Operating System Lab

Fall 2024

Lab 02 Task

RIPHAH INTERNATIONAL UNIVERSITY

Alishba Waqar

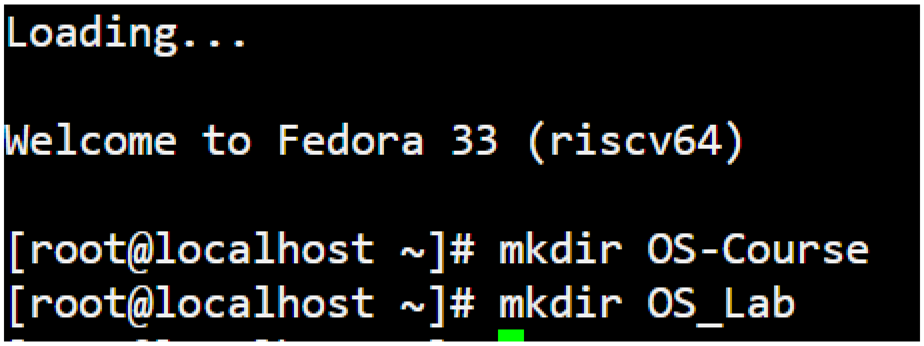
46997

Lab Instructor:

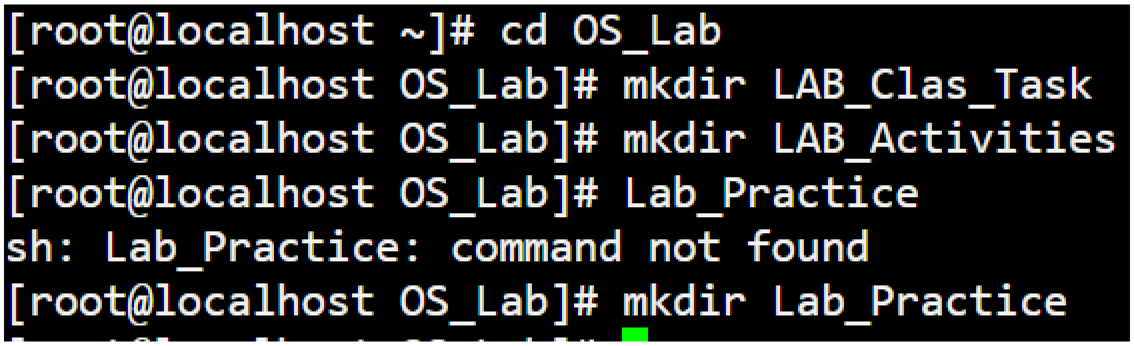
Kausar Nasreen Khattak

Lab Task

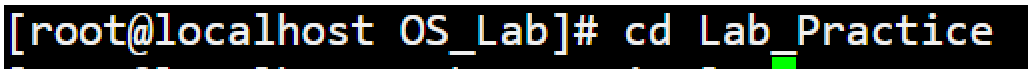
Q1. To begin, you need to set up a structured directory layout in your home directory. Start by creating two directories named **OS_Course** and **OS_Lab**. These directories will serve as the main folders for organizing your OS Lab tasks. After creating these directories, switch to the **OS_Lab** directory. Within OS_Lab, create three more directories named **LAB_Class_Task, LAB_Activities, and Lab_Practice**. Each of these directories will help you categorize different aspects of your lab work. Once you have created these directories, go into the **Lab_Practice** directory and create a file named example.cpp. This file should be empty and will be used for practice later. Finally, move back to your home directory. Make sure to take screenshots of each step, including the creation of directories, the file creation, and your navigation commands to document your process.
Note: Include screenshots, where required to illustrate your explanation.



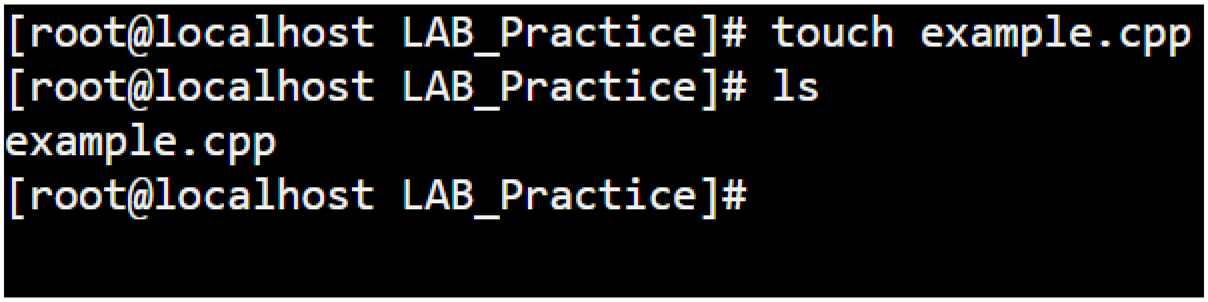
```
Loading...

Welcome to Fedora 33 (riscv64)

[root@localhost ~]# mkdir OS-Course
[root@localhost ~]# mkdir OS_Lab
```



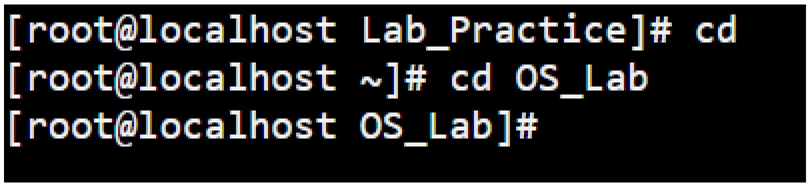
```
[root@localhost ~]# cd OS_Lab
[root@localhost OS_Lab]# mkdir LAB_Clas_Task
[root@localhost OS_Lab]# mkdir LAB_Activities
[root@localhost OS_Lab]# Lab_Practice
sh: Lab_Practice: command not found
[root@localhost OS_Lab]# mkdir Lab_Practice
```



```
[root@localhost OS_Lab]# cd Lab_Practice
```



```
[root@localhost LAB_Practice]# touch example.cpp
[root@localhost LAB_Practice]# ls
example.cpp
[root@localhost LAB_Practice]#
```

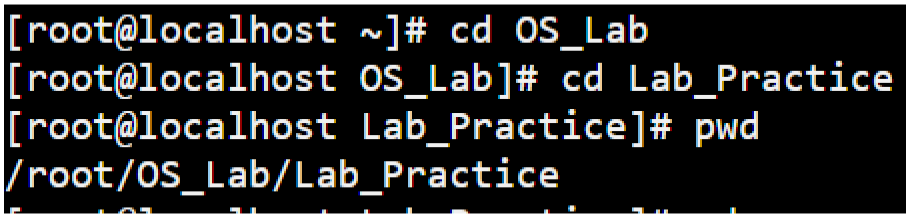


```
[root@localhost Lab_Practice]# cd
[root@localhost ~]# cd OS_Lab
[root@localhost OS_Lab]#
```

• mkdir OS_Course, mkdir OS_Lab
(creating two directories named OS_Course and OS_Lab)
• cd OS_Lab
(Switch to the OS_Lab directory.)
• mkdir Lab_Task_Class, mkdir Lab_Activities, mkdir Lab_Practice
(Within OS_Lab, create three more directories named LAB_Class_Task, LAB_Activities, and Lab_Practice.)
• cd Lab_Practice, touch example.cpp
(go into the Lab_Practice directory and create a file named example.cpp.)
• cd OS_Lab
(move back to your home directory)

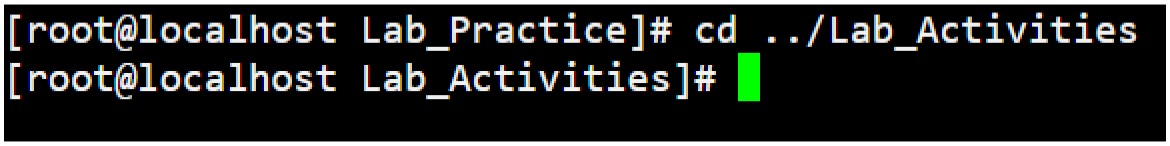
Q2. Finally, you need to understand the concepts of absolute and relative paths. Explain the difference between these two types of paths and provide an example of each. This will help you navigate directories more effectively. If you are currently in the Lab_Practice directory, describe the relative path to access the **LAB_Activities** directory. This will test your understanding of how to move between directories using relative paths.
Note: Include screenshots, where required to illustrate your explanation.

Absolute Path:
The complete path of any directory or file is known as the absolute path from the current working directory.
Example:



```
[root@localhost ~]# cd OS_Lab
[root@localhost OS_Lab]# cd Lab_Practice
[root@localhost Lab_Practice]# pwd
/root/OS_Lab/Lab_Practice
```

Relative Path:
The only path necessary to provide is to reach the file or directory from the current working directory.



```
[root@localhost Lab_Practice]# cd ../Lab_Activities
[root@localhost Lab_Activities]#
```

Q3. Imagine you’re working on your computer when you suddenly need to turn it off quickly. You press and hold the power button until the computer shuts down completely. After an hour, you turn the computer back on, and it quickly shows the login screen or desktop.
Why does your computer start up smoothly and quickly after being turned off? Describe the process that happens between powering off the computer and seeing the login or desktop screen. What steps does the computer go through to get everything ready in a short amount of time?

1. When you press the power button, the computer shuts down suddenly, which might leave some programs or files in an incomplete state.
2. Even when the computer is off, a tiny battery (the CMOS battery) keeps some important settings, like the time, date, and startup settings, saved.
3. When you turn the computer back on, a small program called the BIOS (built into the motherboard) starts running.
4. Self-Check (POST) The BIOS quickly checks the main hardware parts like the processor, memory, and hard drive to make sure everything is working properly.
5. Boot Device Loaded: The BIOS picks the device (usually your hard drive or SSD) where the operating system is stored.
6. A small program on that device (called the bootloader) is started. Its job is to find and load the operating system.
7. The operating system begins to load, starting with its core part, called the kernel.
8. The operating system loads drivers, which are small programs that help your computer talk to devices like your keyboard, mouse, and screen.
9. The operating system also starts any programs that are set to open automatically when you boot up.
10. Once everything is ready, you see the login screen or go directly to your desktop, depending on your settings.