) Create different types of files (.txt , .sh etc.) in the directory you have created
iii) List all the files you have created in the directory using proper commands.
iv) Create a file **project.txt** in directory you have created and suppose you're working on a project with a group. You want all group members to have read and write access to a file named **project.txt**, but no access for others.
    a. What group permission setting is required?
    b. Which command(s) would you use?
v) Given a file **secret.txt** owned by user Alice, with permission **-rw-r--r--**, explain why Bob (a different user) cannot modify the file, even if he can read it.

Note: Each step carried out should have supporting screenshot

1c) Explain the meaning of the following permission string: -rwxr-xr--. Break it down by owner, group, and others.

1d) Convert each of the following symbolic permissions into their numeric equivalent.

i) -rwxrwxrwx
ii) -rw-------
iii) -rwxr-xr-x
iv) -rw-r--r--
v) -rwx------

Part- II (1.5 Marks)

Objective:

You're building a mini shell in C. When a user types a command, your program should create a child process and execute the command using one of the exec family functions. The parent process should wait for the child to finish and then prompt the user again.

Write a C program that:

- **Accepts a command (e.g., ls -l, date, whoami) from the user.**
- **Splits the command into program and arguments.**
- **Forks a child process.**
- **In the child process, uses execvp() to run the command.**
- **The parent waits for the child to finish before prompting again.**

Run at least 3 different commands and capture their output.

Your Submission should include

- Your source code.
- A brief explanation of how fork() and execvp() are used.
- Screenshot or log of your program running.

PART-III (2 Marks)

a) **Execute the following C program, what is the output of calling fork() twice? Draw the process tree for your answer.**

```
#include <stdio.h>
#include <unistd.h>

int main() {
    fork();  // First fork
    fork();  // Second fork
    printf("Hello from process %d\n", getpid());
    return 0;
}
```

b) **Describe the return value of the fork() system call. What does a positive value indicate, what does a value of 0 indicate, and what does a value of -1 signify?**

Submission guidelines:

The student should be instructed to take a snapshot of the screen to showcase their usage of the lab utilization portal along with the required codebase and other needed information.

Screen shot for 1.	
Screen shot for 2.	
Screen shot for 3.	
Screen shot for 4.	
Screen shot for 5.	
Screen shot for 6.	
Screen shot for 7.	
Screen shot for 8.	
Screen shot for 9.	