

DEAC102 – Introduction to Linux Systems Administration

Lab #3.2

Managing Files and Directories

Name: Aldenir D Flauzino

Student No.: CT1010383

Introduction: In this Lab you will learn how to navigate and manage files and directories.

You are required to complete the lab and record your answers. Rename the file using your first name and the lab number, e.g. **washington3.2.docx**, and submit it through Canvas to receive marks for this lab.

Tasks:

In this lab, you will perform the following tasks:

- Understand how to use globbing.
- Creating, moving and deleting files and directories

Equipment Required:

1. Device with Linux (UBUNTU)

Globbing

The use of **glob** characters in Linux is similar to what many operating systems refer to as "**wildcard**" characters. Using glob characters, you match filenames using patterns.

Glob characters are a shell feature, not something that is particular to any specific command. As a result, you can use glob characters with any Linux command.

When glob characters are used, the shell will "expand" the entire pattern to match all files in the specified directory that match the pattern.

For demonstration purposes, we will use the **echo** command to display this expansion process.

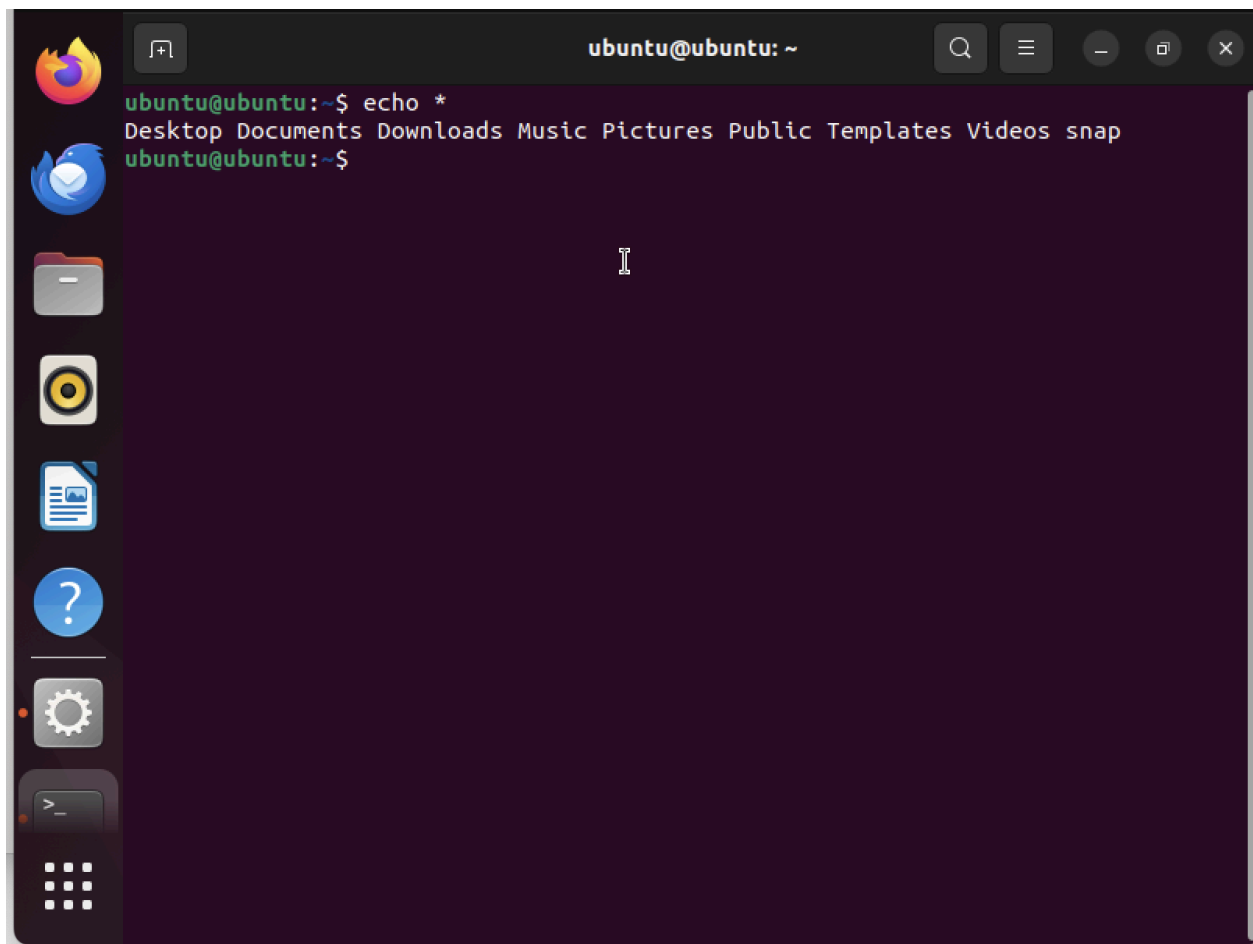
Step 1:

Use the following **echo** command to display all filenames in the current directory that match the glob pattern *****:

```
echo *
```

The **asterisk *** matches "zero or more" characters in a file name. The results in matching all filenames in the current directory.

The **echo** command, in turn, displays the filenames that were matched.



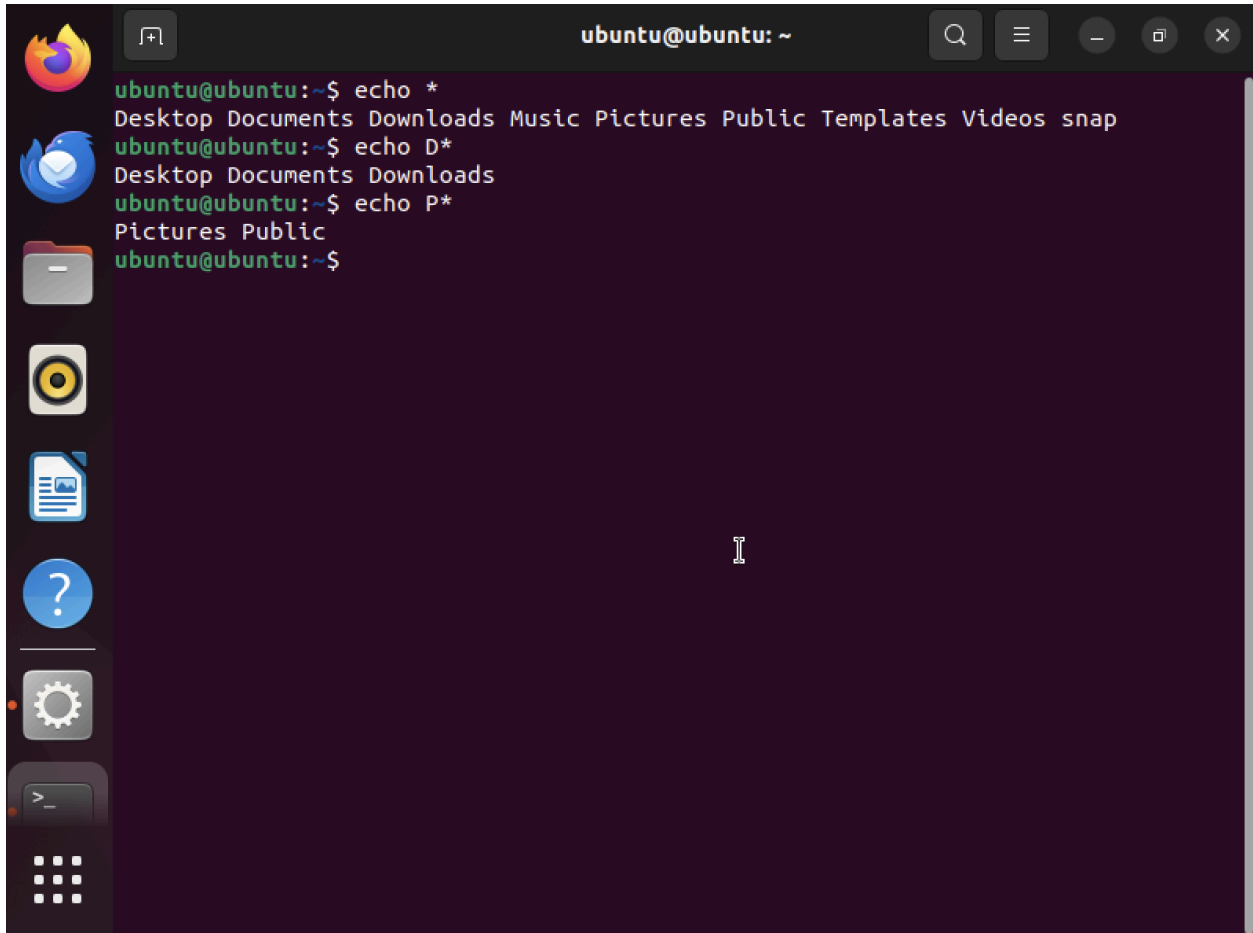
Step 2:

The following commands will display all the files in the current directory that start with the letter D, and the letter P:

```
echo D*
```

```
echo P*
```

Think of the first example, **D***, as "match all filenames in the current directory that begin with a capital d character and have zero or more of any other character after the **D**".

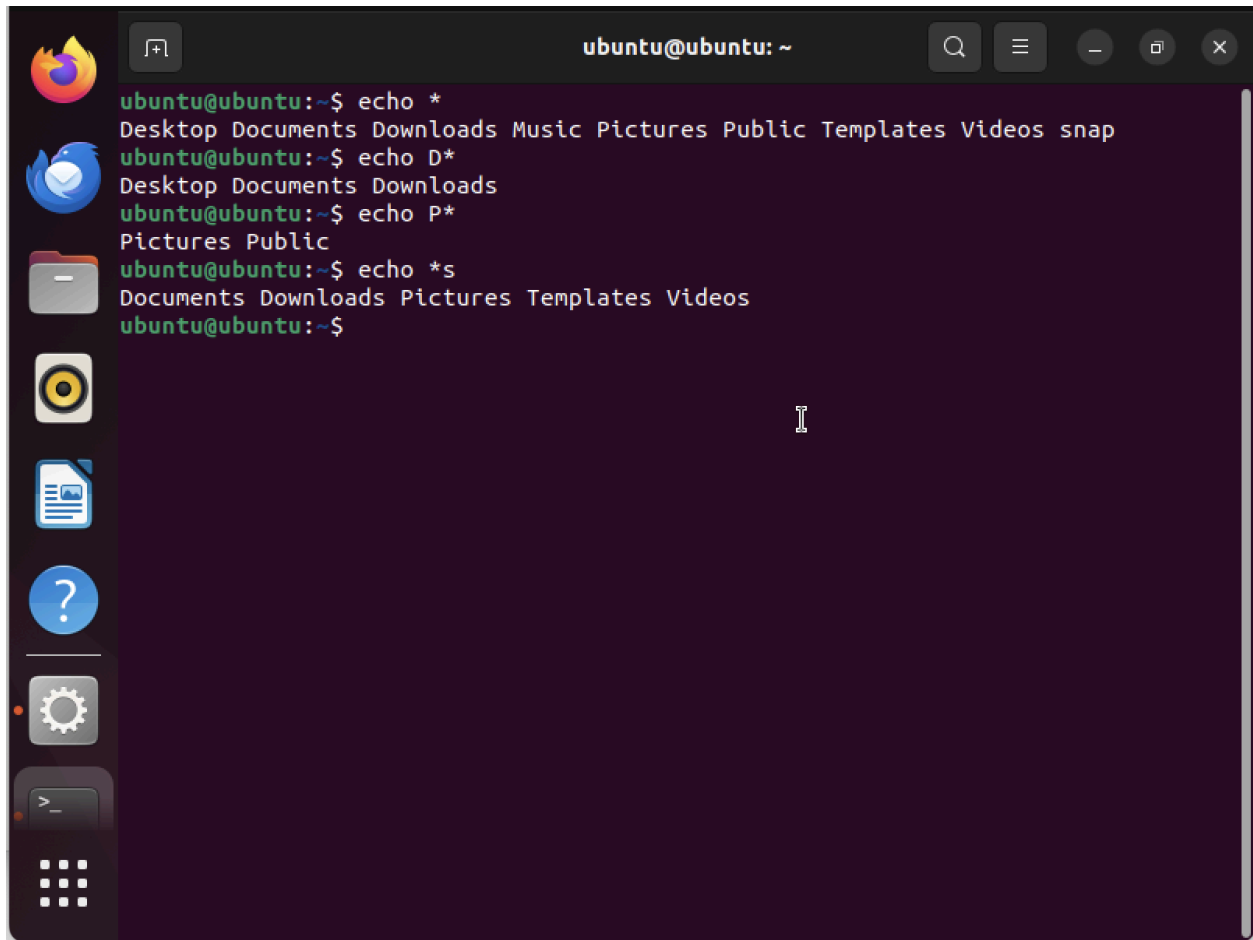
A terminal window titled 'ubuntu@ubuntu: ~' with standard window controls. The terminal shows a series of commands and their outputs. The first command is 'echo *', which lists all files in the home directory: Desktop, Documents, Downloads, Music, Pictures, Public, Templates, Videos, and snap. The second command is 'echo D*', which lists Desktop, Documents, and Downloads. The third command is 'echo P*', which lists Pictures and Public. The prompt 'ubuntu@ubuntu:~\$' is shown at the end of each line.

```
ubuntu@ubuntu:~$ echo *
Desktop Documents Downloads Music Pictures Public Templates Videos snap
ubuntu@ubuntu:~$ echo D*
Desktop Documents Downloads
ubuntu@ubuntu:~$ echo P*
Pictures Public
ubuntu@ubuntu:~$
```

Step 3:

The asterisk ***** can be used anywhere in the string. The following command will display all the files in your current directory that end in the letter **s**:

```
echo *s
```

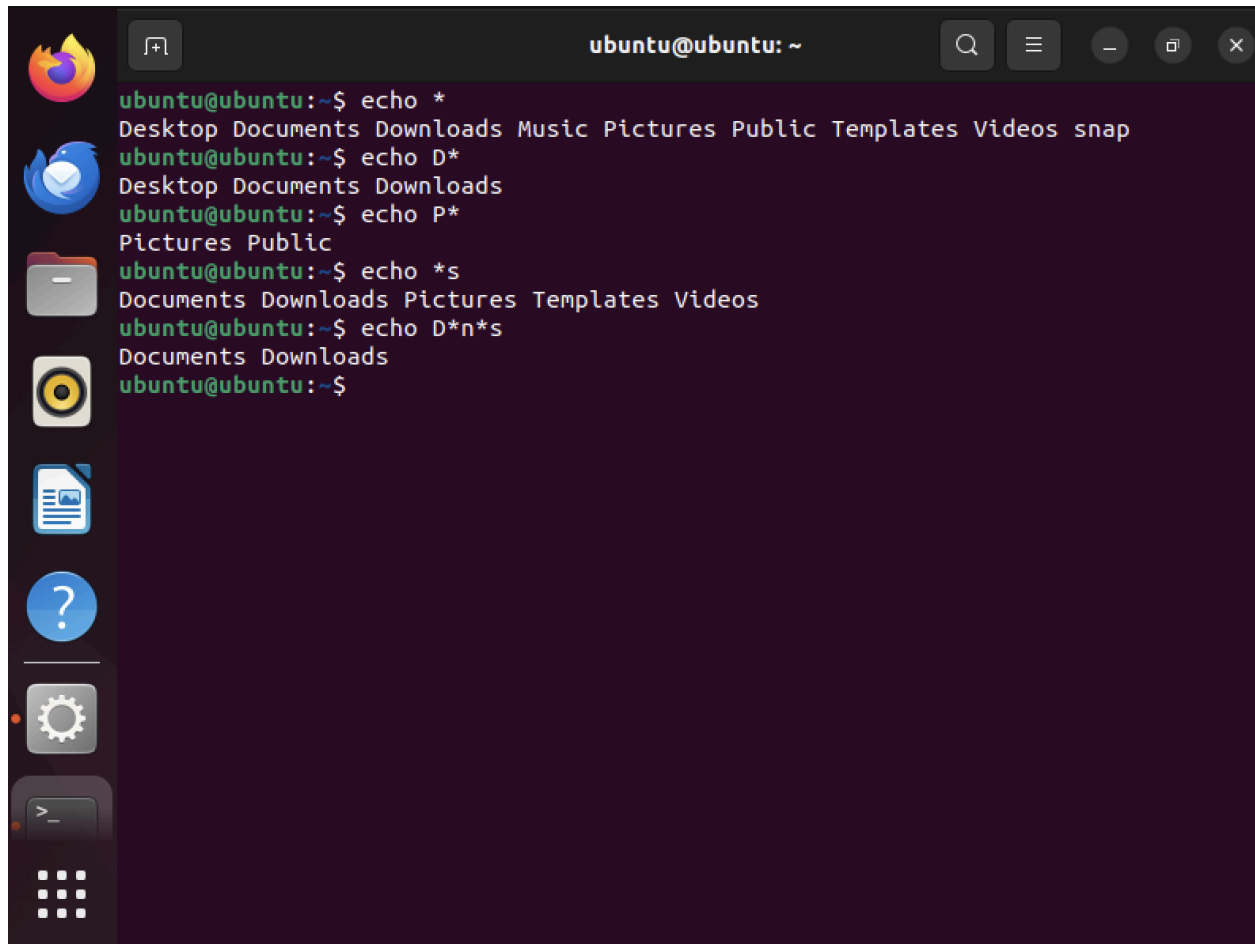
A terminal window titled 'ubuntu@ubuntu: ~' with a dark background. The left sidebar contains icons for Firefox, a mail client, a file manager, a music player, a document viewer, a help icon, and system settings. The terminal shows the following commands and output:

```
ubuntu@ubuntu:~$ echo *
Desktop Documents Downloads Music Pictures Public Templates Videos snap
ubuntu@ubuntu:~$ echo D*
Desktop Documents Downloads
ubuntu@ubuntu:~$ echo P*
Pictures Public
ubuntu@ubuntu:~$ echo *s
Documents Downloads Pictures Templates Videos
ubuntu@ubuntu:~$
```

Step 4:

Notice that the asterisk can also appear multiple times or in the middle of several characters:

```
echo D*n*s
```

A terminal window titled 'ubuntu@ubuntu: ~' with a dark purple background. On the left is a vertical dock with icons for Firefox, a web browser, a file manager, a music player, a document viewer, a help icon, settings, and a terminal. The terminal shows a series of commands and their outputs: 'echo *' lists directories; 'echo D*' lists 'Desktop'; 'echo P*' lists 'Pictures' and 'Public'; 'echo *s' lists 'Documents', 'Downloads', 'Pictures', 'Templates', and 'Videos'; 'echo D*n*s' lists 'Documents' and 'Downloads'.

Step 5:

The next glob metacharacter that we will examine is the **question mark ?**. The question mark matches exactly one character. This single character can be any possible character.

Like the asterisk, it can be used anywhere in a string and can appear multiple times.

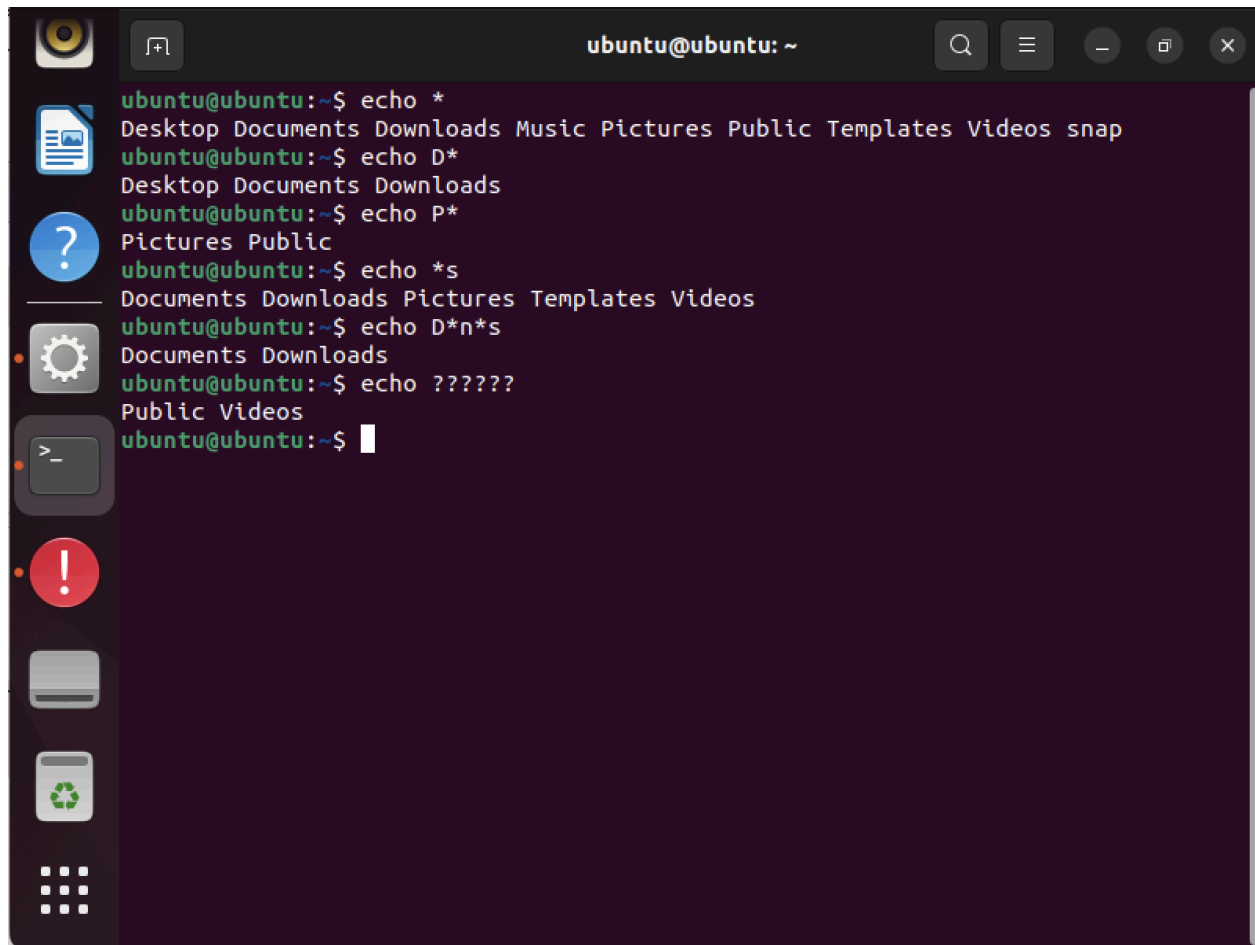
Since each question mark matches one unknown character, typing six of them will match six-character filenames.

Type the following to display the filenames that are exactly six characters long:

```
echo ??????
```

Important

Each ? character must match exactly one character in a filename-no more and no less than one character.



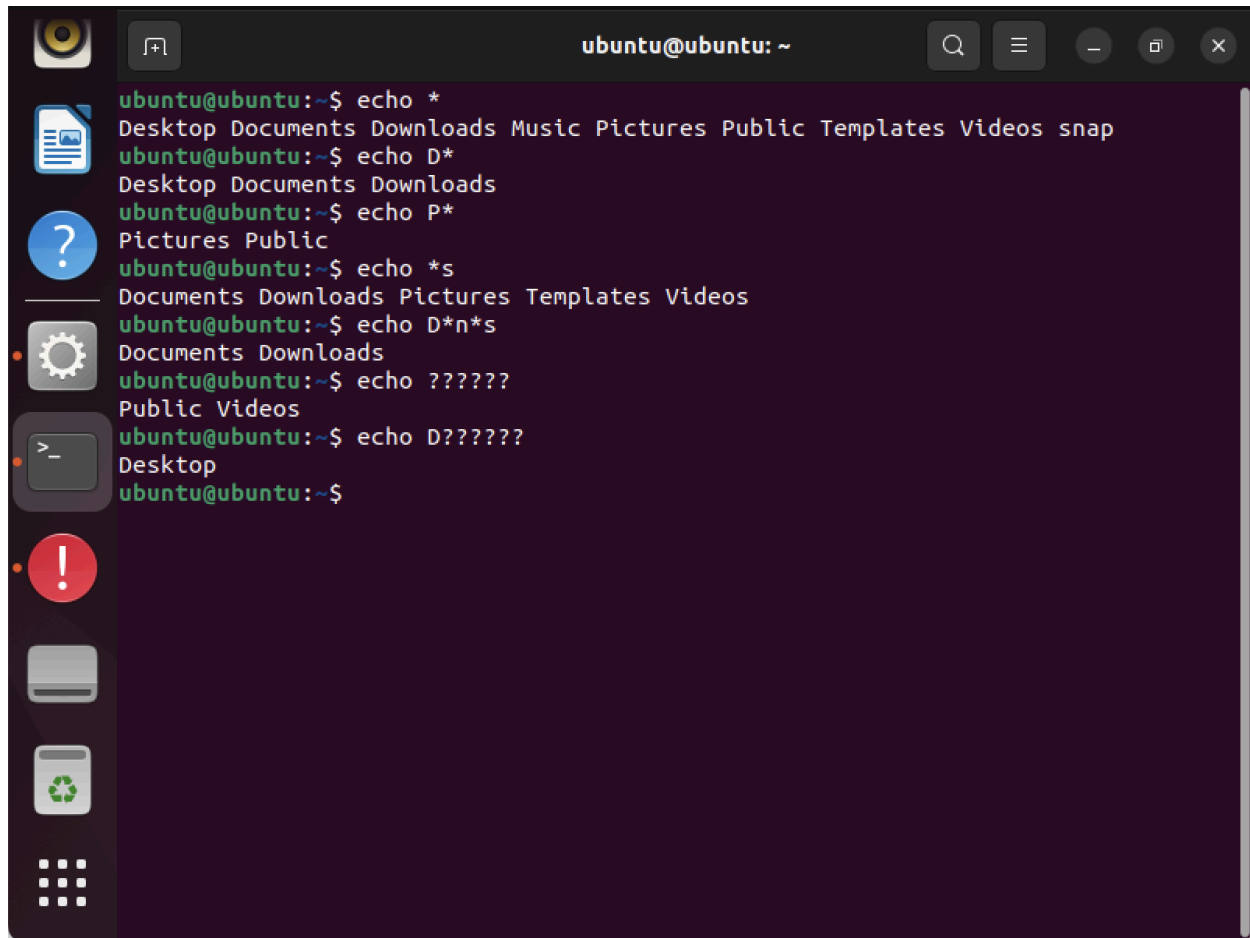
A terminal window titled 'ubuntu@ubuntu: ~' with search, menu, and window control buttons. The terminal shows a series of commands and their outputs:

```
ubuntu@ubuntu:~$ echo *
Desktop Documents Downloads Music Pictures Public Templates Videos snap
ubuntu@ubuntu:~$ echo D*
Desktop Documents Downloads
ubuntu@ubuntu:~$ echo P*
Pictures Public
ubuntu@ubuntu:~$ echo *s
Documents Downloads Pictures Templates Videos
ubuntu@ubuntu:~$ echo D*n*s
Documents Downloads
ubuntu@ubuntu:~$ echo ??????
Public Videos
ubuntu@ubuntu:~$
```

Step 6:

Using the question mark with other characters will limit the matches. Type the following to display the file names that start with the letter D and are exactly nine characters long:

```
echo D?????????
```

A terminal window titled 'ubuntu@ubuntu: ~' with standard window controls. The terminal shows a series of 'echo' commands and their outputs. The first command 'echo *' lists all files in the home directory. Subsequent commands use wildcards like 'D*', 'P*', '*s', 'D*n*s', '??????', and 'D?????' to filter the results. The left sidebar of the terminal window contains several icons: a yellow circle with a black dot, a blue document icon, a blue question mark, a grey gear, a grey terminal icon, a red circle with a white exclamation mark, a grey printer icon, a grey trash can icon, and a 3x3 grid of white dots.

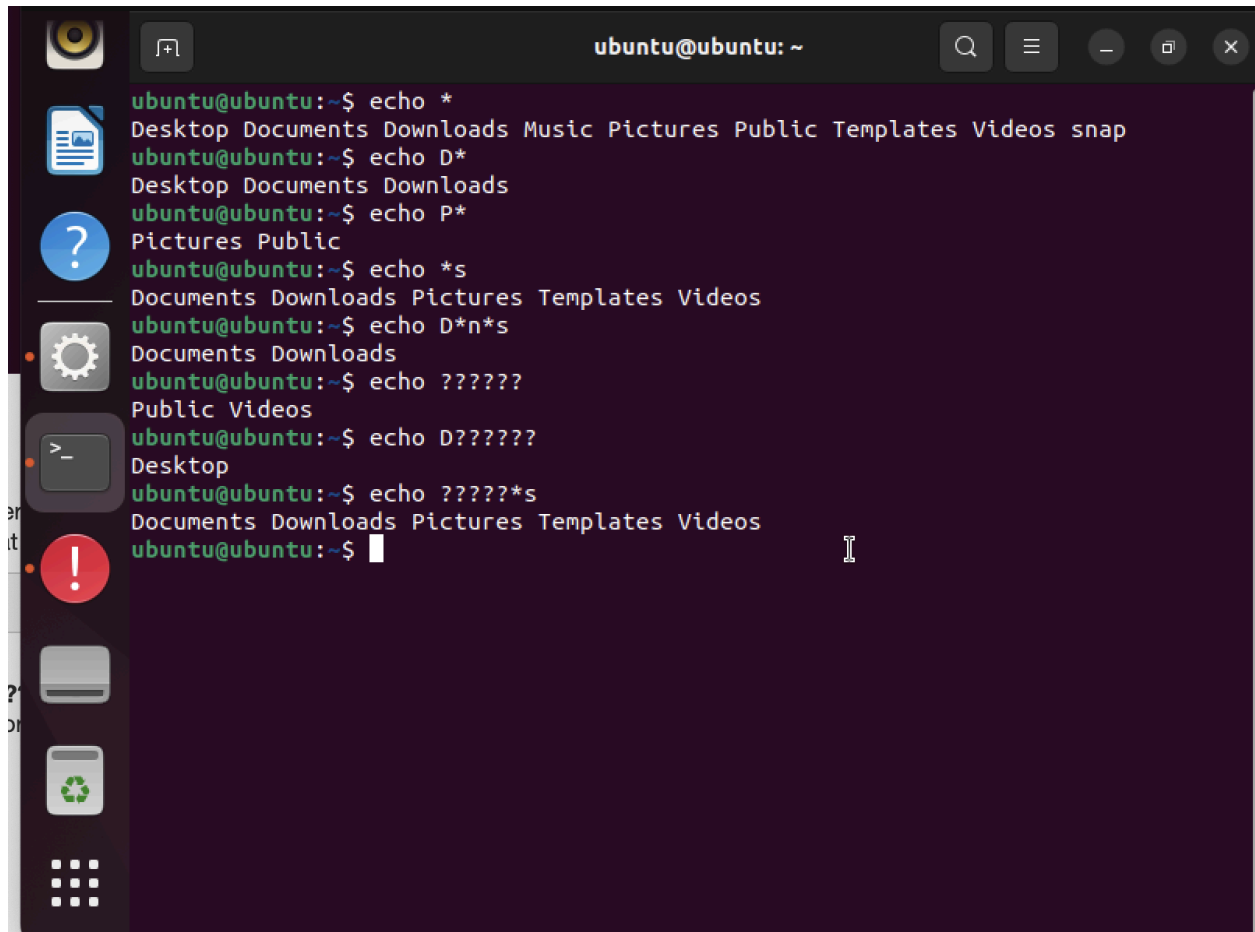
```
ubuntu@ubuntu:~$ echo *
Desktop Documents Downloads Music Pictures Public Templates Videos snap
ubuntu@ubuntu:~$ echo D*
Desktop Documents Downloads
ubuntu@ubuntu:~$ echo P*
Pictures Public
ubuntu@ubuntu:~$ echo *s
Documents Downloads Pictures Templates Videos
ubuntu@ubuntu:~$ echo D*n*s
Documents Downloads
ubuntu@ubuntu:~$ echo ??????
Public Videos
ubuntu@ubuntu:~$ echo D?????
Desktop
ubuntu@ubuntu:~$
```

Step 7:

Wildcards or glob characters can be **combined together**. The following command will display file names that are at least six characters long and end in the letter s.

```
echo ?????*s
```

Think of the pattern **?????*s to mean** "match filenames that begin with any five characters, then have zero or more of any characters and then end with an s character".

A terminal window titled 'ubuntu@ubuntu: ~' with standard window controls. The terminal shows a series of commands and their outputs. The first command is 'echo *', which lists directory contents. Subsequent commands use wildcards like 'D*', 'P*', '*s', and 'D*n*s' to filter the output. The final command is 'echo D?????', which outputs 'Desktop'.

```
ubuntu@ubuntu:~$ echo *
Desktop Documents Downloads Music Pictures Public Templates Videos snap
ubuntu@ubuntu:~$ echo D*
Desktop Documents Downloads
ubuntu@ubuntu:~$ echo P*
Pictures Public
ubuntu@ubuntu:~$ echo *s
Documents Downloads Pictures Templates Videos
ubuntu@ubuntu:~$ echo D*n*s
Documents Downloads
ubuntu@ubuntu:~$ echo ??????
Public Videos
ubuntu@ubuntu:~$ echo D??????
Desktop
ubuntu@ubuntu:~$ echo ?????*s
Documents Downloads Pictures Templates Videos
ubuntu@ubuntu:~$
```

Step 8:

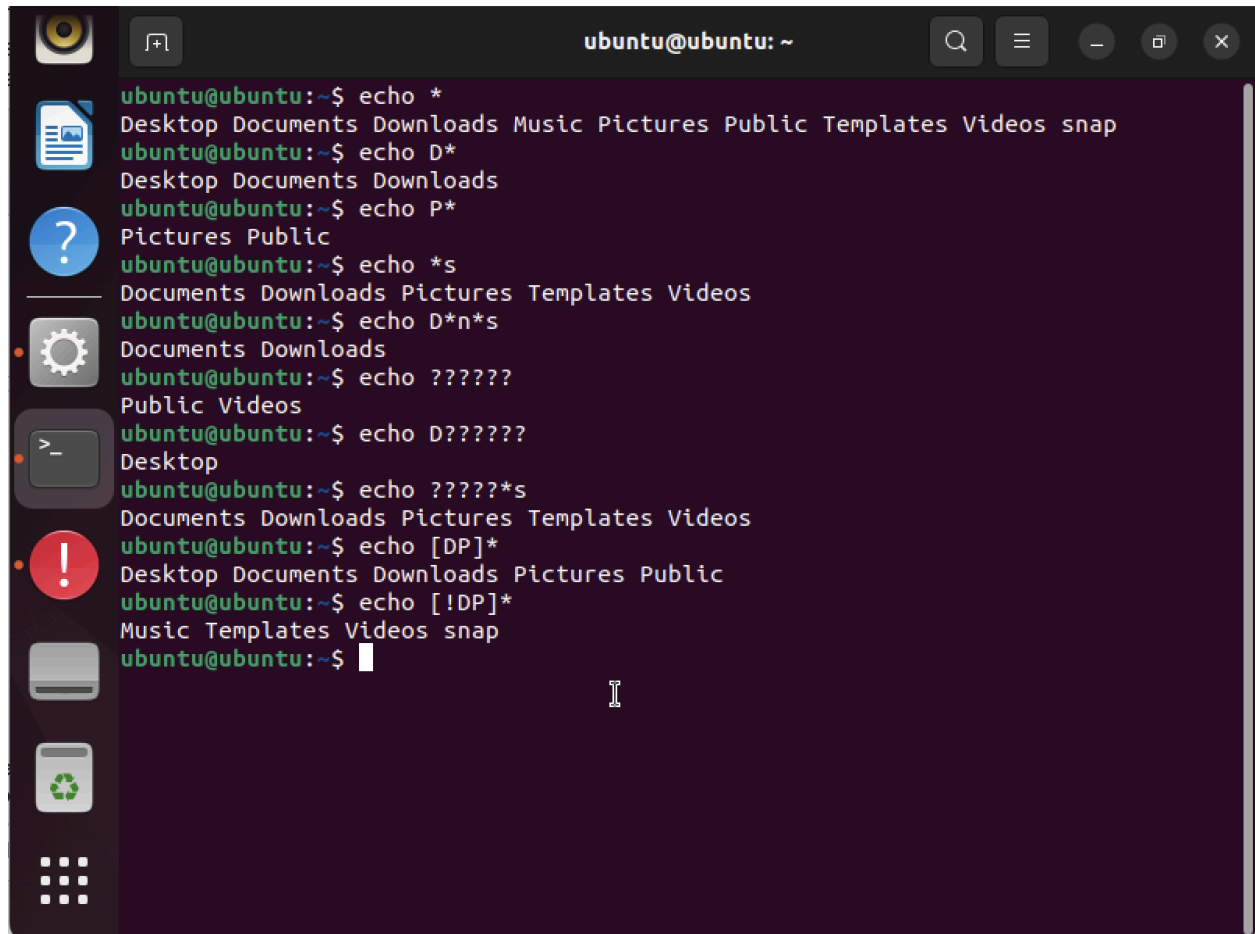
The next glob is similar to the question mark glob to specify one character.

This glob uses a pair of square **brackets** `[]` to specify which one character will be allowed. The allowed characters can be specified as a range, a list, or by what is known as a character class.

The allowed characters can also be negated with an **exclamation point** `!`.

In the first example, the first character of the file name can be either a D or a P. In the second example, the first character can be any character **except** a D or P:

```
echo [DP]*
echo [!DP]*
```


A terminal window titled 'ubuntu@ubuntu: ~' with a dark purple background. The left sidebar contains icons for Home, Files, Applications, Settings, Terminal, and a red warning icon. The terminal shows a series of 'echo' commands using wildcards to list files in the current directory. The output of each command is shown on the next line.

```
ubuntu@ubuntu:~$ echo *
Desktop Documents Downloads Music Pictures Public Templates Videos snap
ubuntu@ubuntu:~$ echo D*
Desktop Documents Downloads
ubuntu@ubuntu:~$ echo P*
Pictures Public
ubuntu@ubuntu:~$ echo *s
Documents Downloads Pictures Templates Videos
ubuntu@ubuntu:~$ echo D*n*s
Documents Downloads
ubuntu@ubuntu:~$ echo ??????
Public Videos
ubuntu@ubuntu:~$ echo D?????
Desktop
ubuntu@ubuntu:~$ echo ?????*s
Documents Downloads Pictures Templates Videos
ubuntu@ubuntu:~$ echo [DP]*
Desktop Documents Downloads Pictures Public
ubuntu@ubuntu:~$ echo [!DP]*
Music Templates Videos snap
ubuntu@ubuntu:~$
```

Step 9:

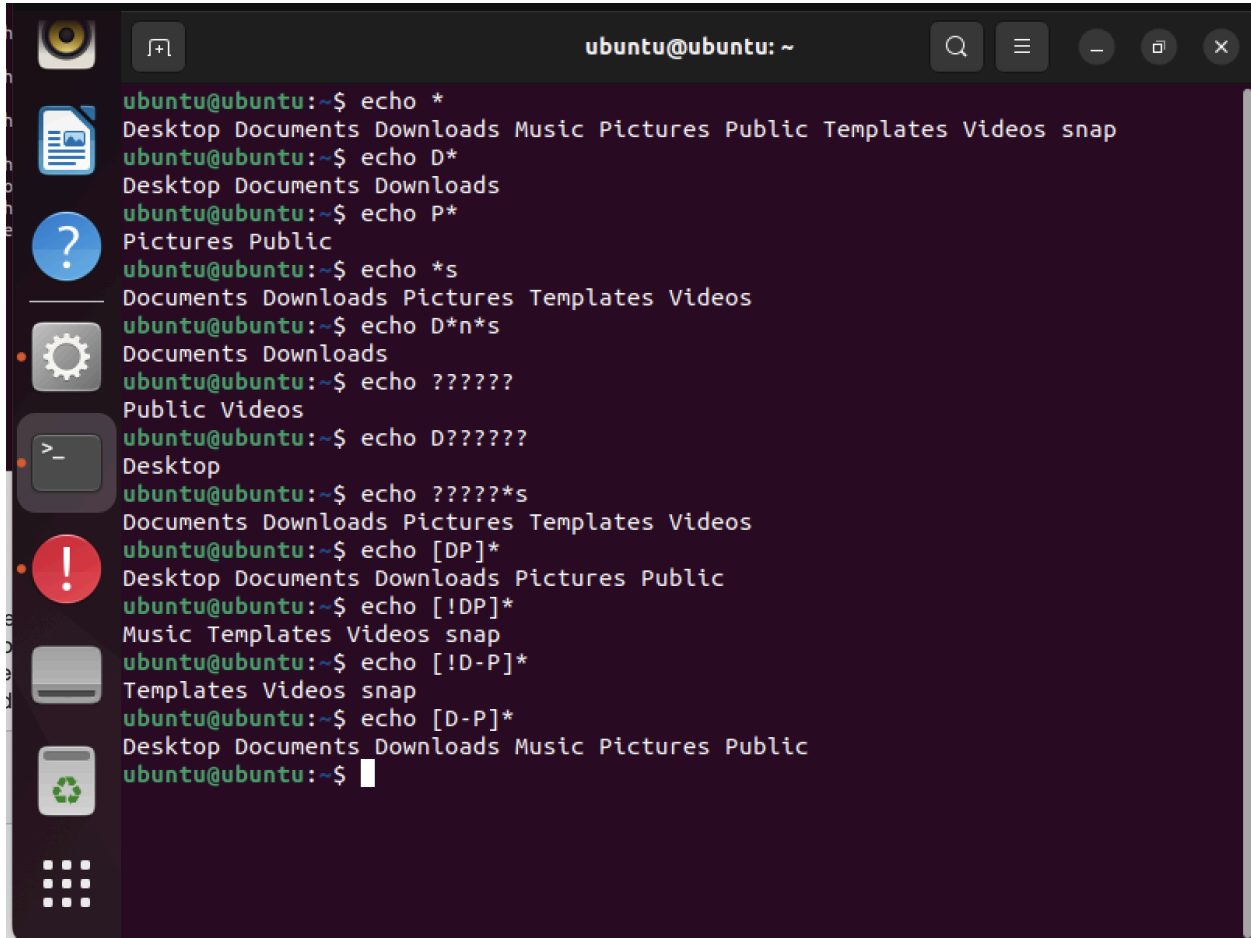
In these next examples, a range of characters will be specified. In the first example, the first character of the file name can be any character starting at **D** and ending at **P**. In the second example, this range of characters is negated, meaning any single character will match as long as it is not between the letters **D** and **P**:

```
echo [D-P] *
echo [!D-P] *
```

You may be asking yourself "who decides what letters come between **D** and **P**"? In this case, the answer is fairly obvious (E, F, G, H, I, J, K, L, M, N and O), but what if the range was **[1-A]**?

The ASCII text table is used to determine the range of characters. You can view this table by searching for it on the Internet or typing the following command: **ascii**

So, what characters does the glob **[1-A]** match? According to the ASCII text table: **1, 2, 3, 4, 5, 6, 7, 8, 9, :, ;, <, =, >, ?, @ and A.**

A terminal window titled 'ubuntu@ubuntu: ~' with a dark background. The left sidebar shows icons for Home, Files, Applications, Settings, Dash, and Recycle Bin. The terminal displays the following commands and their outputs:

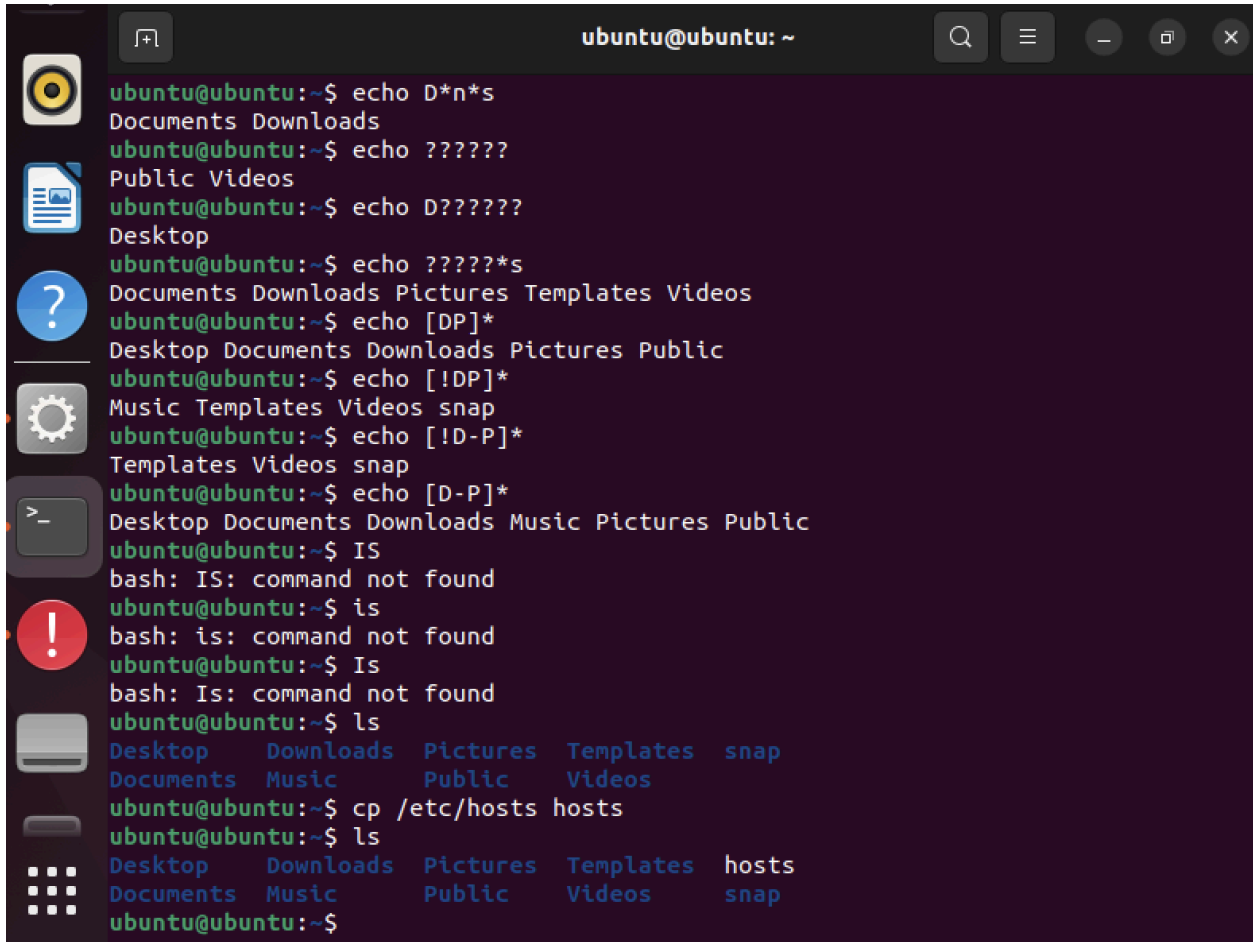
```
ubuntu@ubuntu:~$ echo *
Desktop Documents Downloads Music Pictures Public Templates Videos snap
ubuntu@ubuntu:~$ echo D*
Desktop Documents Downloads
ubuntu@ubuntu:~$ echo P*
Pictures Public
ubuntu@ubuntu:~$ echo *s
Documents Downloads Pictures Templates Videos
ubuntu@ubuntu:~$ echo D*n*s
Documents Downloads
ubuntu@ubuntu:~$ echo ??????
Public Videos
ubuntu@ubuntu:~$ echo D?????
Desktop
ubuntu@ubuntu:~$ echo ?????*s
Documents Downloads Pictures Templates Videos
ubuntu@ubuntu:~$ echo [DP]*
Desktop Documents Downloads Pictures Public
ubuntu@ubuntu:~$ echo [!DP]*
Music Templates Videos snap
ubuntu@ubuntu:~$ echo [!D-P]*
Templates Videos snap
ubuntu@ubuntu:~$ echo [D-P]*
Desktop Documents Downloads Music Pictures Public
ubuntu@ubuntu:~$
```

Step 10:

Make a copy of the `/etc/hosts` file and place it in the current directory. Then, list the contents of the current directory before and after the copy:

```
ls
cp /etc/hosts hosts
ls
```

Notice how the second `ls` command displays a copy of the `hosts` file.

A terminal window titled 'ubuntu@ubuntu: ~' with standard window controls. The terminal shows a series of commands and their outputs for file discovery. The commands include 'echo D*n*s', 'echo ??????', 'echo D?????', 'echo ?????*s', 'echo [DP]*', 'echo [!DP]*', 'echo [!D-P]*', 'echo [D-P]*', 'IS', 'is', 'Is', 'ls', and 'cp /etc/hosts hosts'. The outputs list various system directories like Documents, Downloads, Desktop, Music, Pictures, Templates, Videos, and Public. The 'ls' command shows a directory listing, and the 'cp' command copies the /etc/hosts file to the current directory.

```
ubuntu@ubuntu:~$ echo D*n*s
Documents Downloads
ubuntu@ubuntu:~$ echo ??????
Public Videos
ubuntu@ubuntu:~$ echo D?????
Desktop
ubuntu@ubuntu:~$ echo ?????*s
Documents Downloads Pictures Templates Videos
ubuntu@ubuntu:~$ echo [DP]*
Desktop Documents Downloads Pictures Public
ubuntu@ubuntu:~$ echo [!DP]*
Music Templates Videos snap
ubuntu@ubuntu:~$ echo [!D-P]*
Templates Videos snap
ubuntu@ubuntu:~$ echo [D-P]*
Desktop Documents Downloads Music Pictures Public
ubuntu@ubuntu:~$ IS
bash: IS: command not found
ubuntu@ubuntu:~$ is
bash: is: command not found
ubuntu@ubuntu:~$ Is
bash: Is: command not found
ubuntu@ubuntu:~$ ls
Desktop  Downloads  Pictures  Templates  snap
Documents Music      Public   Videos
ubuntu@ubuntu:~$ cp /etc/hosts hosts
ubuntu@ubuntu:~$ ls
Desktop  Downloads  Pictures  Templates  hosts
Documents Music      Public   Videos   snap
ubuntu@ubuntu:~$
```

Step 11:

Next, you will remove the file, then copy it again, but have the system tell you what is being done. This can be achieved using the `-v` or `--verbose` option.

Enter the following commands:

```
rm hosts
ls
cp -v /etc/hosts hosts
ls
```

Note that the `rm` command is used to delete a file. More information on this command will be provided later in this lab.

Note that the `-v` switch displays the source and target when the `cp` command is executed:

Source

```
`/etc/hosts' -> `hosts'
```

Target

```
`/etc/hosts' -> `hosts'
```

```
ubuntu@ubuntu: ~  
Desktop Documents Downloads Pictures Public  
ubuntu@ubuntu:~$ echo [!DP]*  
Music Templates Videos snap  
ubuntu@ubuntu:~$ echo [!D-P]*  
Templates Videos snap  
ubuntu@ubuntu:~$ echo [D-P]*  
Desktop Documents Downloads Music Pictures Public  
ubuntu@ubuntu:~$ IS  
bash: IS: command not found  
ubuntu@ubuntu:~$ is  
bash: is: command not found  
ubuntu@ubuntu:~$ Is  
bash: Is: command not found  
ubuntu@ubuntu:~$ ls  
Desktop Downloads Pictures Templates snap  
Documents Music Public Videos  
ubuntu@ubuntu:~$ cp /etc/hosts hosts  
ubuntu@ubuntu:~$ ls  
Desktop Downloads Pictures Templates hosts  
Documents Music Public Videos snap  
ubuntu@ubuntu:~$ rm hosts  
ubuntu@ubuntu:~$ ls  
Desktop Downloads Pictures Templates snap  
Documents Music Public Videos  
ubuntu@ubuntu:~$ cp -v /etc/hosts hosts  
'/etc/hosts' -> 'hosts'  
ubuntu@ubuntu:~$ ls  
Desktop Downloads Pictures Templates hosts  
Documents Music Public Videos snap  
ubuntu@ubuntu:~$ ls
```

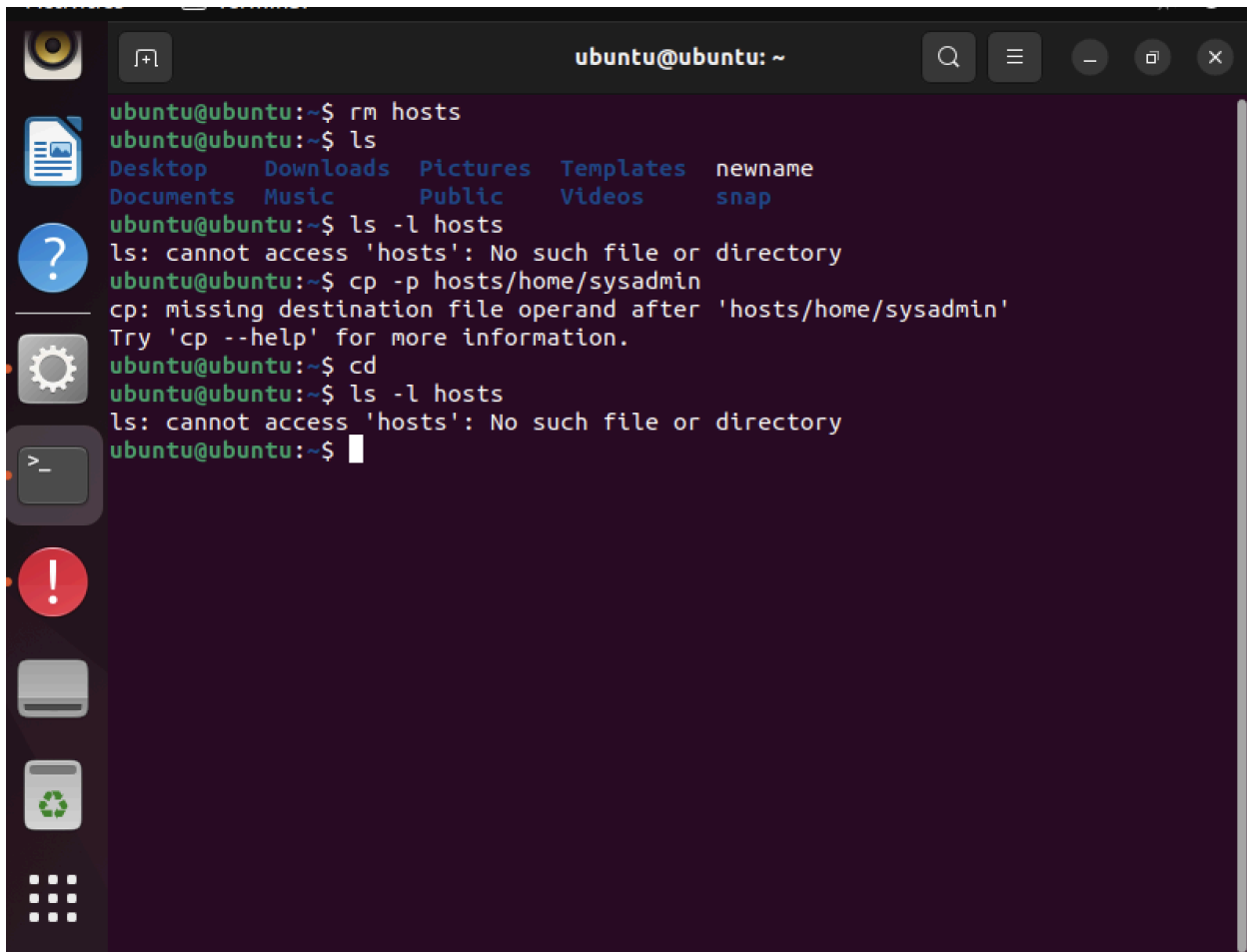
Step 12:

Enter the following commands to copy from the source directory and preserve file attributes by using the `-p` option:

```
rm hosts
```

```
ls
cd /etc
ls -l hosts
cp -p hosts /home/sysadmin
cd
ls -l hosts
```

Notice that the date and permission modes were preserved. Note that the timestamp in the output is the same for both the original and the copy.

A terminal window titled 'ubuntu@ubuntu: ~' with standard window controls. The terminal shows a sequence of commands and their outputs. The user first removes the 'hosts' file, then lists the directory contents. They attempt to copy 'hosts' to '/home/sysadmin' but receive an error about a missing destination file operand. After changing the directory to '~', they attempt to list 'hosts' again, receiving another 'No such file or directory' error.

```
ubuntu@ubuntu:~$ rm hosts
ubuntu@ubuntu:~$ ls
Desktop  Downloads  Pictures  Templates  newname
Documents Music      Public   Videos    snap
ubuntu@ubuntu:~$ ls -l hosts
ls: cannot access 'hosts': No such file or directory
ubuntu@ubuntu:~$ cp -p hosts/home/sysadmin
cp: missing destination file operand after 'hosts/home/sysadmin'
Try 'cp --help' for more information.
ubuntu@ubuntu:~$ cd
ubuntu@ubuntu:~$ ls -l hosts
ls: cannot access 'hosts': No such file or directory
ubuntu@ubuntu:~$
```

Step 13:

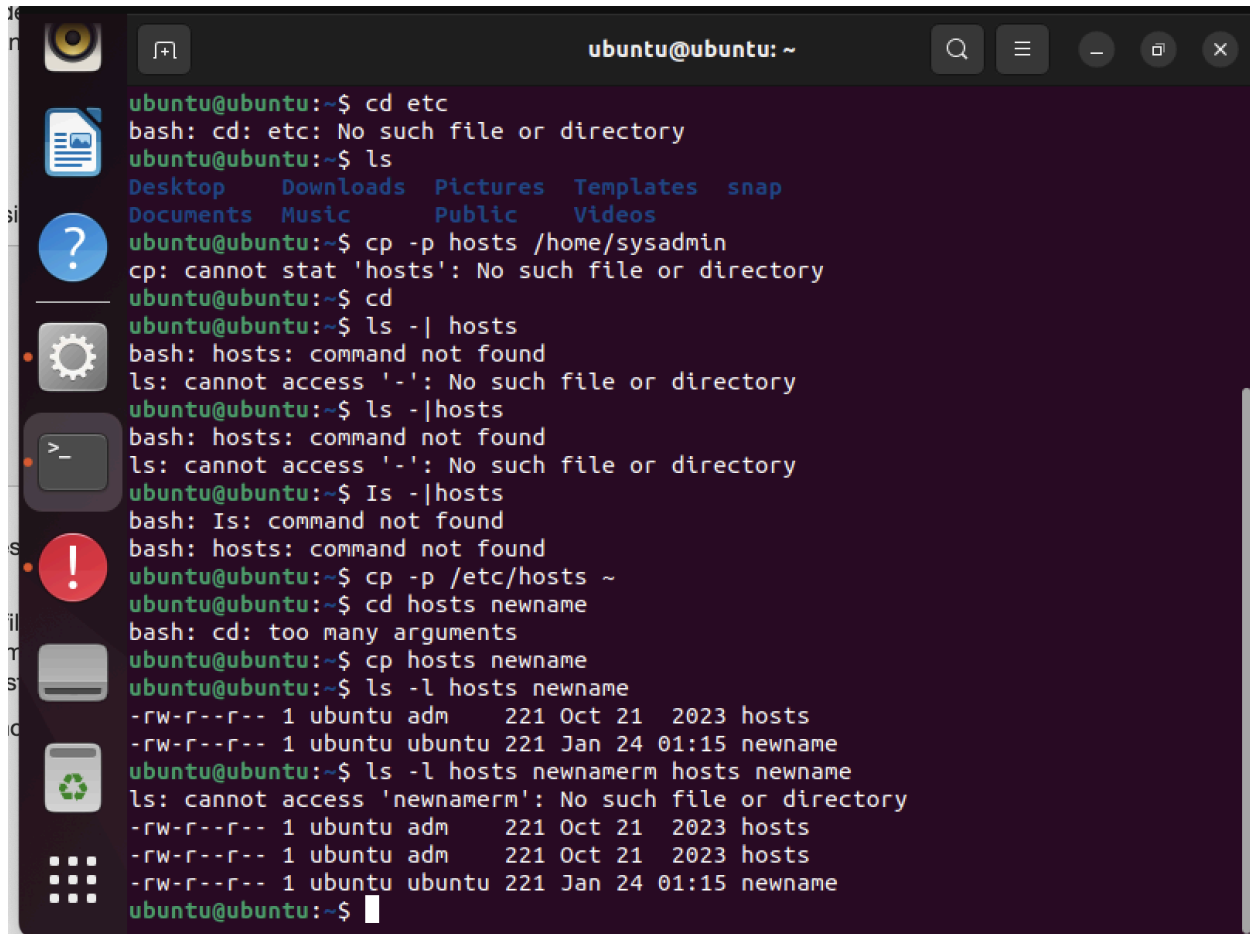
Type the following commands to copy using a different target name:

```
rm hosts
cp -p /etc/hosts ~
cp hosts newname
ls -l hosts newname
rm hosts newname
```

The first copy with the `-p` option preserved the original timestamp. Recall that the tilde `~` represents **your home directory**.

The second copy specified a different filename (**newname**) as the target. Because it was issued without the `-p` option, the system used the current date and time for the target; thus, it did not preserve the original timestamp found in the source file **/etc/hosts**.

Finally, note that you can remove more than one file at a time as shown in the last `rm` command.



```
ubuntu@ubuntu: ~  
ubuntu@ubuntu:~$ cd etc  
bash: cd: etc: No such file or directory  
ubuntu@ubuntu:~$ ls  
Desktop  Downloads  Pictures  Templates  snap  
Documents  Music      Public    Videos  
ubuntu@ubuntu:~$ cp -p hosts /home/sysadmin  
cp: cannot stat 'hosts': No such file or directory  
ubuntu@ubuntu:~$ cd  
ubuntu@ubuntu:~$ ls -l hosts  
bash: hosts: command not found  
ls: cannot access '-': No such file or directory  
ubuntu@ubuntu:~$ ls -l hosts  
bash: hosts: command not found  
ls: cannot access '-': No such file or directory  
ubuntu@ubuntu:~$ ls -l hosts  
bash: ls: command not found  
ubuntu@ubuntu:~$ cp -p /etc/hosts ~  
ubuntu@ubuntu:~$ cd hosts newname  
bash: cd: too many arguments  
ubuntu@ubuntu:~$ cp hosts newname  
ubuntu@ubuntu:~$ ls -l hosts newname  
-rw-r--r-- 1 ubuntu adm    221 Oct 21  2023 hosts  
-rw-r--r-- 1 ubuntu ubuntu 221 Jan 24 01:15 newname  
ubuntu@ubuntu:~$ ls -l hosts newnamerm hosts newname  
ls: cannot access 'newnamerm': No such file or directory  
-rw-r--r-- 1 ubuntu adm    221 Oct 21  2023 hosts  
-rw-r--r-- 1 ubuntu adm    221 Oct 21  2023 hosts  
-rw-r--r-- 1 ubuntu ubuntu 221 Jan 24 01:15 newname  
ubuntu@ubuntu:~$
```

Step 14:

To copy all files in a directory, use the **-R** option.

For this task, we will copy the **/etc/udev** directory into a new directory and display the contents that were copied there. Naturally, the directory must be created before files can be added to it.

In this example we will use the default settings for **mkdir** to create the “**Myetc**” directory.

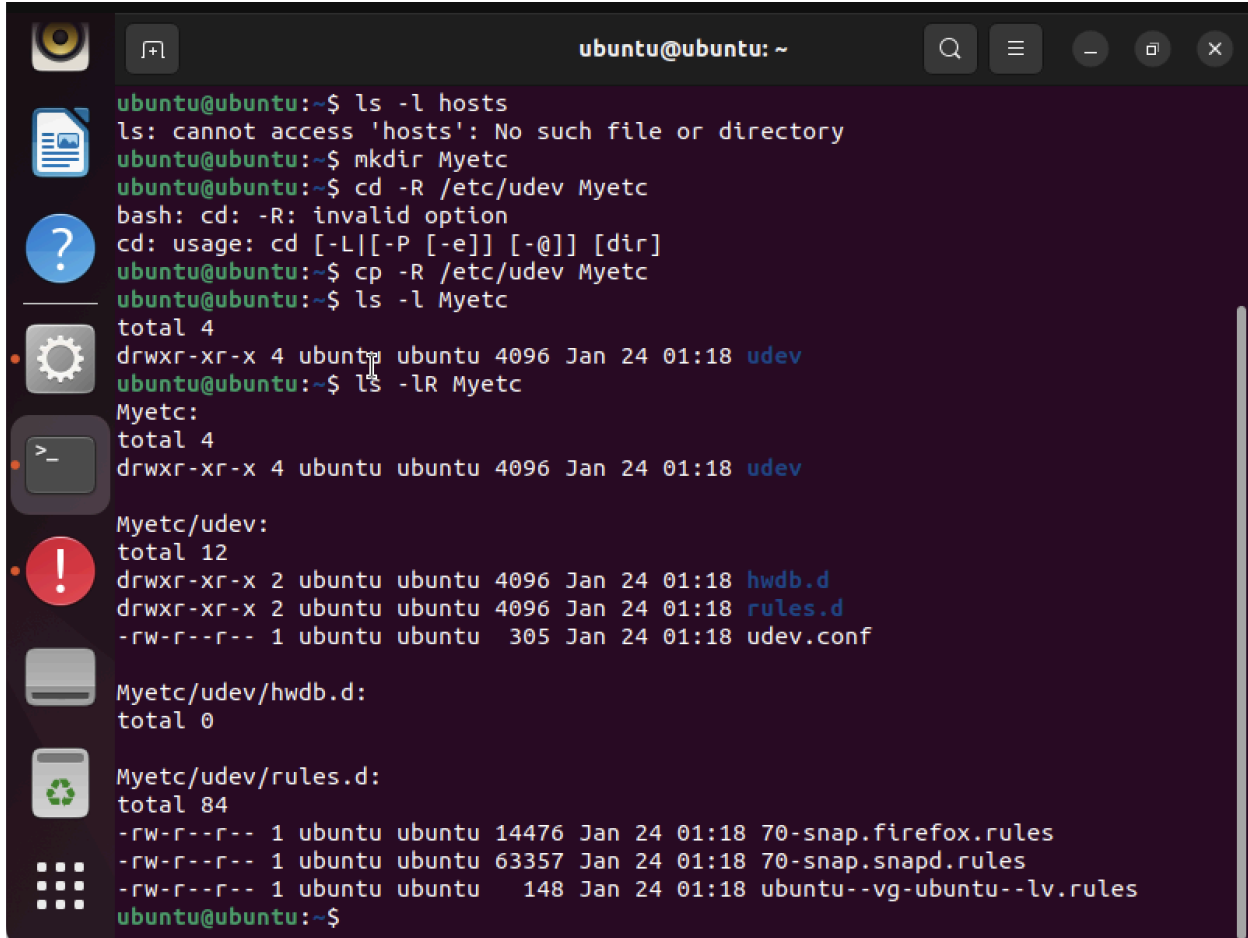
Options are available for the **mkdir** command to set security, permissions and other attributes of a new directory. Once the directory has been copied, the **ls** command is used to list the contents of the directory with both the long and recursive options.

```
mkdir Myetc
```

```
cp -R /etc/udev Myetc
```

```
ls -l Myetc
```

```
ls -lR Myetc
```



```
ubuntu@ubuntu: ~  
ubuntu@ubuntu:~$ ls -l hosts  
ls: cannot access 'hosts': No such file or directory  
ubuntu@ubuntu:~$ mkdir Myetc  
ubuntu@ubuntu:~$ cd -R /etc/udev Myetc  
bash: cd: -R: invalid option  
cd: usage: cd [-L|[-P [-e]] [-@]] [dir]  
ubuntu@ubuntu:~$ cp -R /etc/udev Myetc  
ubuntu@ubuntu:~$ ls -l Myetc  
total 4  
drwxr-xr-x 4 ubuntu ubuntu 4096 Jan 24 01:18 udev  
ubuntu@ubuntu:~$ ls -lR Myetc  
Myetc:  
total 4  
drwxr-xr-x 4 ubuntu ubuntu 4096 Jan 24 01:18 udev  
Myetc/udev:  
total 12  
drwxr-xr-x 2 ubuntu ubuntu 4096 Jan 24 01:18 hwdb.d  
drwxr-xr-x 2 ubuntu ubuntu 4096 Jan 24 01:18 rules.d  
-rw-r--r-- 1 ubuntu ubuntu 305 Jan 24 01:18 udev.conf  
Myetc/udev/hwdb.d:  
total 0  
Myetc/udev/rules.d:  
total 84  
-rw-r--r-- 1 ubuntu ubuntu 14476 Jan 24 01:18 70-snap.firefox.rules  
-rw-r--r-- 1 ubuntu ubuntu 63357 Jan 24 01:18 70-snap.snapd.rules  
-rw-r--r-- 1 ubuntu ubuntu 148 Jan 24 01:18 ubuntu--vg-ubuntu--lv.rules  
ubuntu@ubuntu:~$
```

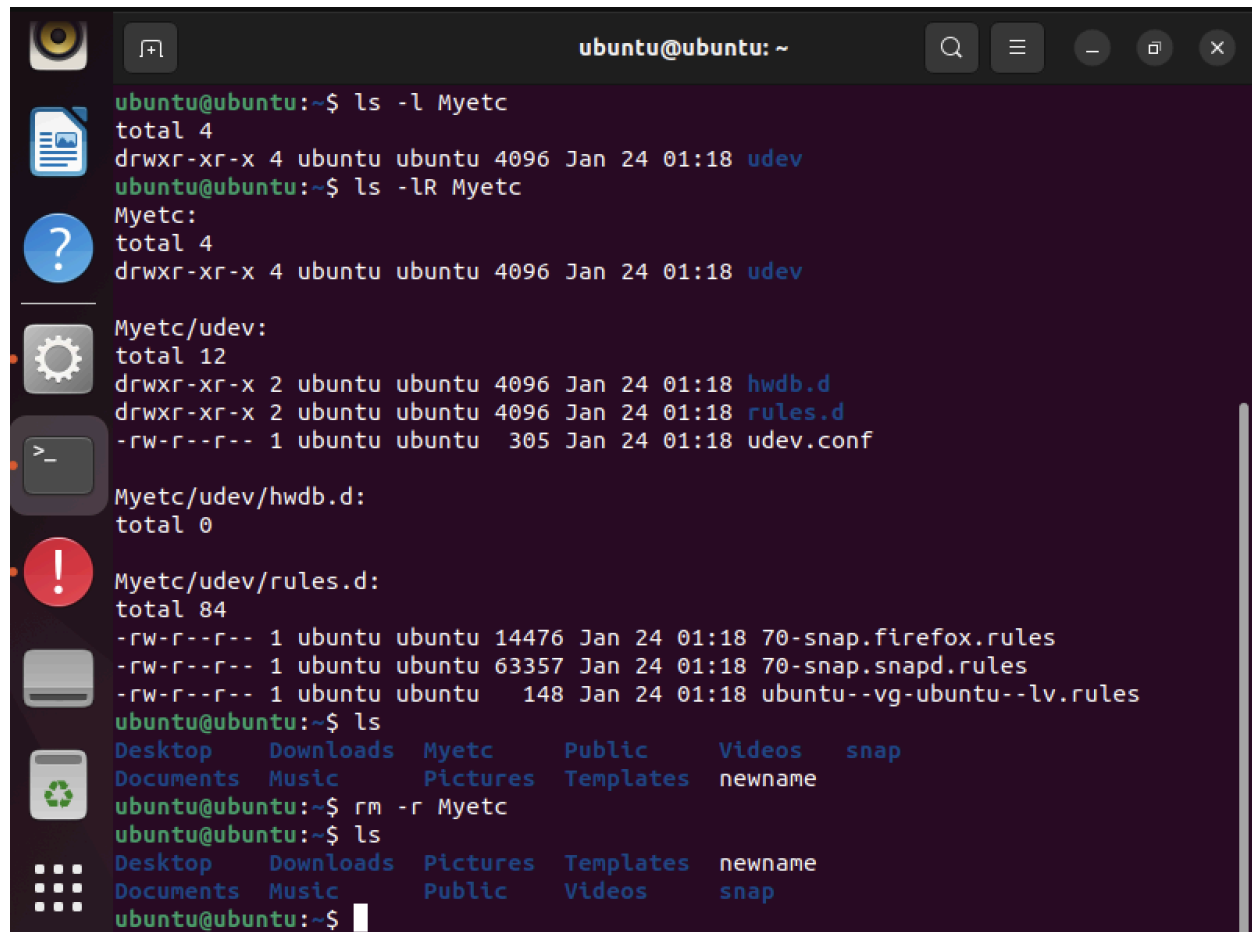
Step 15:

To remove a directory, use the `-r` option to the `rm` command:

```
ls  
rm -r Myetc  
ls
```


Note that the **rmdir** command can also be used to delete directories, but only if the directory is empty (if it contains no files).

Also note the **-r** option. This option removes directories and their contents recursively.



```
ubuntu@ubuntu: ~  
ubuntu@ubuntu:~$ ls -l Myetc  
total 4  
drwxr-xr-x 4 ubuntu ubuntu 4096 Jan 24 01:18 udev  
ubuntu@ubuntu:~$ ls -lR Myetc  
Myetc:  
total 4  
drwxr-xr-x 4 ubuntu ubuntu 4096 Jan 24 01:18 udev  
Myetc/udev:  
total 12  
drwxr-xr-x 2 ubuntu ubuntu 4096 Jan 24 01:18 hwdb.d  
drwxr-xr-x 2 ubuntu ubuntu 4096 Jan 24 01:18 rules.d  
-rw-r--r-- 1 ubuntu ubuntu 305 Jan 24 01:18 udev.conf  
Myetc/udev/hwdb.d:  
total 0  
Myetc/udev/rules.d:  
total 84  
-rw-r--r-- 1 ubuntu ubuntu 14476 Jan 24 01:18 70-snap.firefox.rules  
-rw-r--r-- 1 ubuntu ubuntu 63357 Jan 24 01:18 70-snap.snapd.rules  
-rw-r--r-- 1 ubuntu ubuntu 148 Jan 24 01:18 ubuntu--vg-ubuntu--lv.rules  
ubuntu@ubuntu:~$ ls  
Desktop Downloads Myetc Public Videos snap  
Documents Music Pictures Templates newname  
ubuntu@ubuntu:~$ rm -r Myetc  
ubuntu@ubuntu:~$ ls  
Desktop Downloads Pictures Templates newname  
Documents Music Public Videos snap  
ubuntu@ubuntu:~$
```

Step 16:

Moving a file is analogous to a **"cut and paste"**.

The file is "cut" (removed) from the original location and "pasted" to the specified destination. Move a file in the local directory by executing the following commands:

```
touch premove  
ls  
mv premove postmove
```

```
ls
rm postmove
```

Linux Command	Description
<code>touch premove</code>	Creates an empty file called premove
<code>mv premove postmove</code>	This command “cuts” the premove file and “pastes” it to a file called postmove
<code>rm postmove</code>	Removes postmove file

```
ubuntu@ubuntu: ~
-rw-r--r-- 1 ubuntu ubuntu 305 Jan 24 01:18 udev.conf

Myetc/udev/hwdb.d:
total 0

Myetc/udev/rules.d:
total 84
-rw-r--r-- 1 ubuntu ubuntu 14476 Jan 24 01:18 70-snap.firefox.rules
-rw-r--r-- 1 ubuntu ubuntu 63357 Jan 24 01:18 70-snap.snapd.rules
-rw-r--r-- 1 ubuntu ubuntu 148 Jan 24 01:18 ubuntu--vg-ubuntu--lv.rules
ubuntu@ubuntu:~$ ls
Desktop  Downloads  Myetc      Public     Videos    snap
Documents Music      Pictures   Templates  newname
ubuntu@ubuntu:~$ rm -r Myetc
ubuntu@ubuntu:~$ ls
Desktop  Downloads  Pictures   Templates  newname
Documents Music      Public     Videos    snap
ubuntu@ubuntu:~$ touch premove
ubuntu@ubuntu:~$ ls
Desktop  Downloads  Pictures   Templates  newname  snap
Documents Music      Public     Videos    premove
ubuntu@ubuntu:~$ mv premove postmove
ubuntu@ubuntu:~$ ls
Desktop  Downloads  Pictures   Templates  newname  snap
Documents Music      Public     Videos    postmove
ubuntu@ubuntu:~$ rm postmove
ubuntu@ubuntu:~$ ls
Desktop  Downloads  Pictures   Templates  newname
Documents Music      Public     Videos    snap
ubuntu@ubuntu:~$
```

Remember:

You are required to complete the lab and record your answers. Rename the file using your first name and the lab number, e.g. ***washington3.2.docx***, and submit it through Canvas to receive marks for this lab.