



## Operating Systems

### Lab – 08

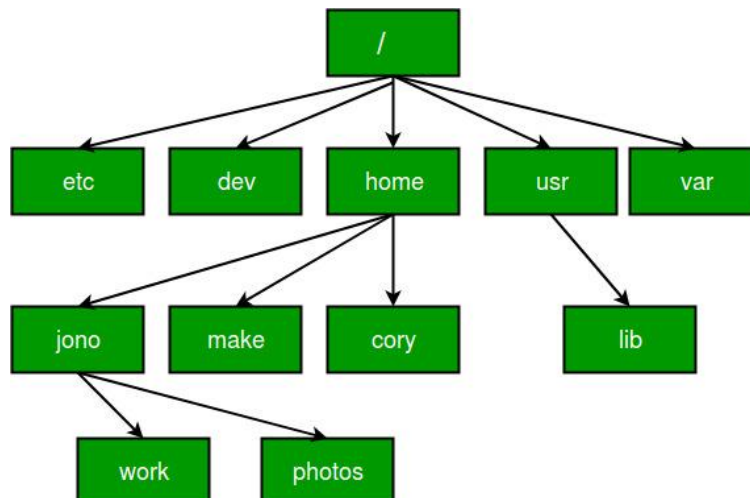
## Linux Environment

Perform all the tasks on your machine and write in your notebook the particular one's.

1. Create a **lab08/** directory on your desktop and perform the practical tasks in it.

#### Task 01:

- a) Differentiate between **internal** and **external** commands with examples?
- b) Define **vim** and explain all its 3 **modes**? (2)
- c) What is the **role** of **environment** variables in the system and how to **list** all of them?
- d) Draw the Diagram of **C compilation** and explain the process? (2)
- e) What is **ELF** and its **role** in the operating system?
- f) Differentiate between **Static** and **Dynamic** Linking with example?
- g) Analyze the image below and give the **absolute** path of the **photos** directory and also its **relative** path if you are in the **lib** directory?



- h) Write a command to **extract** the files from a **tarball**?
- i) How do **external** commands executed in the **shell** explain the procedure?

## Task 02:

- a) Create a file named **README** using **vim** and write “**Everything in UNIX is a file**” in it. Save and close the file.
- b) Open the file **README** again and copy the line and paste 5 times and then **delete** the 3rd and 4th line and then **undo** the last 2 things you have done using vim **modes**.
- c) Copy the file **/etc/passwd** and paste in your current directory **lab08/**. Now concatenate both files **passwd** and **README**.
- d) Create an **alias** named **clc** that does the work of clearing the screen. Make sure this alias will be **permanent**.
- e) Replace all the occurrences of your **username** with **linux** in the **passwd** placed in your current directory.
- f) Write a command to get **information** about how long the **system** has been running?

```
#include <stdlib.h>
#include <sys/types.h>
#include <unistd.h>
int main()
{
    pid_t child_pid = fork();
    if (child_pid > 0)
        sleep(50);
    else
        exit(0);
    return 0;
}
```

- g) Copy the code in some **.c** file, compile it and run the **binary** and Analyze the running processes in the system. Your task is to **find out** what the program is doing?
- h) Create a **local** variable in the shell and print its **value** in the string echoing in the terminal. (echo Learning **\$var** is fun) and var containing the word “**linux**”.
- i) Link a **C** program **statically** and **dynamically** using appropriate flags?
- j) Count the **nologin** string entries in the passwd file using a **single** line command (Use pipes).

---

“Opportunities don't happen, you create them.”