Name of Student – Joydeep Saha
Roll Number – 2022EBCS321
Course Name -- Command Line Interfaces and Scripting
Cohort – 1

Graded Lab Assignment: Modules 1-4

Question 1 [5 points] Imagine you are a junior system administrator responsible for managing a Linux server. Your supervisor has given you various tasks related to common Linux commands. Your goal is to demonstrate your knowledge and proficiency in using these commands to maintain and troubleshoot the server effectively.

Answer ==

a) Change your user account's password to "SecurePwd123."
Bash –

```
passwd
```

Enter your current password and then type in your new password twice.

b) List all users currently logged into the server.
Bash –

```
who
```

This will display a list of all users currently logged into the server, along with their login time and terminal number.

c) Determine your own username and user ID (UID) on the server.
Bash –

```
whoami
id
```

The `whoami` command will display your username, while the `id` command will display your username, UID, and other information about your user account.

d) Create a text file named "important_notes.txt" and write the message "Confidential: For authorized personnel only" into the file.

Bash-

```
touch important_notes.txt
echo "Confidential: For authorized personnel only" >
important_notes.txt
```

The `touch` command creates an empty file, and the `echo` command writes the specified message to the file.

e) Access the manual page for the "ls" command to learn more about its options and usage.

Bash-
```
man ls
```

```
// Tried Installing man incase not found by this code -
```
sudo apt-get install man

This will open the man page for the `ls` command, which provides detailed information about the command's options and usage.

f) List the contents of a directory, including hidden files and directories, in long format.

Bash -
```
ls -la
```

The `-l` option tells `ls` to display files in long format, which includes information about the file's permissions, owner, group, size, and modification time. The `-a` option tells `ls` to display all files, including hidden files and directories.

g) Replace all occurrences of the letter 'a' with 'A' in a text file named "text.txt."

Bash –

```
sed -i 's/a/A/g' text.txt
```

The `sed` command is a stream editor that can be used to search for and replace text in files. The `-i` option tells `sed` to modify the file in place, rather than printing the modified text to the screen. The `s/a/A/g` command tells `sed` to replace all occurrences of the letter 'a' with the letter 'A' in the file.

h) Display the hostname of your server.

Bash -
`hostname`

This will display the hostname of your server.

i) Trace the network route from your local machine to the 'www.example.com' server (I used google for simple use)

Bash –

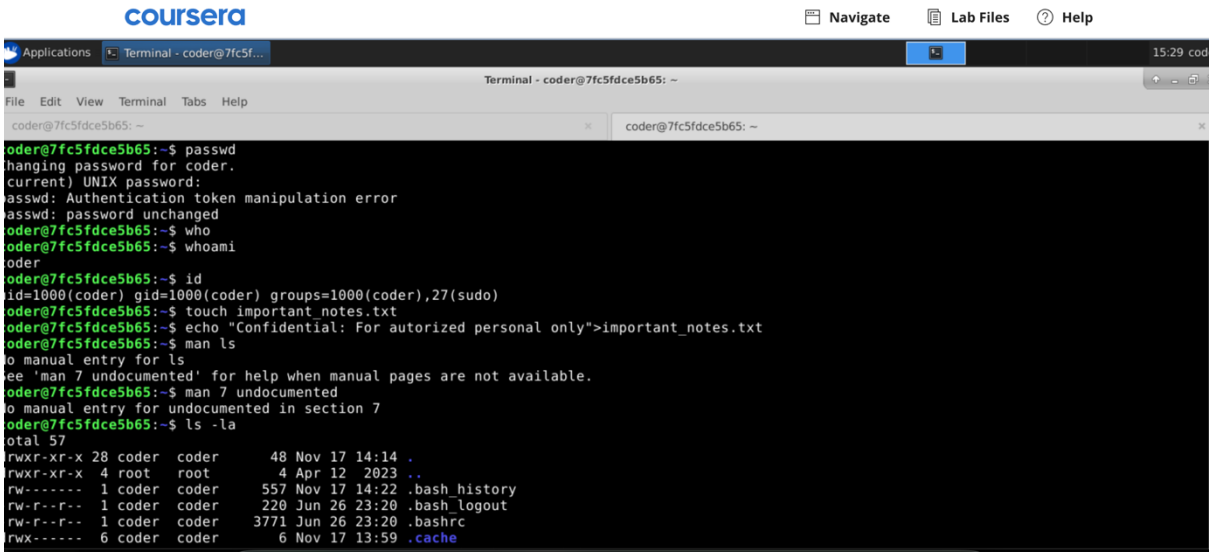
`traceroute www.google.com`

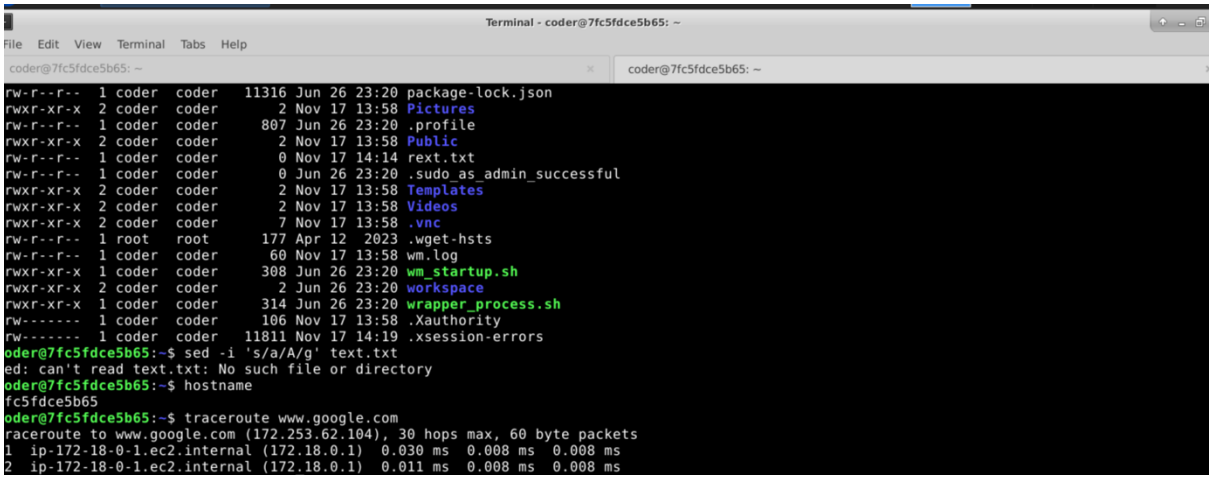
The `traceroute` command traces the network route from your local machine to the specified host. It displays a list of hops along the route, along with the time it takes to reach each hop.

j) Perform a basic port scan of a target server with the IP address "10.0.0.1."

Bash -
`nmap 10.0.0.1`

The `nmap` command is a port scanner that can be used to scan a target server for open ports. The `nmap 10.0.0.1` command will scan all TCP ports on the server with the IP address "10.0.0.1" and display a list of open ports and their associated services.





Question 2 –

You are working as a junior system administrator in a company that uses Linux servers for various tasks. Your supervisor has assigned you a series of tasks related to file permissions, file and directory operations, copying, and moving files and directories using Linux commands. Your goal is to successfully complete these tasks to ensure the proper management of the company's data and files.

Answer ==

1. Create a "reports" directory in your home folder with permissions set so that only the owner has read, write, and execute privileges while others have no access.
   - To navigate to the home directory and create the directory:
     Bash --

mkdir reports
chmod 700 reports

2. Inside the "reports" directory, generate a file named "monthly_report.txt" allowing the owner to read and write, the group to read, and others to have no access.
- Creating the file and setting permissions:
Bash --
cd reports
touch monthly_report.txt
chmod 640 monthly_report.txt

3. Copy the "monthly_report.txt" file to another directory while preserving its original permissions.
- Copying the file to a new location:

Bash --
cp -p ~/reports/monthly_report.txt ~/reports_temp

4. Move the "monthly_report.txt" file from a different directory to the "reports" directory ensuring it retains its initial permissions.
- Moving the file while retaining permissions:

Bash --
mv ~/reports_temp/monthly_report.txt ~/reports

5. List all files and directories within the "reports" directory, including hidden ones.**
- Displaying contents of the directory:

Bash --
ls -la ~/reports

6. Rename the "reports" directory to "quarterly_reports" without altering its location or content.
- Renaming the directory:

Bash --
mv ~/reports ~/quarterly_reports

7. Create a "backups" directory in your home directory and make "daily" and "weekly" subdirectories inside it.

- Creating the directory structure:

Bash –

mkdir ~/backups
mkdir ~/backups/daily
mkdir ~/backups/weekly

8. Generate an empty file named "important.txt" in the "daily" subdirectory within the "backups" directory.

- Creating an empty file:

Bash –

touch ~/backups/daily/important.txt

9. Move the "important.txt" file from the "daily" subdirectory to the "weekly" subdirectory while ensuring the file keeps its original name.

- Moving the file while retaining its name:

Bash --
mv ~/backups/daily/important.txt ~/backups/weekly

10. Display the absolute path of your present working directory.

- Showing the absolute path:

Bash --
pwd

coursera Navigate Lab Files Help

Applications Terminal - coder@5a73... 16:06 coc

Terminal - coder@5a7318d9ef2a: ~/reports

File Edit View Terminal Tabs Help

```
coder@5a7318d9ef2a:~$ mkdir report
coder@5a7318d9ef2a:~$ mkdir reports
coder@5a7318d9ef2a:~$ chmod 700 reports
coder@5a7318d9ef2a:~$ cd reports
coder@5a7318d9ef2a:~/reports$ touch monthly_report.txt
coder@5a7318d9ef2a:~/reports$ chmod 640 monthly_report.txt
coder@5a7318d9ef2a:~/reports$ cp -p ~reports/monthly_report.txt~/reports_temp
cp: missing destination file operand after '~reports/monthly_report.txt~/reports_temp'
Try 'cp --help' for more information.
coder@5a7318d9ef2a:~/reports$ cp monthly_report.txt /path/to/destination/directory
cp: cannot create regular file '/path/to/destination/directory': No such file or directory
coder@5a7318d9ef2a:~/reports$ mv ~reports_temp/monthly_reports.txt~/reports
mv: missing destination file operand after '~reports_temp/monthly_reports.txt~/reports'
Try 'mv --help' for more information.
coder@5a7318d9ef2a:~/reports$ cd ~reports
bash: cd: ~reports: No such file or directory
coder@5a7318d9ef2a:~/reports$ ls -la
total 4
drwx------  2 coder coder  3 Nov 17 15:48 .
drwxr-xr-x 30 coder coder 45 Nov 17 15:48 ..
-rw-r-----  1 coder coder  0 Nov 17 15:48 monthly_report.txt
coder@5a7318d9ef2a:~/reports$ ls -la reports
ls: cannot access 'reports': No such file or directory
coder@5a7318d9ef2a:~/reports$ mv reports quarterly_reports
mv: cannot stat 'reports': No such file or directory
coder@5a7318d9ef2a:~/reports$ mkdir ~backups/daily/
mkdir: cannot create directory ‘~backups/daily/’: No such file or directory
```

```
Try 'mv --help' for more information.
coder@5a7318d9ef2a:~/reports$ cd ~reports
bash: cd: ~reports: No such file or directory
coder@5a7318d9ef2a:~/reports$ ls -la
total 4
drwx------  2 coder coder  3 Nov 17 15:48 .
drwxr-xr-x 30 coder coder 45 Nov 17 15:48 ..
-rw-r-----  1 coder coder  0 Nov 17 15:48 monthly_report.txt
coder@5a7318d9ef2a:~/reports$ ls -la reports
ls: cannot access 'reports': No such file or directory
coder@5a7318d9ef2a:~/reports$ mv reports quarterly_reports
mv: cannot stat 'reports': No such file or directory
coder@5a7318d9ef2a:~/reports$ mkdir ~backups/daily/
mkdir: cannot create directory ‘~backups/daily/’: No such file or directory
coder@5a7318d9ef2a:~/reports$ mkdir ~backups/daily
mkdir: cannot create directory ‘~backups/daily’: No such file or directory
coder@5a7318d9ef2a:~/reports$ mkdir ~/backups/daily
mkdir: cannot create directory ‘/home/coder/backups/daily’: No such file or directory
coder@5a7318d9ef2a:~/reports$ mkdir ~/backups
coder@5a7318d9ef2a:~/reports$ mkdir ~/backups/daily
coder@5a7318d9ef2a:~/reports$ mkdir ~/backups/weekly
coder@5a7318d9ef2a:~/reports$ touch ~/backup/daily/important.txt
touch: cannot touch '/home/coder/backup/daily/important.txt': No such file or directory
coder@5a7318d9ef2a:~/reports$ touch ~/backups/daily/important.txt
coder@5a7318d9ef2a:~/reports$ mv ~/backups/daily/important.txt ~/backups/weekly
coder@5a7318d9ef2a:~/reports$
```

coursera Navigate Lab Files Help

Applications Terminal - coder@5a73... 16:06 coder

Terminal - coder@5a7318d9ef2a: ~/reports

File Edit View Terminal Tabs Help

```
bash: cd: ~reports: No such file or directory
coder@5a7318d9ef2a:~/reports$ ls -la
total 4
drwx------  2 coder coder  3 Nov 17 15:48 .
drwxr-xr-x 30 coder coder 45 Nov 17 15:48 ..
-rw-r-----  1 coder coder  0 Nov 17 15:48 monthly_report.txt
coder@5a7318d9ef2a:~/reports$ ls -la reports
ls: cannot access 'reports': No such file or directory
coder@5a7318d9ef2a:~/reports$ mv reports quarterly_reports
mv: cannot stat 'reports': No such file or directory
coder@5a7318d9ef2a:~/reports$ mkdir ~backups/daily/
mkdir: cannot create directory ‘~backups/daily/’: No such file or directory
coder@5a7318d9ef2a:~/reports$ mkdir ~backups/daily
mkdir: cannot create directory ‘~backups/daily’: No such file or directory
coder@5a7318d9ef2a:~/reports$ mkdir ~/backups/daily
mkdir: cannot create directory ‘/home/coder/backups/daily’: No such file or directory
coder@5a7318d9ef2a:~/reports$ mkdir ~/backups
coder@5a7318d9ef2a:~/reports$ mkdir ~/backups/daily
coder@5a7318d9ef2a:~/reports$ mkdir ~/backups/weekly
coder@5a7318d9ef2a:~/reports$ touch ~/backup/daily/important.txt
touch: cannot touch '/home/coder/backup/daily/important.txt': No such file or directory
coder@5a7318d9ef2a:~/reports$ touch ~/backups/daily/important.txt
coder@5a7318d9ef2a:~/reports$ mv ~/backups/daily/important.txt ~/backups/weekly
coder@5a7318d9ef2a:~/reports$ pwd
/home/coder/reports
coder@5a7318d9ef2a:~/reports$
```

Question 3 - You have been hired as a junior system administrator for a company that manages a Linux server hosting critical data and applications. Your supervisor has assigned you a series of tasks related to understanding and managing links, inodes, and superblocks using Linux commands. Your goal is to successfully complete these tasks to ensure the proper maintenance of the server.

Answer –

1. Create a hard link from the "important_data.txt" file in the root directory to your home directory and confirm the linkage by checking the inode.
  - Command:
    Bash--
    ln ~/important_data.txt ~/hl_data.txt
    ls -i ~/important_data.txt ~/hl_data.txt

2. In your home directory, generate a symbolic link to the "important_data.txt" file located in the root directory. Verify permissions and the symbolic link's inode number.
  - Command:
    Bash --
    ln -s ~/important_data.txt ~/sl_data.txt
    ls -li ~/sl_data.txt

3. Display all hard links to the "important_data.txt" file, including their full paths.**
  - Command:
    Bash --
    find / -samefile /important_data.txt 2>/dev/null

4. Retrieve detailed information about the "important_data.txt" file, such as its size, owner, group, and permissions.
  - Command:
    Bash --
    ls -l ~/important_data.txt

5. Display the disk space usage of the filesystem where the "important_data.txt" file is located, showing available space, used space, and filesystem size.

  - Command: (This would depend on the filesystem and its corresponding command, e.g., `df -h /path/to/filesystem`)

6. Extract and present details about the superblock of the filesystem where "important_data.txt" resides, including block size and filesystem features.

- Command:
  Bash -
  dumpe2fs /dev/sda2

7. Create a "backup" subdirectory in your home directory, move "important_data.txt" into it from the root directory, and check the file's inode post-relocation.
- Command:
  Bash --
  mkdir ~/backup
  mv ~/important_data.txt ~/backup/
  ls -i ~/backup/important_data.txt

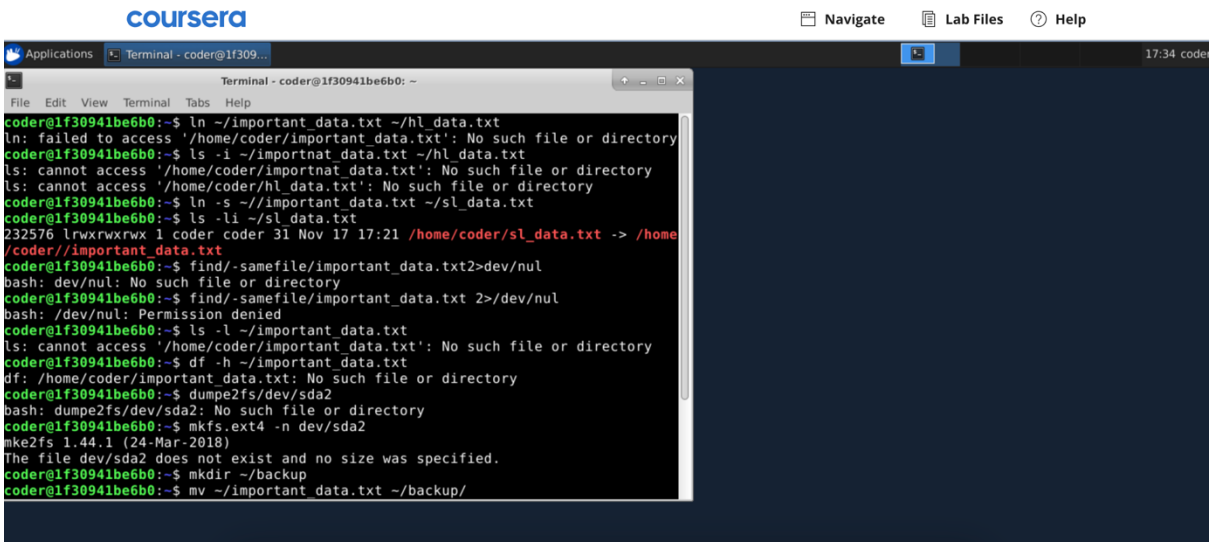
8. Generate a new symbolic link in your home directory pointing to the relocated "important_data.txt" file within the "backup" directory. Confirm the validity of the link.
- Command:
  bash --
  ln -s ~/backup/important_data.txt ~/sl_backup_data.txt
  ls -li ~/sl_backup_data.txt

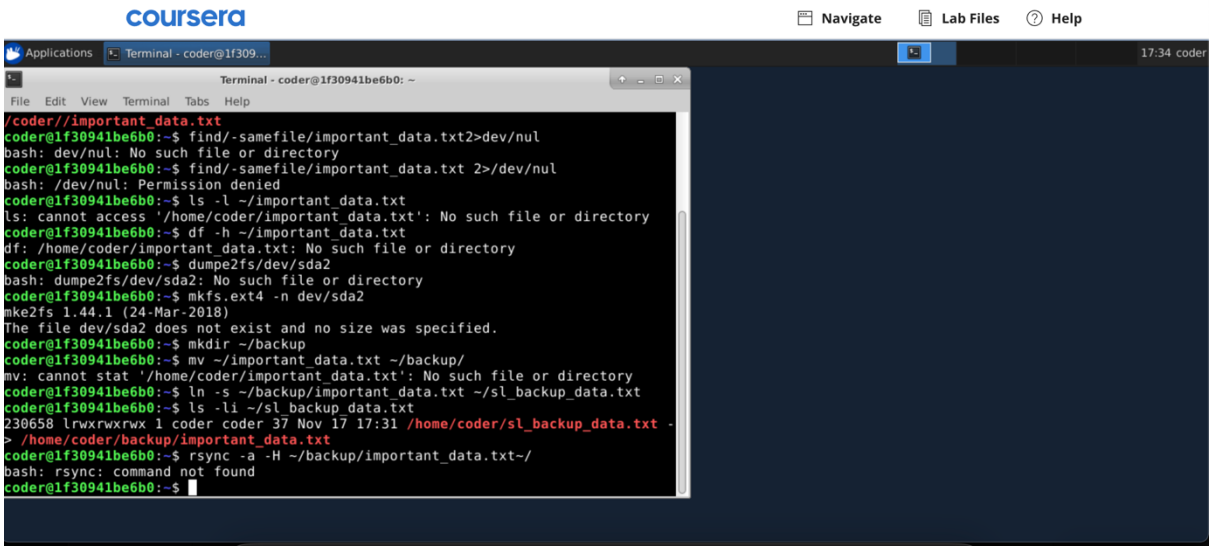
9. Explain why hard links cannot span across different filesystems, referencing inodes and superblocks.
- **Explanation**:
  Hard links rely on the inode structure within a single filesystem. Each inode uniquely identifies a file within that filesystem. As different filesystems have their independent inode structures and superblocks, a hard link cannot cross filesystems since the inode numbers aren’t synchronized between them.

10. Outline a backup strategy for "important_data.txt" ensuring link integrity and efficient disk usage.
- Strategy: Utilize a backup tool compatible with hard and symbolic links, like rsync with the `-H` option. For instance:

  Bash --
  rsync -a -H ~/backup/important_data.txt ~/

  This command preserves hard links and properly handles symbolic links, ensuring link integrity and efficient disk usage during backup.


coursera
Navigate
Lab Files
Help
Applications
Terminal - coder@1f309...
17:34 code
Terminal - coder@1f30941be6b0: ~
File Edit View Terminal Tabs Help
coder@1f30941be6b0:~$ ln ~/important_data.txt ~/hl_data.txt
ln: failed to access '/home/coder/important_data.txt': No such file or directory
coder@1f30941be6b0:~$ ls -i ~/importnat_data.txt ~/hl_data.txt
ls: cannot access '/home/coder/importnat_data.txt': No such file or directory
ls: cannot access '/home/coder/hl_data.txt': No such file or directory
coder@1f30941be6b0:~$ ln -s ~//important_data.txt ~/sl_data.txt
coder@1f30941be6b0:~$ ls -li ~/sl_data.txt
232576 lrwxrwxrwx 1 coder coder 31 Nov 17 17:21 /home/coder/sl_data.txt -> /home
/coder//important_data.txt
coder@1f30941be6b0:~$ find/-samefile/important_data.txt2>dev/nul
bash: dev/nul: No such file or directory
coder@1f30941be6b0:~$ find/-samefile/important_data.txt 2>/dev/nul
bash: /dev/nul: Permission denied
coder@1f30941be6b0:~$ ls -l ~/important_data.txt
ls: cannot access '/home/coder/important_data.txt': No such file or directory
coder@1f30941be6b0:~$ df -h ~/important_data.txt
df: /home/coder/important_data.txt: No such file or directory
coder@1f30941be6b0:~$ dumpe2fs/dev/sda2
bash: dumpe2fs/dev/sda2: No such file or directory
coder@1f30941be6b0:~$ mkfs.ext4 -n dev/sda2
mke2fs 1.44.1 (24-Mar-2018)
The file dev/sda2 does not exist and no size was specified.
coder@1f30941be6b0:~$ mkdir ~/backup
coder@1f30941be6b0:~$ mv ~/important_data.txt ~/backup/


coursera
Navigate
Lab Files
Help
Applications
Terminal - coder@1f309...
17:34 coder
Terminal - coder@1f30941be6b0: ~
File Edit View Terminal Tabs Help
/coder//important_data.txt
coder@1f30941be6b0:~$ find/-samefile/important_data.txt2>dev/nul
bash: dev/nul: No such file or directory
coder@1f30941be6b0:~$ find/-samefile/important_data.txt 2>/dev/nul
bash: /dev/nul: Permission denied
coder@1f30941be6b0:~$ ls -l ~/important_data.txt
ls: cannot access '/home/coder/important_data.txt': No such file or directory
coder@1f30941be6b0:~$ df -h ~/important_data.txt
df: /home/coder/important_data.txt: No such file or directory
coder@1f30941be6b0:~$ dumpe2fs/dev/sda2
bash: dumpe2fs/dev/sda2: No such file or directory
coder@1f30941be6b0:~$ mkfs.ext4 -n dev/sda2
mke2fs 1.44.1 (24-Mar-2018)
The file dev/sda2 does not exist and no size was specified.
coder@1f30941be6b0:~$ mkdir ~/backup
coder@1f30941be6b0:~$ mv ~/important_data.txt ~/backup/
mv: cannot stat '/home/coder/important_data.txt': No such file or directory
coder@1f30941be6b0:~$ ln -s ~/backup/important_data.txt ~/sl_backup_data.txt
coder@1f30941be6b0:~$ ls -li ~/sl_backup_data.txt
230658 lrwxrwxrwx 1 coder coder 37 Nov 17 17:31 /home/coder/sl_backup_data.txt -
> /home/coder/backup/important_data.txt
coder@1f30941be6b0:~$ rsync -a -H ~/backup/important_data.txt~/
bash: rsync: command not found
coder@1f30941be6b0:~$

Question 4 -You have been assigned the role of a system administrator for a Linux server responsible for managing multiple hard disks. Your supervisor has tasked you with various responsibilities related to inspecting hard disks, partitions, and sectors using various Linux commands. Your objective is to successfully complete these tasks to ensure the proper maintenance and monitoring of the server's storage components.

Answer ==

1. List all available block devices on the system, showcasing device names, sizes, and mount points where applicable.
  - Command:
    Bash-
    lsblk

2. Access detailed information about a specific hard disk, including its model, size, and SMART status using the suitable command.
  - Command:
    Bash-
    smartctl -a /dev/sda

3. Display details about the partitions on a specific hard disk, highlighting their sizes and filesystem types, covering both mounted and unmounted partitions.
  - Command:
    Bash -
    blkid /dev/sda2

4. Check available disk space on the system and present it in a human-readable format, illustrating total size, used space, and available space for each mounted filesystem.
  - Command:
    Bash-
    df -H

5. Examine the partition table of a specific hard disk, detailing partition sizes, types, and start sectors.
  - Commands:
    Bash-
    sudo fdisk /dev/sda
    sudo parted /dev/sda print

6. **Inspect sector-level information of a specific partition on a hard disk, retrieving data like sector size, total sectors, and used sectors.**

- Command:
  Bash-
  sudo fdisk -l /dev/sda

7. Check the SMART status of a particular hard disk, interpreting results to identify potential disk health issues.
- Command:
  Bash-
  smartctl -H /dev/sda

8. Create a new partition on a specific hard disk, specifying partition type, size, and filesystem, explaining the purpose of the new partition.
- Command:
  Bash -
  mkpart primary ext4 1MB 1855MB

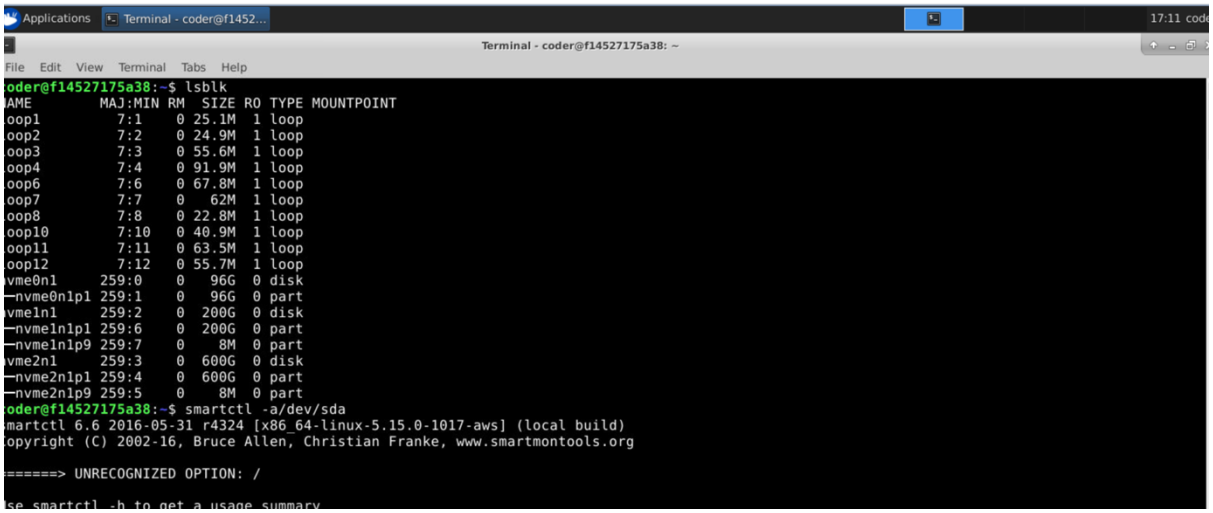
- **Explanation**: The command establishes an ext4 filesystem, initializing a partition starting at 1MB and ending at 1855MB.

9. Perform data recovery from a damaged sector on a partition, outlining steps involved in the data recovery process.

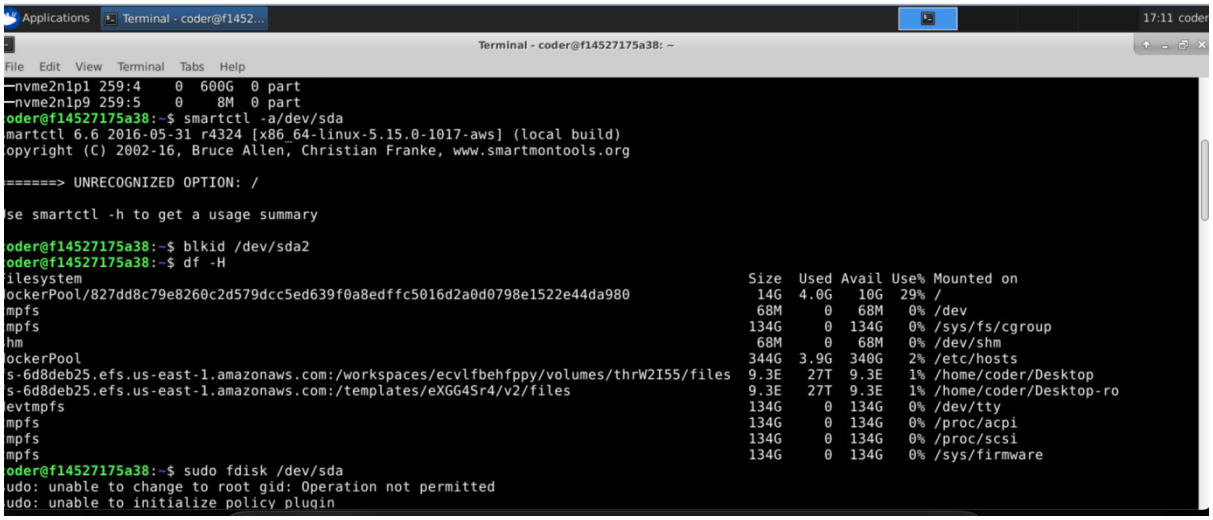
- **My Response**: If a sector is damaged, data recovery can be challenging. Typically, utilizing backup copies of the data, if available, is the first step. Tools like **ddrescue** or specialized recovery software might be used. It's crucial to work on a copy of the damaged partition to avoid further data loss.

10. Determine disk usage of a specific directory and its subdirectories, showcasing space consumed by each subdirectory.
- Command:
  Bash --
  du -h --max-depth=2 ~/

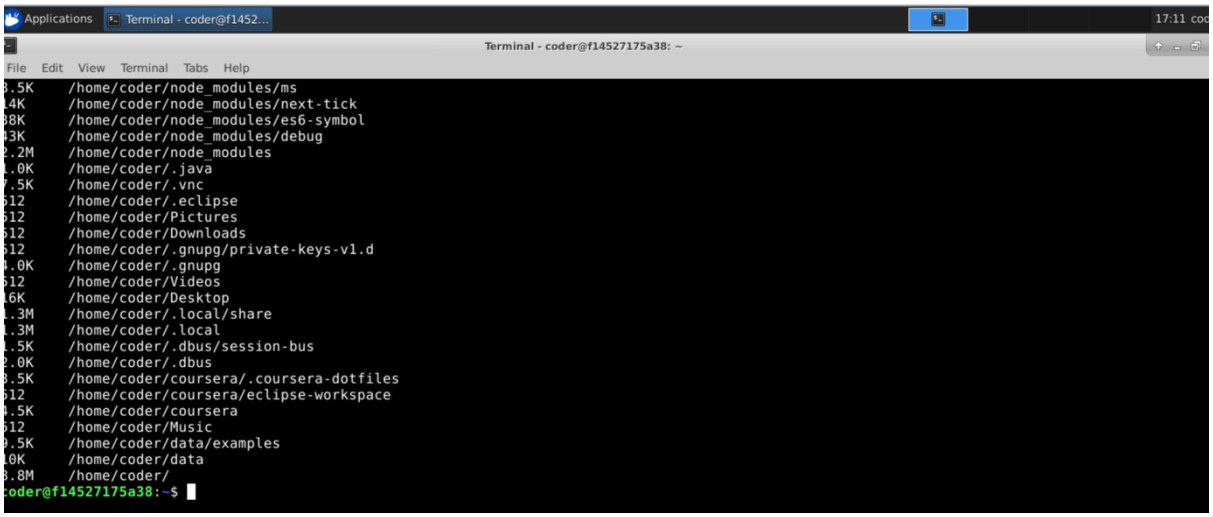
coursera Navigate Lab Files Help



coursera Navigate Lab Files Help



coursera Navigate Lab Files Help



coursera

Navigate Lab Files Help