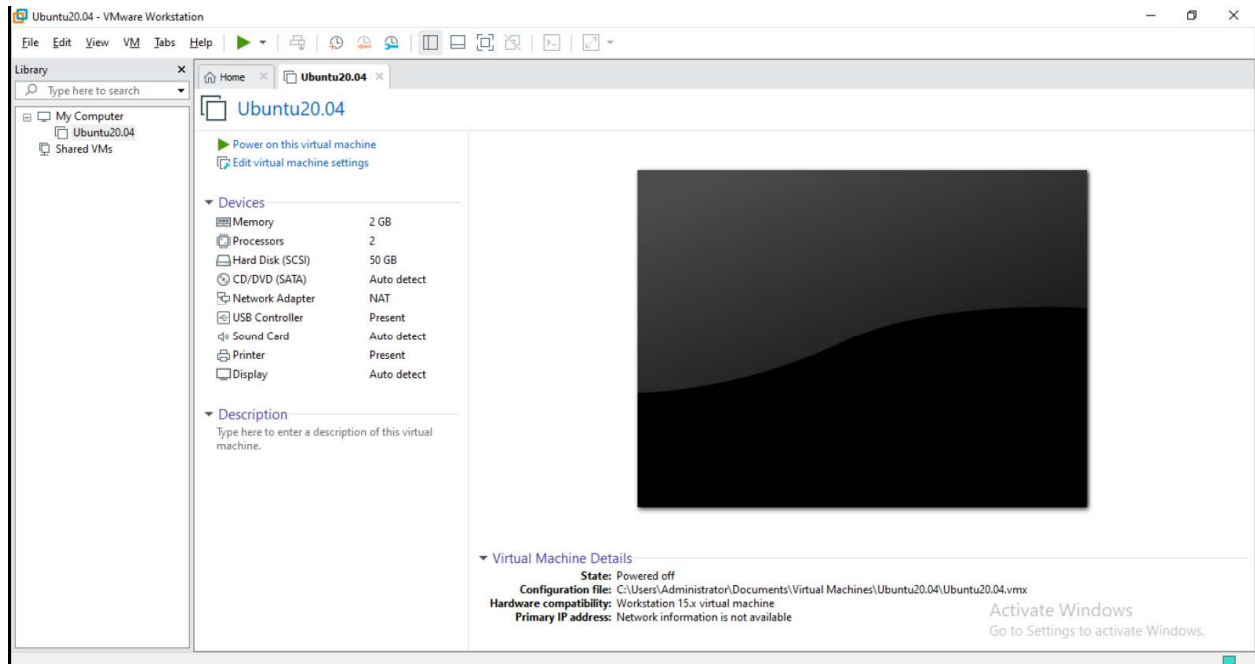


Assignments:

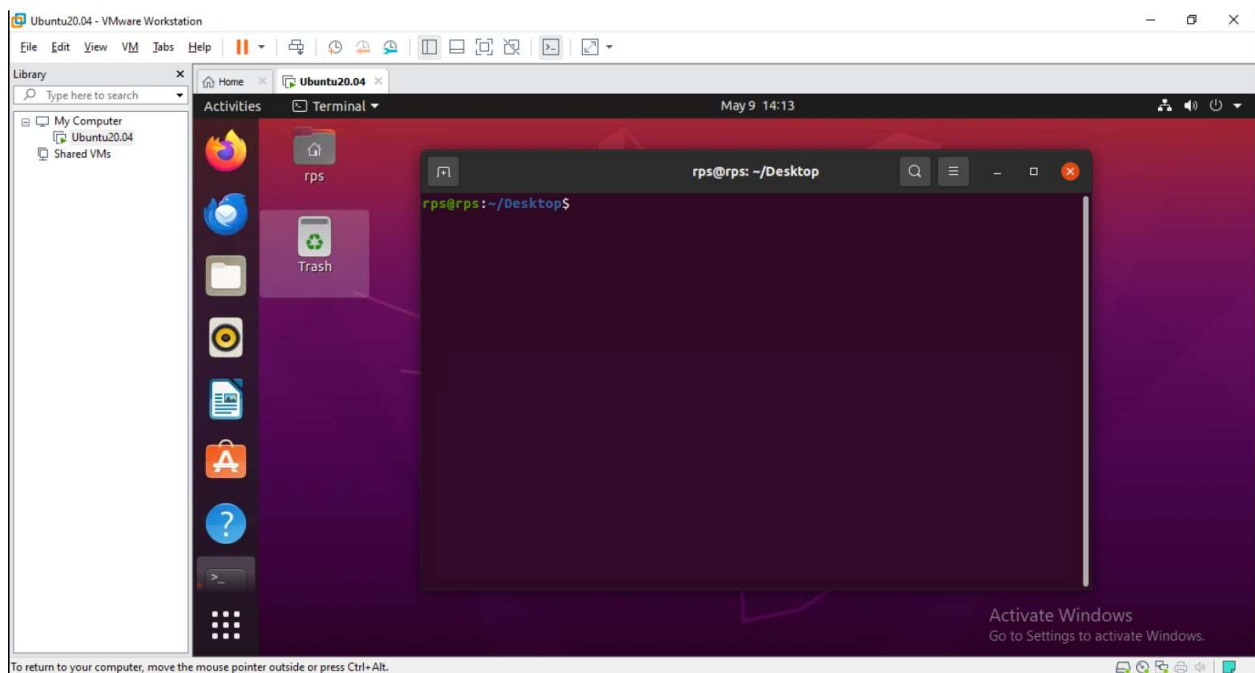
1. Access vm successfully and open a terminal.

Accessed VM and opened terminal successfully. Image is attached below,

VM Access

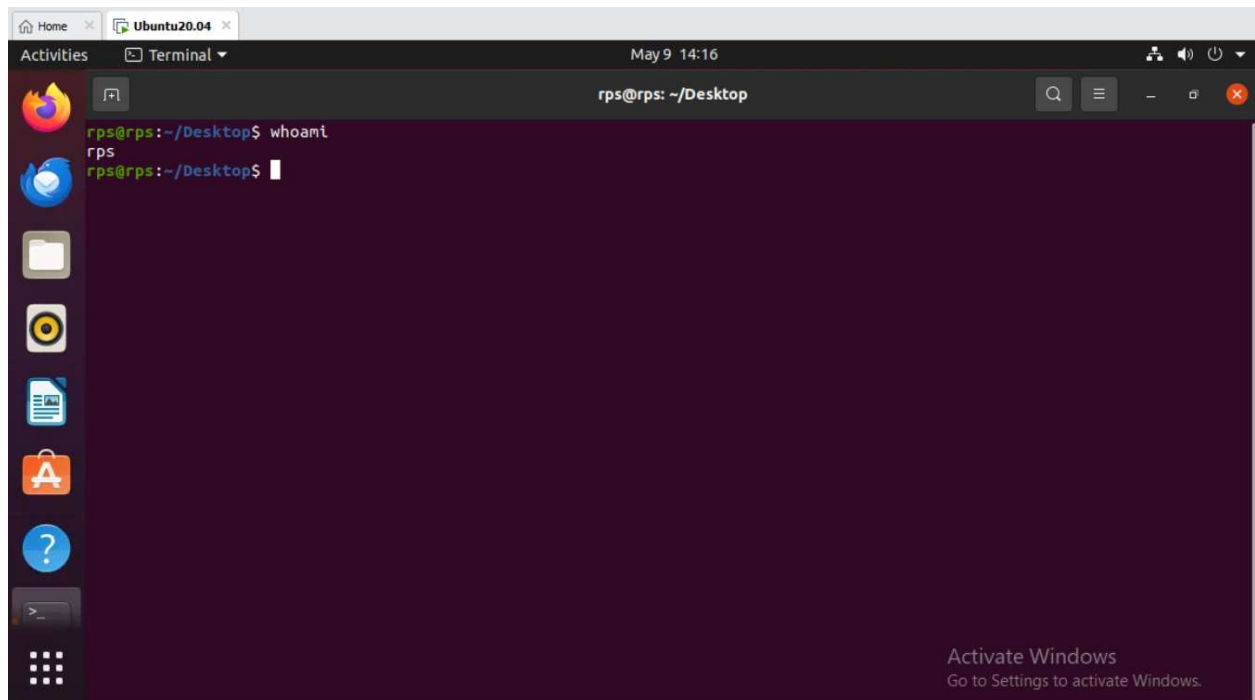


Terminal open:



2. find the current user.

Current user find:

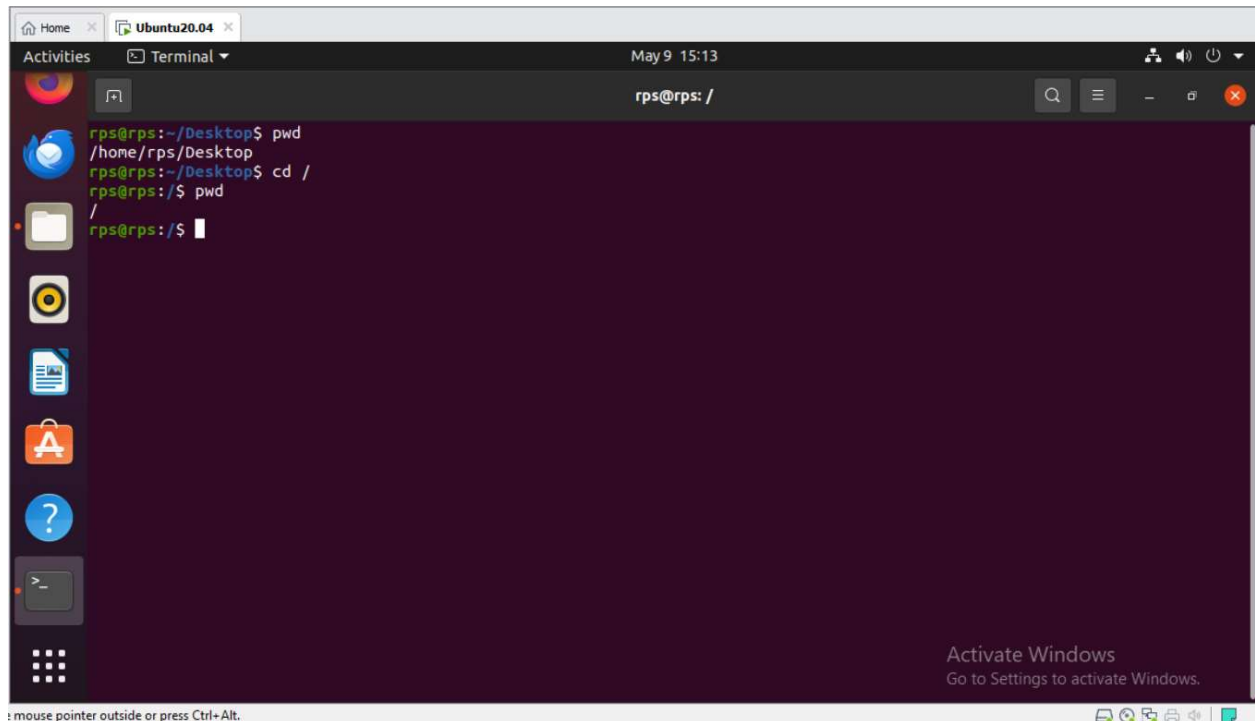


A screenshot of an Ubuntu 20.04 terminal window. The window title is 'Ubuntu20.04'. The terminal shows the user 'rps' at the prompt 'rps@rps: ~/Desktop'. The command 'whoami' has been entered, and the output is 'rps'. The terminal background is dark purple. The desktop environment is visible on the left with various application icons. The system clock at the top right shows 'May 9 14:16'. An 'Activate Windows' watermark is visible in the bottom right corner.

```
rps@rps:~/Desktop$ whoami
rps
rps@rps:~/Desktop$
```

3. Navigate to root directory from home directory using terminal.

Navigation to root directory using terminal:

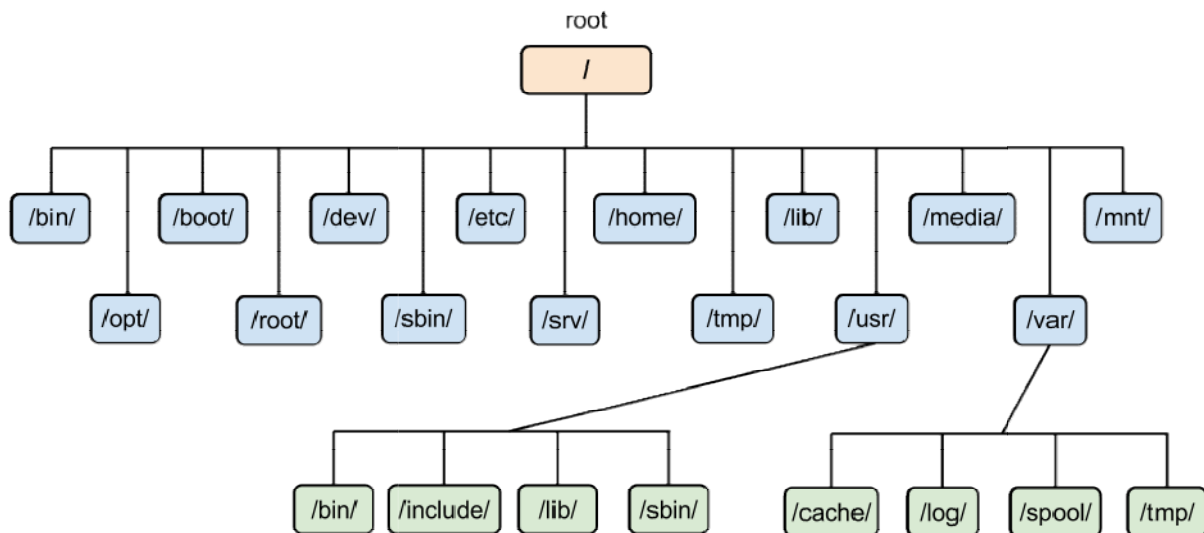


A screenshot of an Ubuntu 20.04 terminal window. The window title is 'Ubuntu20.04'. The terminal shows the user 'rps' at the prompt 'rps@rps: /'. The commands entered are 'pwd' (output: '/home/rps/Desktop'), 'cd /', and 'pwd' (output: '/'). The terminal background is dark purple. The desktop environment is visible on the left with various application icons. The system clock at the top right shows 'May 9 15:13'. An 'Activate Windows' watermark is visible in the bottom right corner.

```
rps@rps:~/Desktop$ pwd
/home/rps/Desktop
rps@rps:~/Desktop$ cd /
rps@rps:/$ pwd
/
rps@rps:/$
```

4. Explain File system hierarchy in Linux OS.

File system hierarchy in Linux OS:



The File System Hierarchy in Linux organizes the storage of files and directories in a structured manner. It's crucial for understanding how data is stored, managed, and accessed within the operating system. Here's a comprehensive explanation that should help you score well on your assignment:

The Linux File System Hierarchy Standard (FHS) defines the directory structure and organization of files in a Linux operating system. It establishes a consistent layout across different distributions, ensuring compatibility and ease of navigation for users and administrators.

At the root of the file system is the "/" directory, which serves as the starting point for all file paths. Here's a breakdown of the key directories and their purposes:

1. /bin:

- Contains essential user command binaries necessary for basic system functionality.
- Commands stored here are fundamental to system operation, such as ls, cp, and mv.

2. /boot:

- Holds files required for booting the Linux kernel, including boot loader configurations and kernel images.
- Critical for the system's initial startup process and handling of kernel loading.

3. /dev:

- Contains device files representing hardware devices connected to the system.
- Allows user applications and the kernel to communicate with hardware devices through special file interfaces.

4. /etc:

- Stores system-wide configuration files for various applications and services.
- Includes network configuration files, user authentication files, and other system settings.

5. /home:

- Provides individual user directories for storing personal files and settings.
- Each user has a unique subdirectory within /home corresponding to their username.

6. /lib and /lib64:

- Store shared libraries required by programs and applications at runtime.
- /lib contains 32-bit libraries, while /lib64 contains 64-bit libraries on compatible systems.

7. /media and /mnt:

- Serve as mount points for removable media and external file systems.
- When devices like USB drives or network shares are mounted, they appear under these directories.

8. /opt:

- Reserved for optional or third-party software installations.
- Provides a standardized location for installing additional software packages not included in the base system.

9. /proc and /sys:

- Virtual file systems that expose kernel and system information.
- /proc contains information about running processes and system resources, while /sys exposes kernel attributes and settings.

10. /root:

- Home directory for the root user (system administrator).
- Contains administrative files and configurations specific to the root user.

11. /sbin:

- Stores system binaries primarily used for system administration tasks.
- Contains essential executables necessary for system maintenance and repair.

12. /srv:

- Used for storing data related to services provided by the system.
- Often used for hosting website data, FTP servers, or other service-related files.

13. /tmp:

- Temporary directory for storing transient files created by applications and users.
- Files stored here are typically deleted upon system reboot or periodic cleanup.

14. /usr:

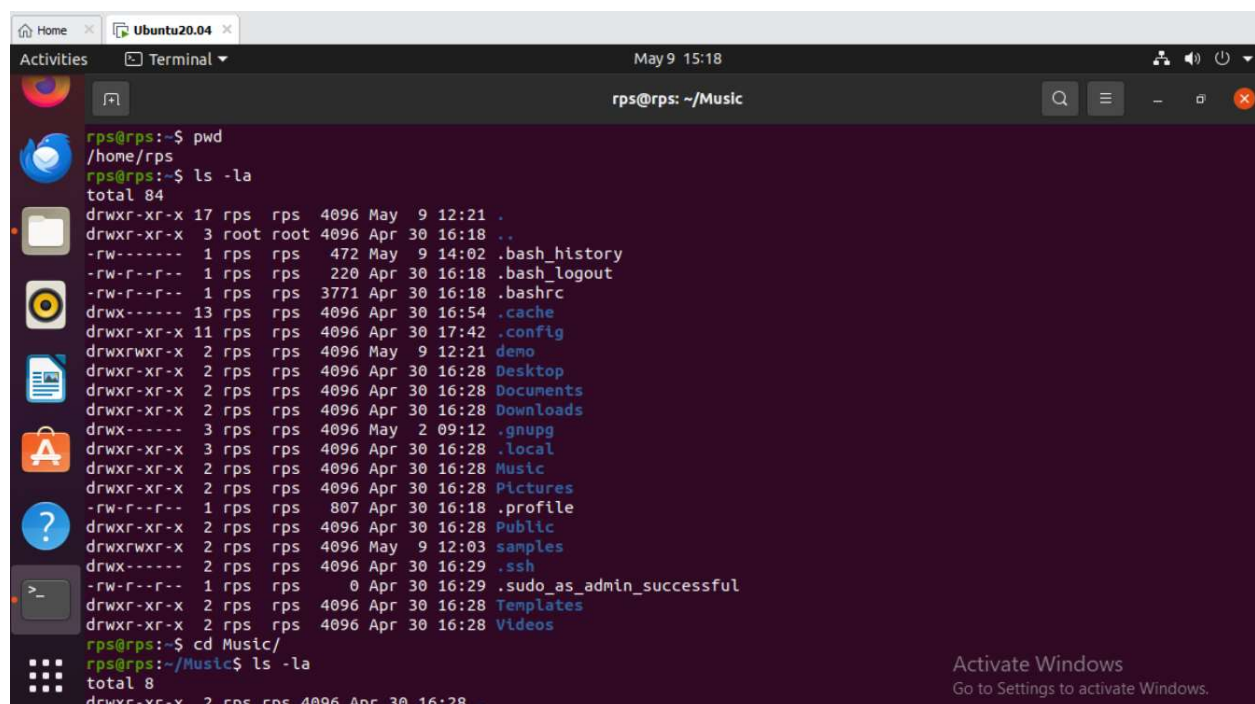
- Contains user-related files, including applications, libraries, and documentation.
- Organized into subdirectories like /usr/bin for user binaries and /usr/share for shared data.

15. /var:

- Holds variable data files that may change in size during system operation.
- Includes log files, spool directories, temporary files, and other dynamic data.

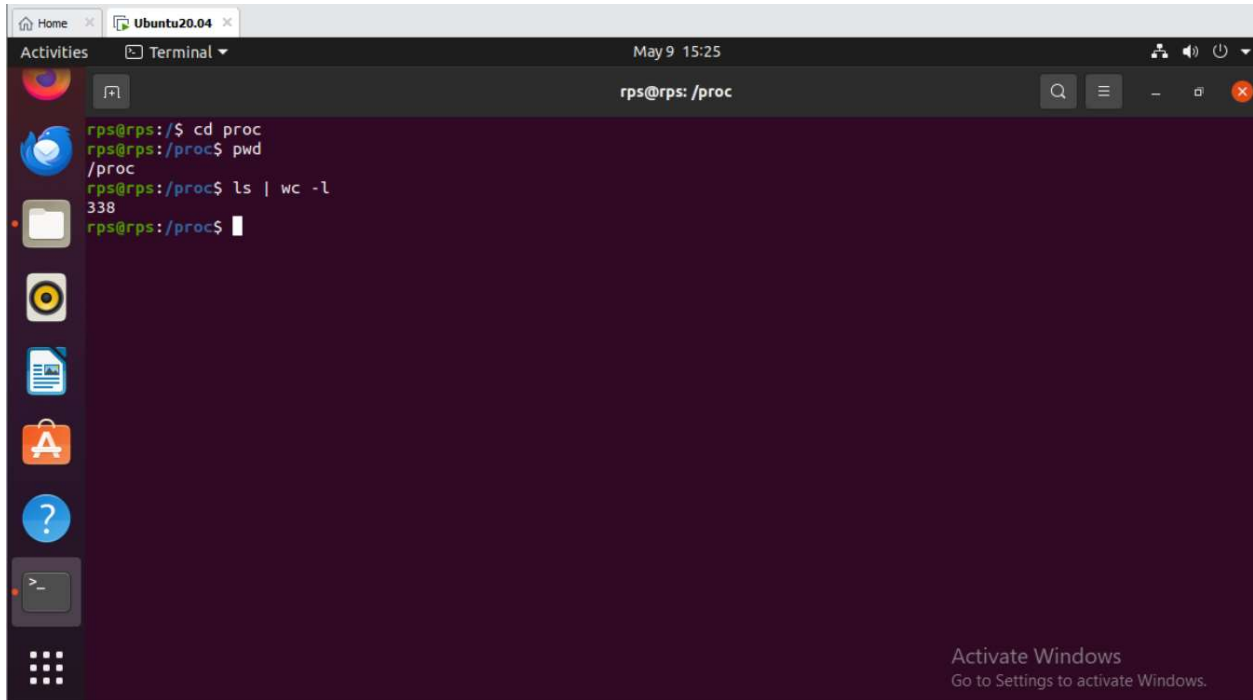
Understanding the Linux File System Hierarchy is essential for system administrators, developers, and users alike. It provides a standardized way to organize and access files and directories, facilitating efficient management and troubleshooting of Linux-based systems.

5. List all the contents of a directory.



```
rps@rps:~$ pwd
/home/rps
rps@rps:~$ ls -la
total 84
drwxr-xr-x 17 rps rps 4096 May  9 12:21 .
drwxr-xr-x  3 root root 4096 Apr 30 16:18 ..
-rw-r--r--  1 rps rps  472 May  9 14:02 .bash_history
-rw-r--r--  1 rps rps  220 Apr 30 16:18 .bash_logout
-rw-r--r--  1 rps rps 3771 Apr 30 16:18 .bashrc
drwx----- 13 rps rps 4096 Apr 30 16:54 .cache
drwxr-xr-x 11 rps rps 4096 Apr 30 17:42 .config
drwxrwxr-x  2 rps rps 4096 May  9 12:21 demo
drwxr-xr-x  2 rps rps 4096 Apr 30 16:28 Desktop
drwxr-xr-x  2 rps rps 4096 Apr 30 16:28 Documents
drwxr-xr-x  2 rps rps 4096 Apr 30 16:28 Downloads
drwx-----  3 rps rps 4096 May  2 09:12 .gnupg
drwxr-xr-x  3 rps rps 4096 Apr 30 16:28 .local
drwxr-xr-x  2 rps rps 4096 Apr 30 16:28 Music
drwxr-xr-x  2 rps rps 4096 Apr 30 16:28 Pictures
-rw-r--r--  1 rps rps  807 Apr 30 16:18 .profile
drwxr-xr-x  2 rps rps 4096 Apr 30 16:28 Public
drwxrwxr-x  2 rps rps 4096 May  9 12:03 samples
drwx-----  2 rps rps 4096 Apr 30 16:29 .ssh
-rw-r--r--  1 rps rps    0 Apr 30 16:29 .sudo_as_admin_successful
drwxr-xr-x  2 rps rps 4096 Apr 30 16:28 Templates
drwxr-xr-x  2 rps rps 4096 Apr 30 16:28 Videos
rps@rps:~$ cd Music/
rps@rps:~/Music$ ls -la
total 8
drwxr-xr-x  2 rps rps 4096 Apr 30 16:28 .
```

6. Perform a directory count.



```
rps@rps:/$ cd /proc
rps@rps:/proc$ pwd
/proc
rps@rps:/proc$ ls | wc -l
338
rps@rps:/proc$
```

The screenshot shows a terminal window titled 'Terminal' with the Ubuntu 20.04 logo. The user 'rps' is logged in. The terminal shows the user navigating to the '/proc' directory using 'cd /proc', confirming the current directory with 'pwd', and then running 'ls | wc -l' to count the number of files and directories. The output is '338'. The terminal window has a dark purple background and a sidebar with application icons on the left. The top bar shows the date 'May 9 15:25' and system icons on the right.

7. Explain the output of ls -l command.

The `ls -l` command in Linux is used to list directory contents in a detailed format. When we run this command, it provides a long listing of files and directories within the specified directory, including various attributes such as permissions, ownership, size, and modification time. Here's an explanation of the output:

1. File Type and Permissions: The first character indicates the type of file:

- `-` indicates a regular file.
- `d` indicates a directory.
- `l` indicates a symbolic link.
- `c` indicates a character special file.
- `b` indicates a block special file.
- `p` indicates a FIFO (named pipe).
- `s` indicates a socket.

Following the file type, there are nine characters representing the file permissions. Each group of three characters represents permissions for the file owner, the group, and others, respectively. The characters can be:

- `r` for read permission.
- `w` for write permission.
- `x` for execute permission.
- `-` indicates no permission for a particular action.

2. Number of Links: The second field indicates the number of hard links associated with the file or directory.

3. Owner: The third field specifies the owner of the file or directory.

4. Group: The fourth field specifies the group associated with the file or directory.

5. File Size: The fifth field indicates the size of the file in bytes.

6. Modification Time: The sixth field indicates the date and time when the file was last modified.

7. Filename: The last field displays the name of the file or directory.

Here's an example output of the `ls -l` command:

```
drwxr-xr-x 2 rps rps 4096 Apr 30 15:30 Desktop
```

In this example:

The example represents the detailed output of the `ls -l` command for a directory named "Desktop". Let's break down each component:

1. File Type and Permissions: The first character represents the file type. In this case, `d` indicates that it's a directory. The following nine characters represent the file permissions:

- `rw`: The owner (`rps`) has read, write, and execute permissions.
- `r-x`: The group (`rps`) has read and execute permissions.
- `r-x`: Others (users not in the owner group) also have read and execute permissions.

So, the directory permissions are `drwxr-xr-x`.

2. Number of Links: The second field indicates the number of hard links associated with the directory. In this case, there are 2 hard links.

3. Owner: The third and fourth fields indicate the owner and group of the directory, respectively. **In this case, both the owner and group are "rps".**

4. File Size: The fifth field represents the size of the directory. Since it's a directory, this size usually represents the amount of disk space it occupies. In this case, it's 4096 bytes.

5. Modification Time: The sixth field shows the date and time when the directory was last modified. In this case, it was modified on April 30th at 15:30.

6. Filename: The last field displays the name of the directory, which is "Desktop" in this example.

So, in summary, the output ``drwxr-xr-x 2 rps rps 4096 Apr 30 15:30 Desktop`` indicates that the "Desktop" directory is owned by the user "rps", belongs to the group "rps", has read, write, and execute permissions for the owner, and has read and execute permissions for both the group and others. It was last modified on April 30th at 15:30, and it occupies 4096 bytes of disk space.

Understanding the output of ``ls -l`` is essential for managing files and directories in Linux, as it provides detailed information about their attributes and permissions.

8. Difference between `ls -l` and `ls -la` command.

The ``ls -l`` and ``ls -la`` commands in Linux are both used to list directory contents in a detailed format, but they differ in the inclusiveness of their output. Here's the difference between them:

1. `ls -l`:

- This command provides a long listing of files and directories within the specified directory, including various attributes such as permissions, ownership, size, and modification time.
- It lists only the non-hidden files and directories. Hidden files and directories (those starting with a dot ``.``) are not displayed.

2. `ls -la`:

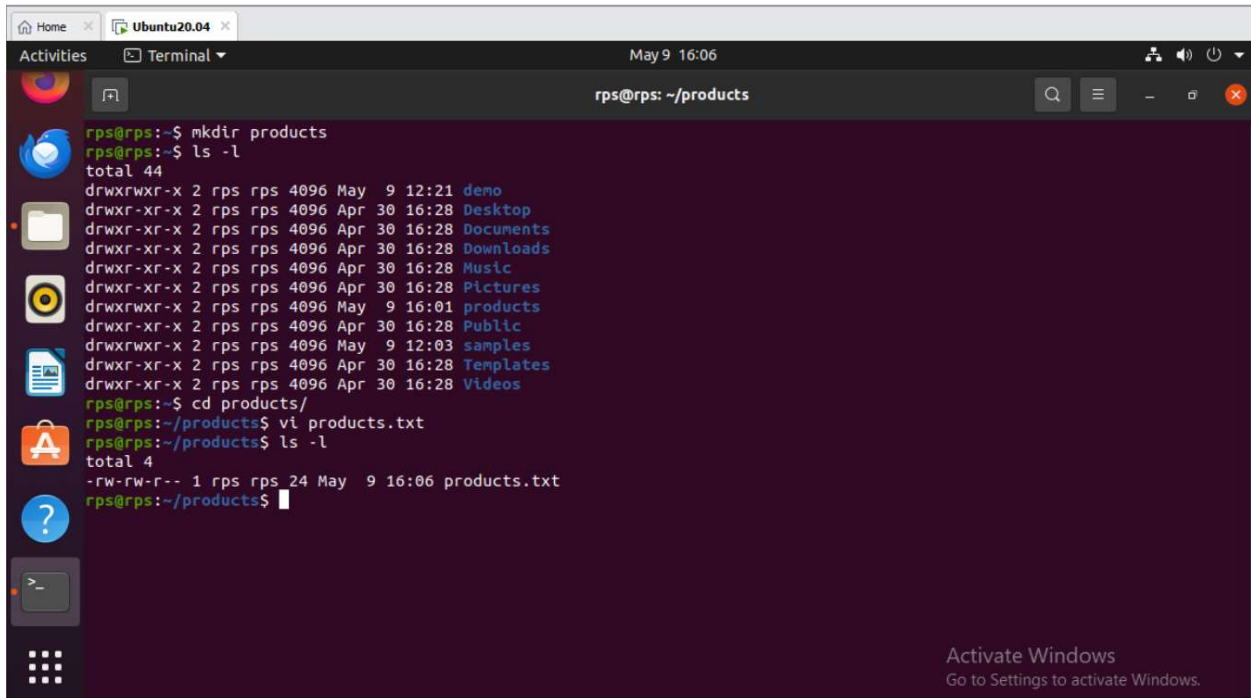
- This command also provides a long listing of files and directories within the specified directory, similar to ``ls -l``.
- However, it includes hidden files and directories in the output. The ``-a`` option stands for "all", which instructs ``ls`` to show all files, including those whose names begin with a dot ``.``.

In summary:

- `ls -l`: Lists non-hidden files and directories with detailed information.
- `ls -la`: Lists all files and directories, including hidden ones, with detailed information.

So, the key difference between ``ls -l`` and ``ls -la`` lies in whether they include hidden files and directories in their output.

9. Create a directory products in home directory, navigate using cd command and create a file products.txt.

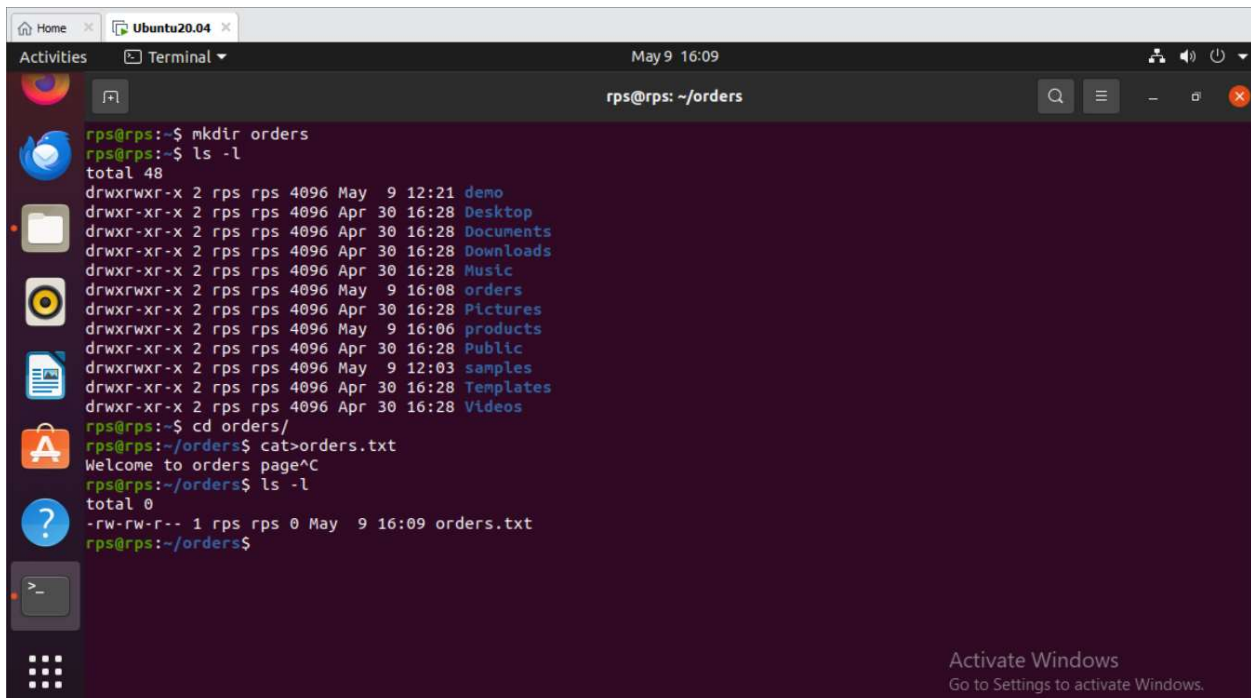


A terminal window titled 'Ubuntu20.04' showing the following commands and output:

```
rps@rps:~$ mkdir products
rps@rps:~$ ls -l
total 44
drwxrwxr-x 2 rps rps 4096 May  9 12:21 demo
drwxr-xr-x 2 rps rps 4096 Apr 30 16:28 Desktop
drwxr-xr-x 2 rps rps 4096 Apr 30 16:28 Documents
drwxr-xr-x 2 rps rps 4096 Apr 30 16:28 Downloads
drwxr-xr-x 2 rps rps 4096 Apr 30 16:28 Music
drwxr-xr-x 2 rps rps 4096 Apr 30 16:28 Pictures
drwxrwxr-x 2 rps rps 4096 May  9 16:01 products
drwxr-xr-x 2 rps rps 4096 Apr 30 16:28 Public
drwxrwxr-x 2 rps rps 4096 May  9 12:03 samples
drwxr-xr-x 2 rps rps 4096 Apr 30 16:28 Templates
drwxr-xr-x 2 rps rps 4096 Apr 30 16:28 Videos
rps@rps:~$ cd products/
rps@rps:~/products$ vi products.txt
rps@rps:~/products$ ls -l
total 4
-rw-rw-r-- 1 rps rps 24 May  9 16:06 products.txt
rps@rps:~/products$
```

The terminal window also shows a sidebar with application icons and a bottom status bar with the text 'Activate Windows Go to Settings to activate Windows.'

10. Create a directory orders in home directory, navigate using cd command and create a file orders.txt.



A terminal window titled 'Ubuntu20.04' showing the following commands and output:

```
rps@rps:~$ mkdir orders
rps@rps:~$ ls -l
total 48
drwxrwxr-x 2 rps rps 4096 May  9 12:21 demo
drwxr-xr-x 2 rps rps 4096 Apr 30 16:28 Desktop
drwxr-xr-x 2 rps rps 4096 Apr 30 16:28 Documents
drwxr-xr-x 2 rps rps 4096 Apr 30 16:28 Downloads
drwxr-xr-x 2 rps rps 4096 Apr 30 16:28 Music
drwxrwxr-x 2 rps rps 4096 May  9 16:08 orders
drwxr-xr-x 2 rps rps 4096 Apr 30 16:28 Pictures
drwxrwxr-x 2 rps rps 4096 May  9 16:06 products
drwxr-xr-x 2 rps rps 4096 Apr 30 16:28 Public
drwxrwxr-x 2 rps rps 4096 May  9 12:03 samples
drwxr-xr-x 2 rps rps 4096 Apr 30 16:28 Templates
drwxr-xr-x 2 rps rps 4096 Apr 30 16:28 Videos
rps@rps:~$ cd orders/
rps@rps:~/orders$ cat>orders.txt
Welcome to orders page^C
rps@rps:~/orders$ ls -l
total 0
-rw-rw-r-- 1 rps rps 0 May  9 16:09 orders.txt
rps@rps:~/orders$
```

The terminal window also shows a sidebar with application icons and a bottom status bar with the text 'Activate Windows Go to Settings to activate Windows.'

11. What is Path? Define Absolute Path and Relative Path.

A path in computing refers to the location of a file or directory within a file system. It specifies the route to access a particular file or directory starting from the root directory of the file system. There are two main types of paths: absolute paths and relative paths.

1. Absolute Path:

- An absolute path provides the complete location of a file or directory starting from the root directory of the file system.
- It includes all directories leading up to the target file or directory, starting from the root directory ("/" in Unix-like systems).
- It always begins with a forward slash "/".
- Example: `/home/user/Documents/file.txt`

2. Relative Path:

- A relative path specifies the location of a file or directory relative to the current working directory.
- It does not begin with the root directory; instead, it starts from the current directory.
- It is used when the exact location of the target file or directory is determined in relation to the current directory.
- Relative paths can use special symbols like `..` to refer to the parent directory and `.` to refer to the current directory.
- Example: If the current directory is `/home/user/`, and we want to refer to a file named `"document.txt"` located in the `"Documents"` directory within the current directory, the relative path would be `Documents/document.txt`.

In summary, an absolute path provides the full and exact location of a file or directory from the root directory, while a relative path specifies the location of a file or directory relative to the current working directory. Both types of paths are used to navigate and access files and directories within a file system.

12. List the differences between Absolute Path and Relative Path.

Aspect	Absolute Path	Relative Path
Definition	Provides complete location from root directory	Specifies location relative to current working directory
Starting Point	Root directory	Current working directory
Prefix	Starts with root directory	Does not start with root

	symbol ("/")	directory symbol
Usage	Precisely locate files/directories regardless of cwd	Specify location relative to current working directory
Portability	Less portable, specific to file system structure	More portable, depends on current working directory
Length	Can be longer, includes full directory path from root	Often shorter, specifies path relative to cwd
Dynamic vs. Static	Static, does not change regardless of cwd	Dynamic, changes based on current working directory
Ease of Use	Requires knowledge of file system structure	Generally easier to use, especially within projects

13. Navigate between products and orders with cd command using relative path reference.

```

rps@rps: ~/products
rps@rps:~/orders$ cd ../products
rps@rps:~/products$ pwd
/home/rps/products
rps@rps:~/products$ cd -
/home/rps/orders
rps@rps:~/orders$ pwd
/home/rps/orders
rps@rps:~/orders$ cd ../products/
rps@rps:~/products$ cd ../orders/
rps@rps:~/orders$ pwd
/home/rps/orders
rps@rps:~/orders$ cd -
/home/rps/products
rps@rps:~/products$ cd -
/home/rps/orders
rps@rps:~/orders$ cd -
/home/rps/products
rps@rps:~/products$

```

The screenshot shows a terminal window titled "Terminal" with a tab for "Ubuntu20.04". The user is logged in as "rps" and is currently in the directory "~/products". The terminal output shows a series of commands and their results, demonstrating how to navigate between the ~/products and ~/orders directories using relative paths (cd ../products, cd ../orders, cd -). The pwd command is used to verify the current directory at each step. The terminal window has a dark theme and a sidebar with application icons on the left. At the bottom right, there is a watermark that says "Activate Windows Go to Settings to activate Windows."