**Assignment 2 – Working with Standard Linux commands**

**(Files & Directory manipulation)**

**What to and how to demo**

* **A complete document with blanks filled, Questions answered and necessary screenshots to be shown to your lab professor** (Use the same document to answer the question (fill in the blanks, and add screenshot where asked) *(10 Points)*
* **Due Date : Start of next Lab**

**Deliverables**

* This lab assignment is divided into 2 tasks.
* Use the same document to answer the question and add screenshots.
* After completing all tasks, show this document (with your answers and screenshot added) to your lab professors for grading and demo.
* Demo your work to your lab professor during lab class for grades.

(No submission will be accepted; student must demo during the lab class.)

* Note: All Linux Commands are with small letters!

(Total 10 points, you lose 1 point for each mistake, Missing Answers is considered as mistake (-1)).

Note: All Linux Commands are with small letters!

**Note:** Make sure you answer all questions with necessary screen shots, beyond the shadow of doubt. These are performance based questions so **you lose mark for each mistake**. Also make sure you understand what you are doing and if you had any question, ask your Lab professor, also take notes from your observation in the lab which helps you a lot in your future tests

**Working with Standard Linux commands**

* **cd –**  change directory
* **ls –**  list directory content
* **touch –** create an empty file
* **mv –** Rename/Move a file
* **mkdir –**  create one or more new empty directories
* **pwd –**  show present ( current ) working directory
* **rmdir –**  remove **empty** directories
* **rm –** delete a file
* **rm -r**  (delete nonempty directory)

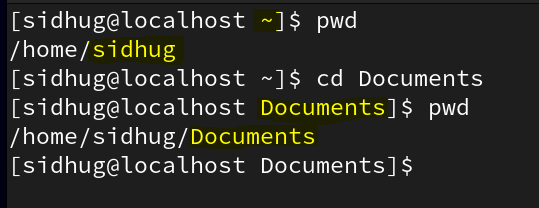
**Task 1**

## “cd” , “pwd”

As discussed directories are like nested room. Now to traverse (navigate) through the directories in Linux from **current** to **another** directory we use **cd** command

**$ cd *[directoryname]***

Typing **cd** with **no** ***directoryname*** argument will take you to your personal **HOME** directory (which is **not** the same thing as the directory called **/home** - be careful!). Providing a single ***directoryname*** parameter will change your shell's **current working directory** to the given directory. While you are working with the **cd** command, watch the shell prompt; it will change to display the **basename** of the current working directory after each **cd** command. (Refer screenshot below)



**Fill in the blanks**

1. At the command prompt type **cd** without any parameters. Record here the directory **basename** shown at the **right end** of the bash shell prompt:
2. Type **pwd** at the prompt and record the output here: **/home/jean0319**
3. **cd /** This will change the current directory to the top-level “**ROOT**” directory.

What directory **basename** is shown in the bash prompt after this command? **/**

1. Give the output of the **pwd** command now: **/**
2. **cd /etc** What directory **basename** is shown in the bash prompt after this command? **/etc**
3. Give the output of the **pwd** command now: **/etc**
4. **cd ..** (**Two periods**.) This command will “go up” one directory level (to the **ROOT**).

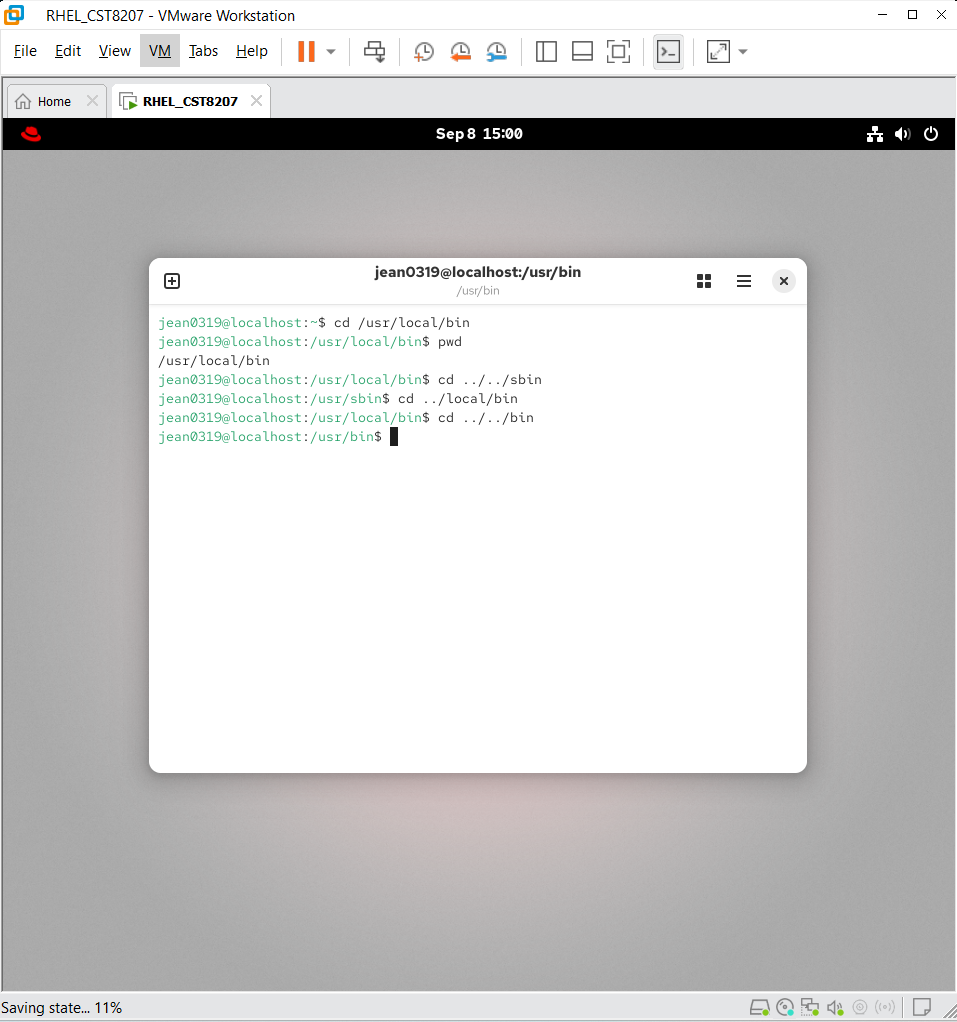
What directory **basename** is shown in the bash prompt after this command? **/**

1. Give the output of the **pwd** command now: **/**
2. **cd home/*user1 (is this path valid ?) no***  
   What directory **basename** is shown in the bash prompt after this command? **No such file or directory**
3. Give the output of the **pwd** command now: **/**
4. **cd /usr/local/bin** What is the **basename** in the bash prompt after this command? **/usr/local/bin**
5. Give the output of the **pwd** command now: **/usr/local/bin**
6. **cd ../../sbin** What is the **basename** in the bash prompt after this command? **/usr/sbin**
7. Give the output of the **pwd** command now: **/usr/sbin**
8. **cd ../local/bin** What is the **basename** in the bash prompt after this command? **/usr/local/bin**
9. Give the output of the **pwd** command now: **/usr/local/bin**
10. **cd ../../bin** What is the **basename** in the bash prompt after this command? **/usr/bin**

r) What is the output of the **pwd** command now: **/usr/bin**

Ans :-

add screenshot



## ls

To list the content of a directory we use **ls** command

Two important options is : **-a** to show **hidden** files and **-l** (**long** listing) as follows:

* **ls -a -l (using 2 different options (two arguments to command ls) )**
* **ls -la (creating a bundle of 2 different options (one argument to command ls) )**

### Perform the following commands and observe output

1. **ls /bin/ls [/bin/ls]**
2. **ls -l /bin/ls** [-rwxr-xr-x. 1 root root 141184 Nov 25 2024 /bin/ls]
3. **ls -li /bin/ls [17804742 –rwxr-xr-x. 1 root root 141184 Nov 25 2024 /bin/ls]**

Question : - what is the function of -i option that we used above with ls command (Hint: refer manual page for ls command ): the function of the –i option is to print the index number of each file

Ans :

NOTE: Replace student with the name of your Personal HOME directory name in executing below command lines.

1. **ls /home/student [doesnt return anything]**
2. **ls -a /home/student [. ..]**
3. **ls -al /home/student**  
   **total 4**  
   **drwxr-xr-x. 2 jean0319 jean0319 6 Sep 4 18:34 .**  
   **drwx------. 15 jean0319 jean0319 4096 Sep 4 18:36 ..**
4. **ls -ld /home/student**   
   **drwxr-xr-x/ 2 jean0319 jean0319 6 Sep 4 18:34 /home/jean0319/41269022**

If you pay attention. **s -al** shows files which you don’t see with **ls -l**

-a provides information about hidden files (remember in Linux everything is treated like a file.

Hidden files are high risk files that need to be another layer of protection. We will talk about in future more.

What is the function of using option -d

**Ans:- the function of using option –d is to list directories themselves, and not their content**

## mkdir , touch

To create one or more new, empty directories use **mkdir** command

To create new, empty file, use **touch** command

Execute following commands under your HOME directory **(/home/student**) in sequence.

$ **mkdir lab2**

$ **cd lab2**

$ **mkdir dir1 dir2**

$ **ls**

Screenshot here



$ **cd dir1**

$ **ls –a**

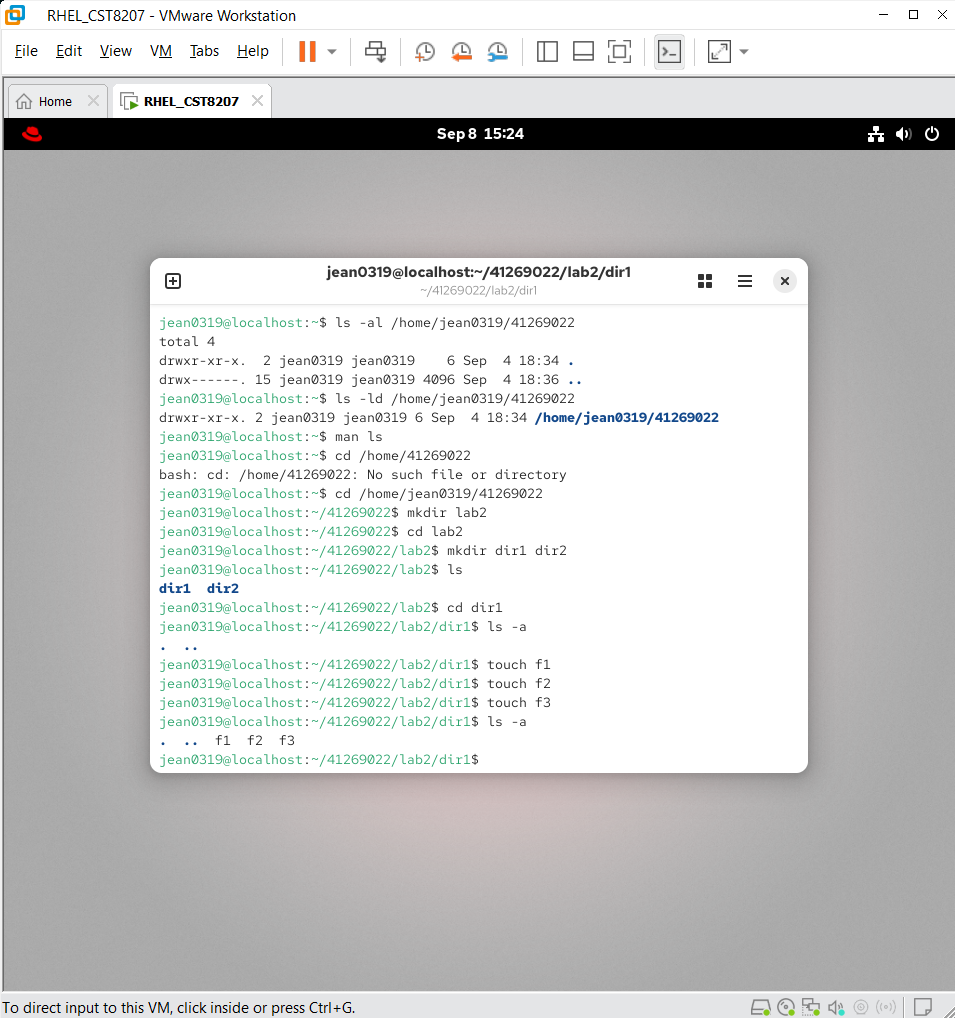
$ **touch f1 #create file called “f1” in dir1 directory**

$ **touch f2 #create file called “f2” in dir1 directory**

$ **touch f3 #create file called “f3” in dir1 directory**

$ **ls –a**

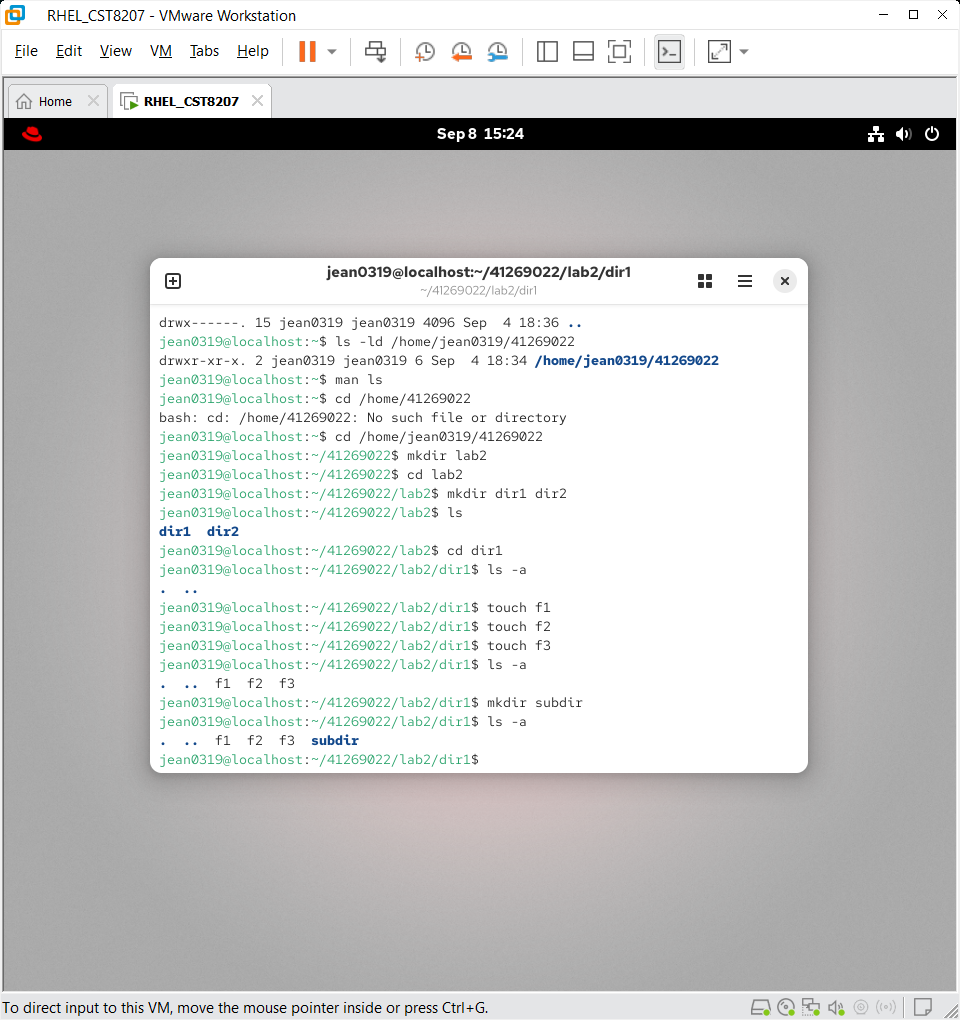
Screenshot here



$ **mkdir subdir**

$ **ls –a**

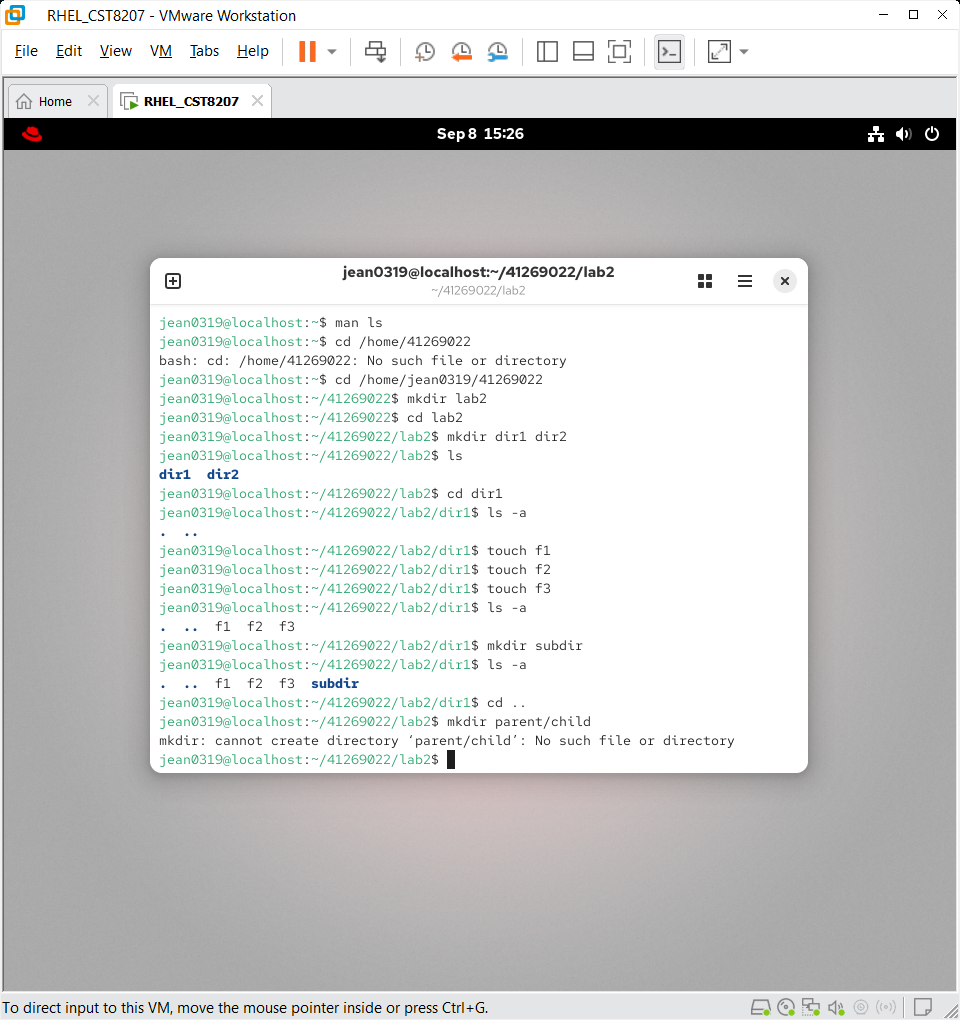
Screenshot here



$ **cd ..** Note: *two dot means: go up one directory level*

$ **mkdir parent/child**

Screenshot here



1. **Explain** **why** the above command **failed** and did not execute as expected:

The command failed because we are trying to make a directory child within the parent directory that doesn't yet exist

1. $ **mkdir –p parent/child** *# look up* **-p** *in the man page for* **mkdir**
2. The above command **succeeds** with no errors. What does the **-p** option to the **mkdir** command do?  
   The –p option creates a parent directory if there isn't one already, “make parent directory as needed”

## rmdir , rm , mv

To remove one or more EMPTY directories, use **rmdir** command

Execute following commands in sequence and use ls to verify what you have created

$ **cd (executing cd command without any argument will thake you to which directory ) Will take you to your home directory**

Ans:

Now being in this directory execute below commands

$ **mkdir lab2A**  *# create a new, empty sub-directory*

$ **cd lab2A**  # *make* ***lab2A*** *the current directory*

$ **mkdir dir1 dir2 test** *# create three new, empty directories*

$ **touch f1 f2 f3** *# create three new files*

$ **ls -l #** *option* **-l** *is lower-case letter* **L***, not the digit* **1**

**$ mv f3 f4** *#rename f3 to f4*

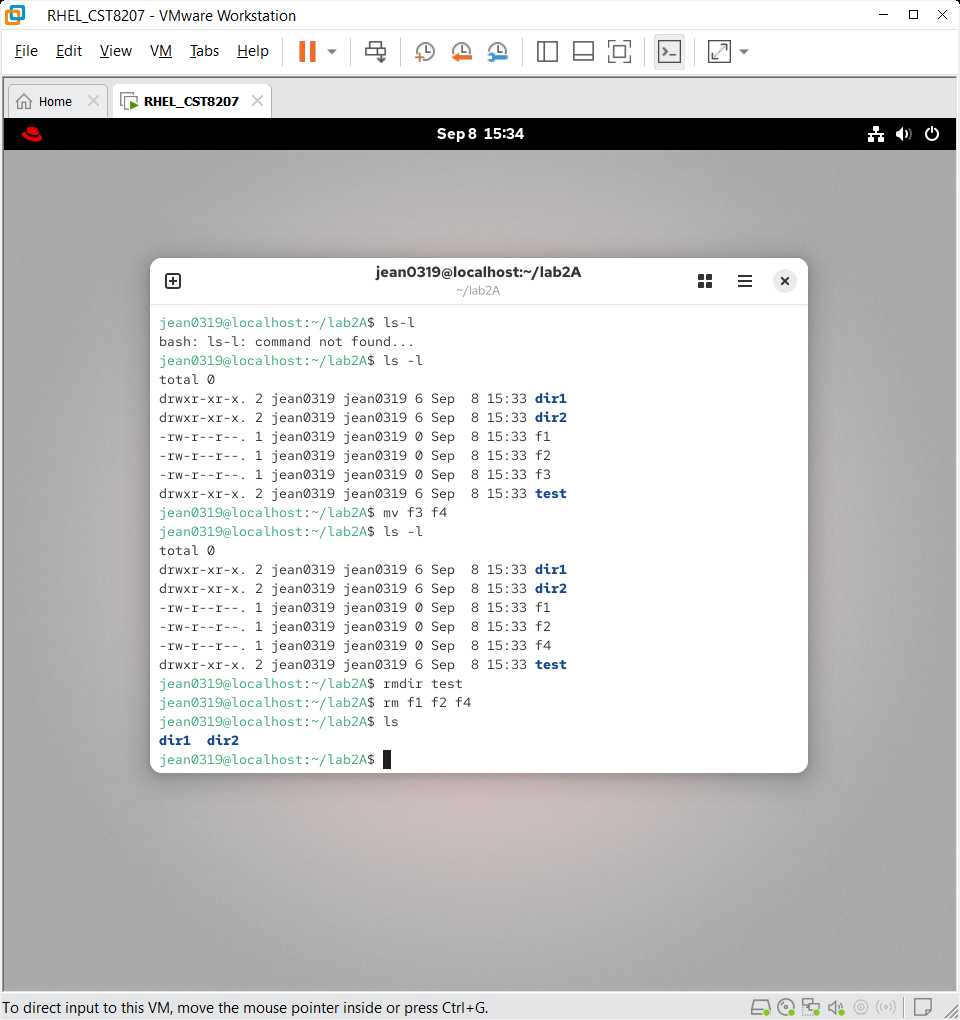
**$ ls –l**

What happened to f3 file? it was renamed to f4

$ **rmdir test** *#delete test directory*

$ **rm f1 f2 f4**  *#delete f1 f2 f4 files*

$ **ls**

Screenshot here   
  


$ **mkdir –p dir1/subdir parent/child**

$ **cd dir1**

$ **rmdir dir2** # *this* ***fails*** *with an error message*

What is the error message: no such file or directory error

1. Explain **why** the command failed.

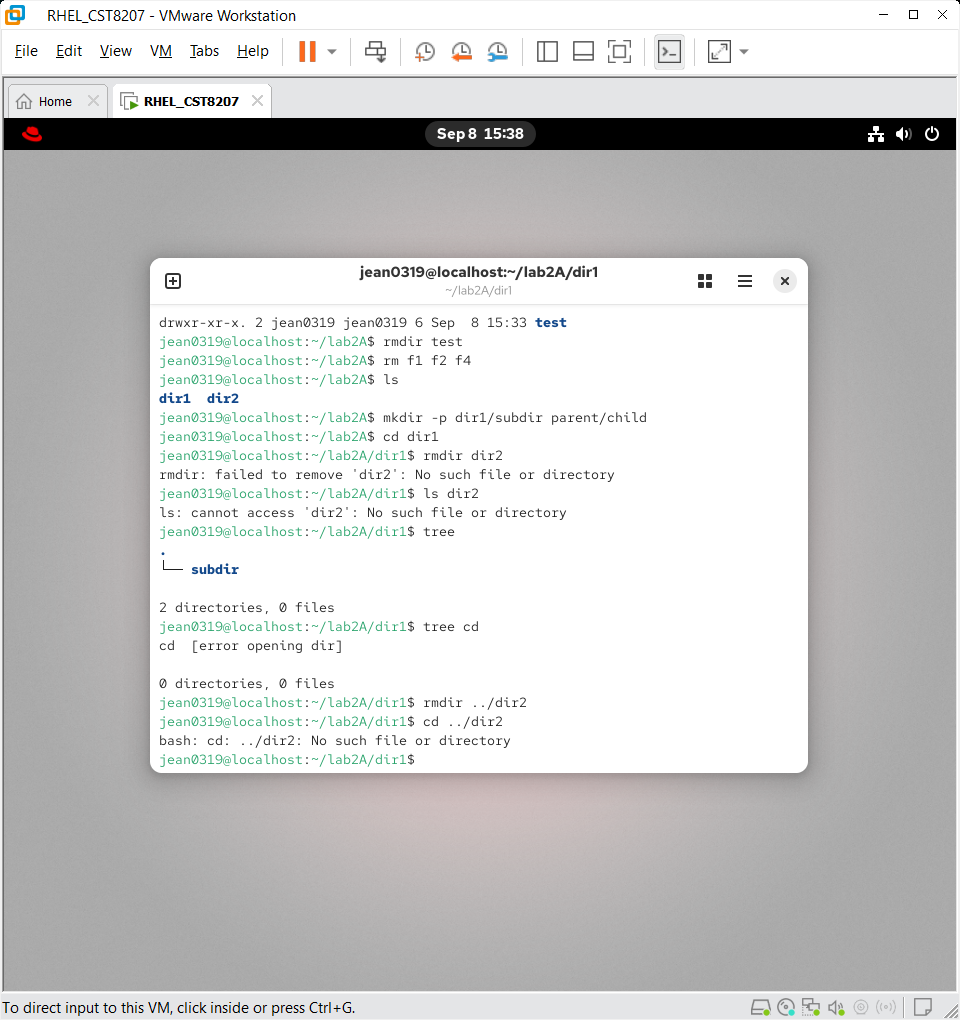
Ans: there is no dir2 within the dir1 directory

Ls

$ **rmdir ../dir2**

$ **cd ../dir2** # *this* ***fails*** *with an error message*

Screenshot here



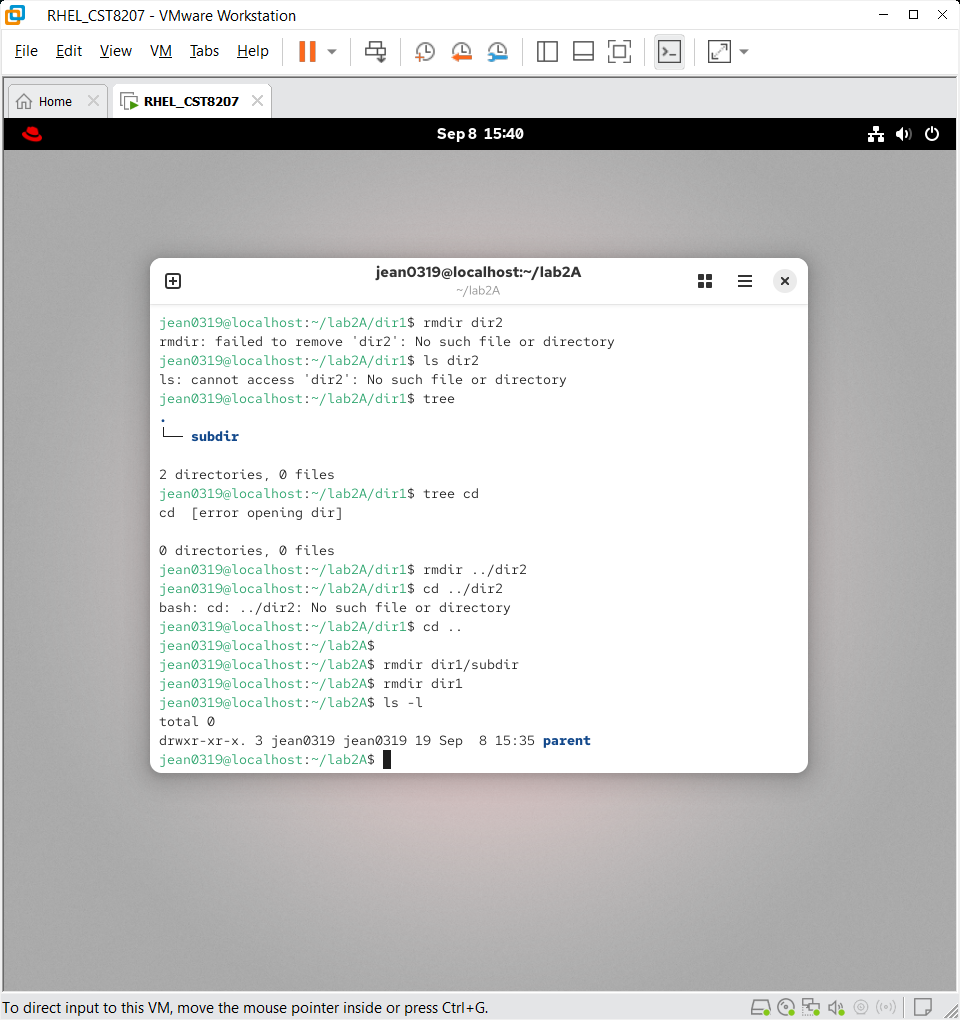
$ **cd .. #** *two dots means go up one directory level*

$ **rmdir dir1/subdir**

$ **rmdir dir1**

$ **ls -l**

Screenshot here



$ **rmdir parent/child parent**

m) **Why** doesn't the above command produce an **error message** about the non-empty directory **parent**?

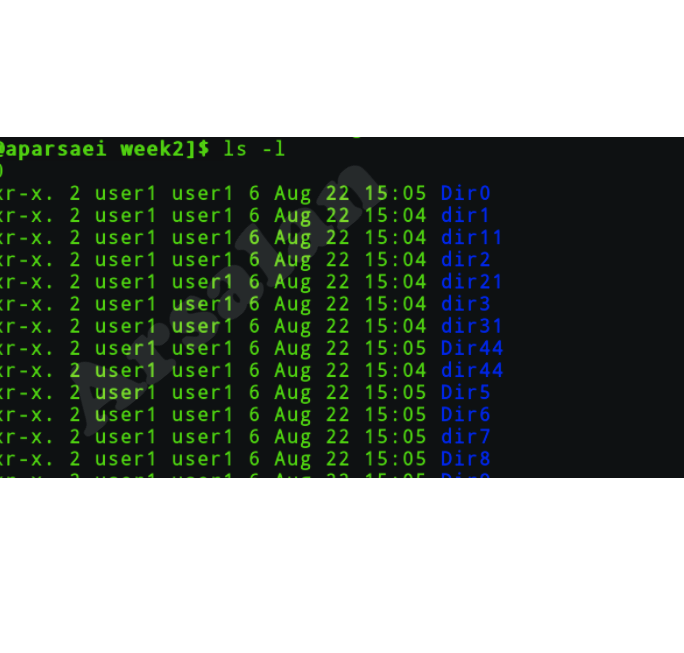
Ans: the command didn’t return an error message beacuse it removed the subdirectory child first before it removed the parent directory

Cont…..

Task 2:

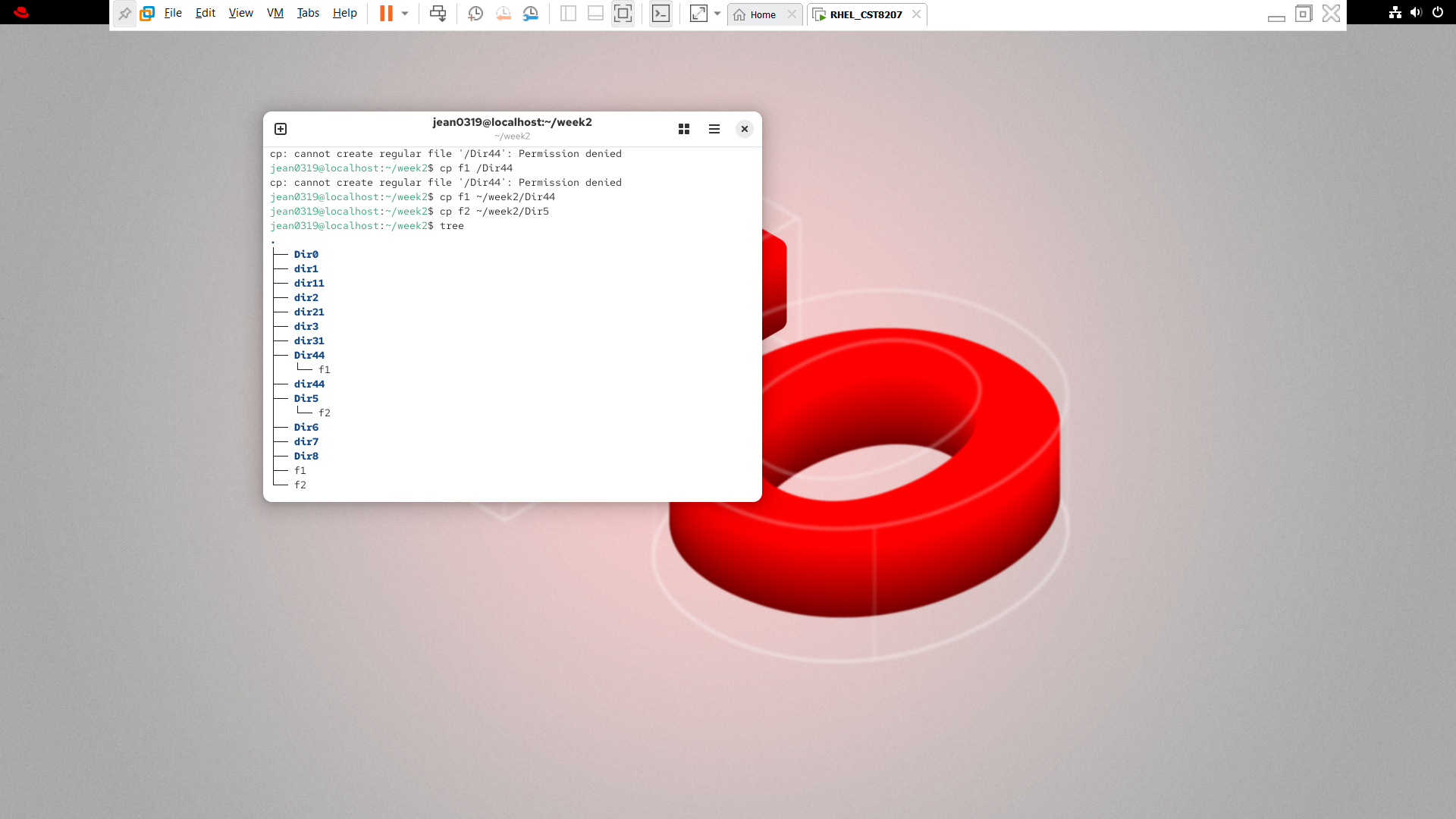
Create ~/week2 directory and make week2 directory as your working directory.

1. Now create following directory structure inside week2 **exactly** as you see in the following:



Now create two files called f1 f2 in /home/student/week2

* Copy f1 to Dir44 (Screenshot of command line)
* Copy f2 to Dir5 (Screenshot of command line and output from tree command)



* Rename f2 in Dir5 to f3 in Dir5
* Move f3 to Dir44

---------------------------------------------------------

This part should be demonstrated to your lab professor to get the mark:

Run **tree** command while you are in week2 directory and show the output to your Lab professor your professor makes ask questions about how you accomplished the task, so do it by yourself to gain skill.   
  
