

Tribhuvan University
Faculty of Humanities and Social Sciences



Lab report on:
Operating System Lab 2:
GNU/Linux

Submitted to:
Mr. Roshan Maharjan,
Er. Himal Chand Thapa ,
Department of Computer Application,
Himalaya College of Engineering,
Chyasal,Lalitpur

Submitted by:
Sujal Gurung
Roll no: 34
BCA II/IV
September 20, 2023

1 Objectives

- To be familiar with basic commands used in GNU/Linux operating systems
- To be familiar with `vi` editor
- To re-use previous lab's knowledge on using & navigating in the terminal instead of the GUI

2 Introduction

2.1 GNU/Linux

GNU/Linux (commonly referred to as only Linux) is a free and open source Operating System using the Linux kernel. It is inspired by & shares similarities with the old UNIX OS. It includes the `shell`, an interpreter that provides a command line interface for interacting with the OS. The terminal allows running executable script files or compiled binaries by passing them as text commands to the shell. There are many useful built-in commands known as GNU utilities. Users can also write their own scripts using the shell's scripting language. Thus, the terminal can be more powerful at doing many tasks than the GUI. There are many variations of GNU/Linux, called distributions. Some popular ones are: Ubuntu, Arch & Debian. The following commands should work on almost any distribution.

- `ls`
prints contents of current or specified directory.
- `cd`
change directory to specified one. If none specified, goes to home directory.
- `cp source-file destination`
creates copy of file
- `mv source-file destination`
moves file. If filename is specified instead of path, file is renamed.
- `touch file-name`
creates new file
- `rm file-name`
permanently deletes file
- **executing binary or executable text file**
Just enter path (absolute or relative to current directory). For example, if there is an `a.out` file in the current directory, the relative path to execute it would be `./a.out`. Here, `.` refers to the current directory. `..` would mean one folder above/outside the current one.

- `gcc`
GNU C Compiler. The basic command to compile a C file is `gcc file-name.c`. By default, it names the compiled binary file as `a.out`, though we can specify a filename using the `-o` flag.

2.2 Vi editor

Vi is a text editor that was created for use in UNIX & can run directly in the terminal. It, along with modern derivatives like Vim & NeoVim, are popular among programmers for their fast performance, features & customizability. Text editors like Notepad & VSCode will always enter text when keys are pressed. Vi is a modal editor i.e. it has different modes for interacting with the text & keys behave differently depending on the mode. Thus, Vi can be used fully without a mouse, increasing efficiency. The modes are:

- **Normal mode** for navigating through the text file. Can be entered by pressing `Esc`. `h`, `j`, `k`, `l` move the cursor left, down, up and right respectively. Some other movement keys include `w`, `b` for moving across words, `0`, `$` for moving to start & end of line. There are many more shortcuts that make navigation effortless.
- **Insert mode** for inserting text. While in Normal mode, `i` enters insert mode to the left of cursor, while `a` moves to the right. Keys behave normally.
- **Visual mode** for selecting text to manipulate. While in Normal mode, `v` starts selecting from cursor, while `V` selects entire line. Some manipulation shortcuts are `d` to delete, `y` to yank(copy), `p` to paste.
- **Command mode** for running Vi commands. Can be entered by pressing `:` while in Normal mode. Some commands include `:w` to write(save) file, `:q` to quit, `:sp` to split the window & allow having multiple files on the screen.

One of Vi's powerful features is the ability to chain shortcuts together. For example, `yy` yanks the entire line, while `5p` pastes yanked text 5 times.

3 Lab Work

NOTE: *In code snippets, \$ at the start of a line represents the shell prompt & is not part of the entered commands.*

3.0.1 Write c source file for execute “hello world“ and save it as file xyz.c using vim.

```
$ vim
```

Inside vim, we write the following program:

```
#include <stdio.h>
void main() {
    printf("Hello World");
}
```

Press `Esc` to enter normal mode, then type `:w xyz.c` & press `Enter`. This will write the contents to file `xyz.c`

3.0.2 Using terminal commands & vim, create file `calc.c`. Write C code for calculating interest in it, compile and execute it.

```
$ touch calc.c # creates calc.c file
$ vim calc.c   # opens file with vim
```

Inside vim we write the following program:

```
#include <stdio.h>
void main() {
    float p, t, r;
    printf("Enter principal, time(in years), & interest rate: ");
    scanf("%f %f %f", &p, &t, &r);
    printf("Simple interest will be: %f", p*t*r / 100);
}
```

In normal mode, type `:wq` to write & quit.

```
$ gcc calc.c -o calc # compiles calc.c to calc binary file
$ ./calc # run binary
Enter principal, time(in years), & interest rate: 100
2
5
Simple interest will be: 10.000000
```

4 Conclusion

Thus, we learned about the GNU/Linux family of operating systems. Similarly to the 1st lab with DOS, here, we learned to use GNU utilities for performing different actions. We also learned to use the `vim` text editor & its different features.