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.

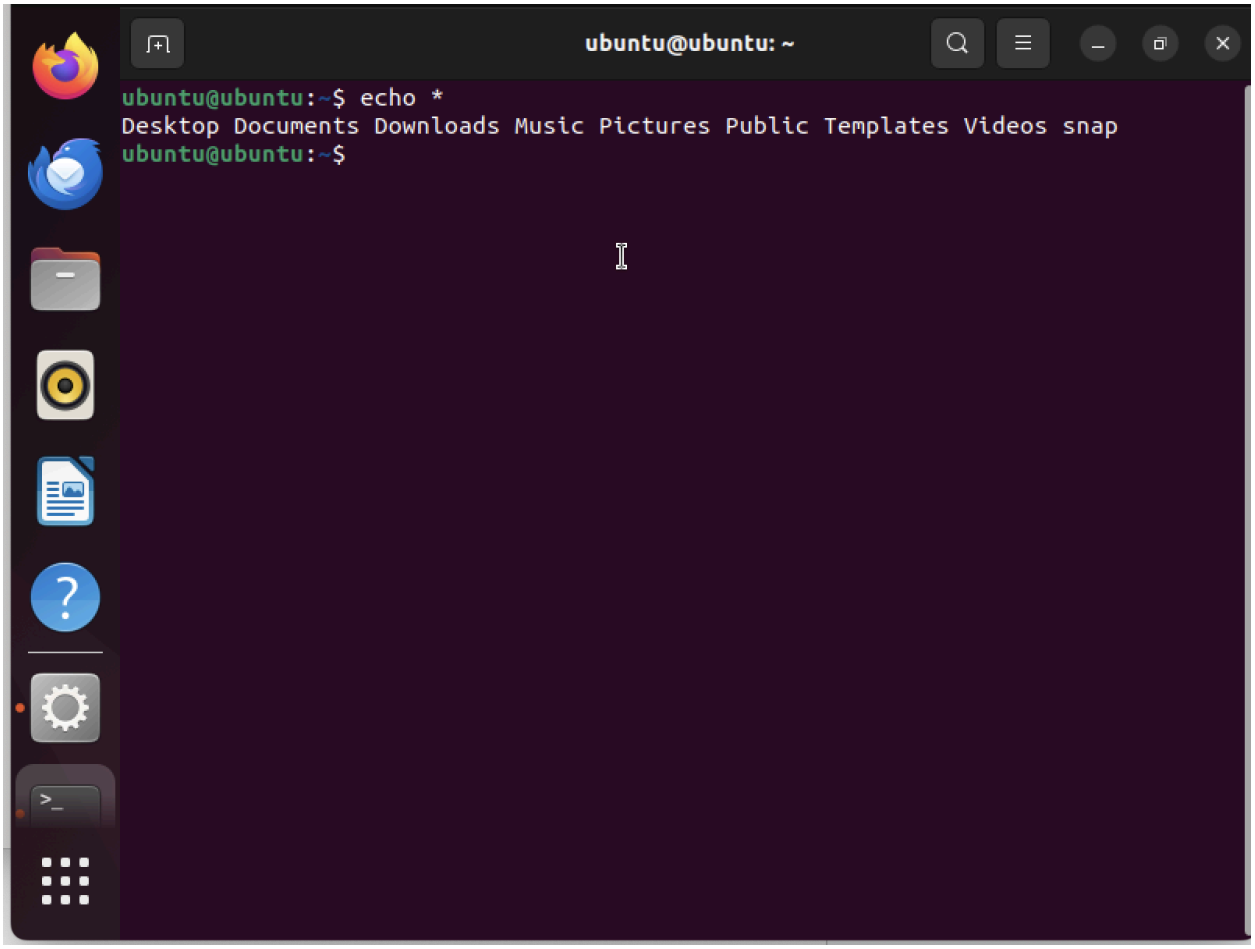
Step 1:

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.



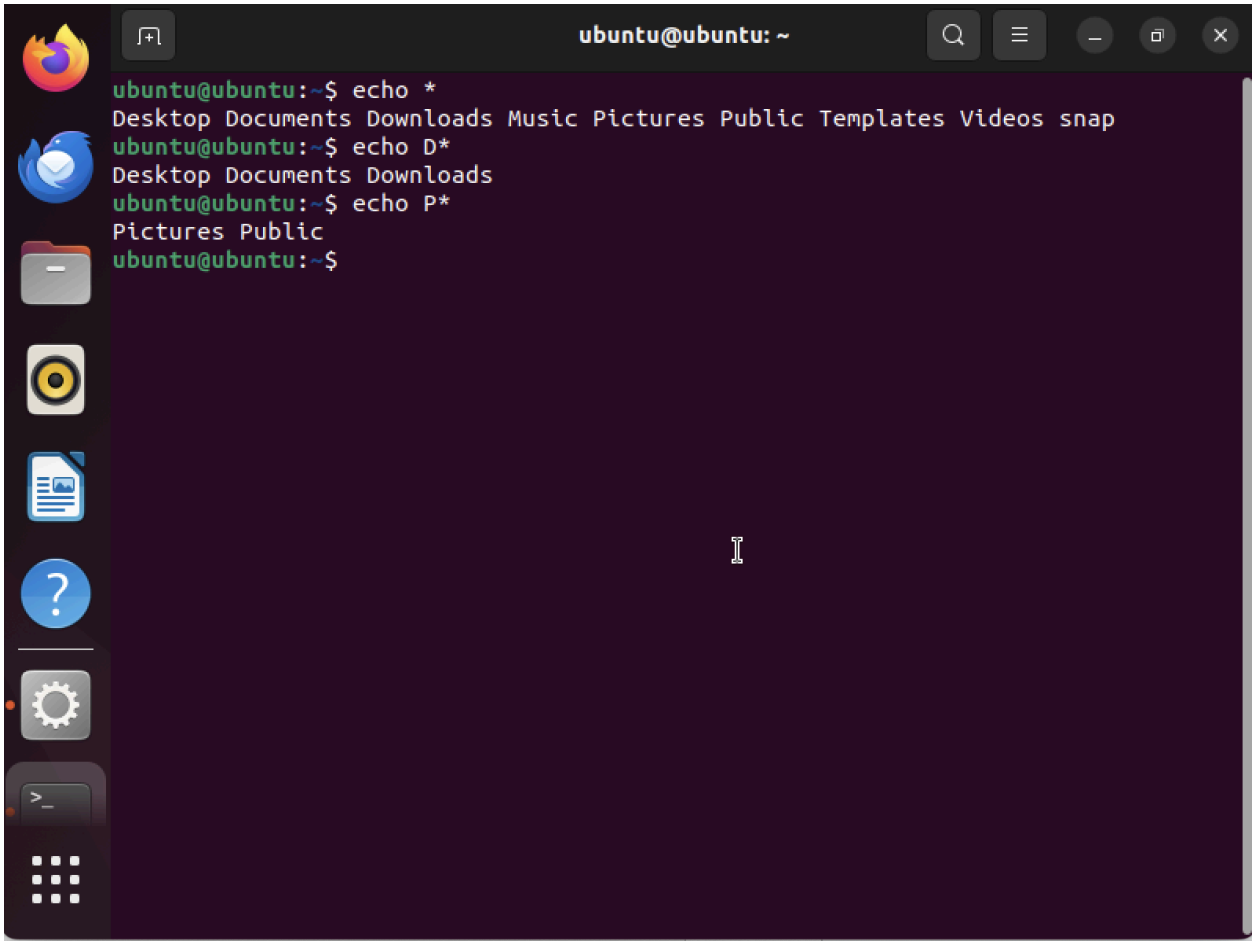
Step 2:

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*
```

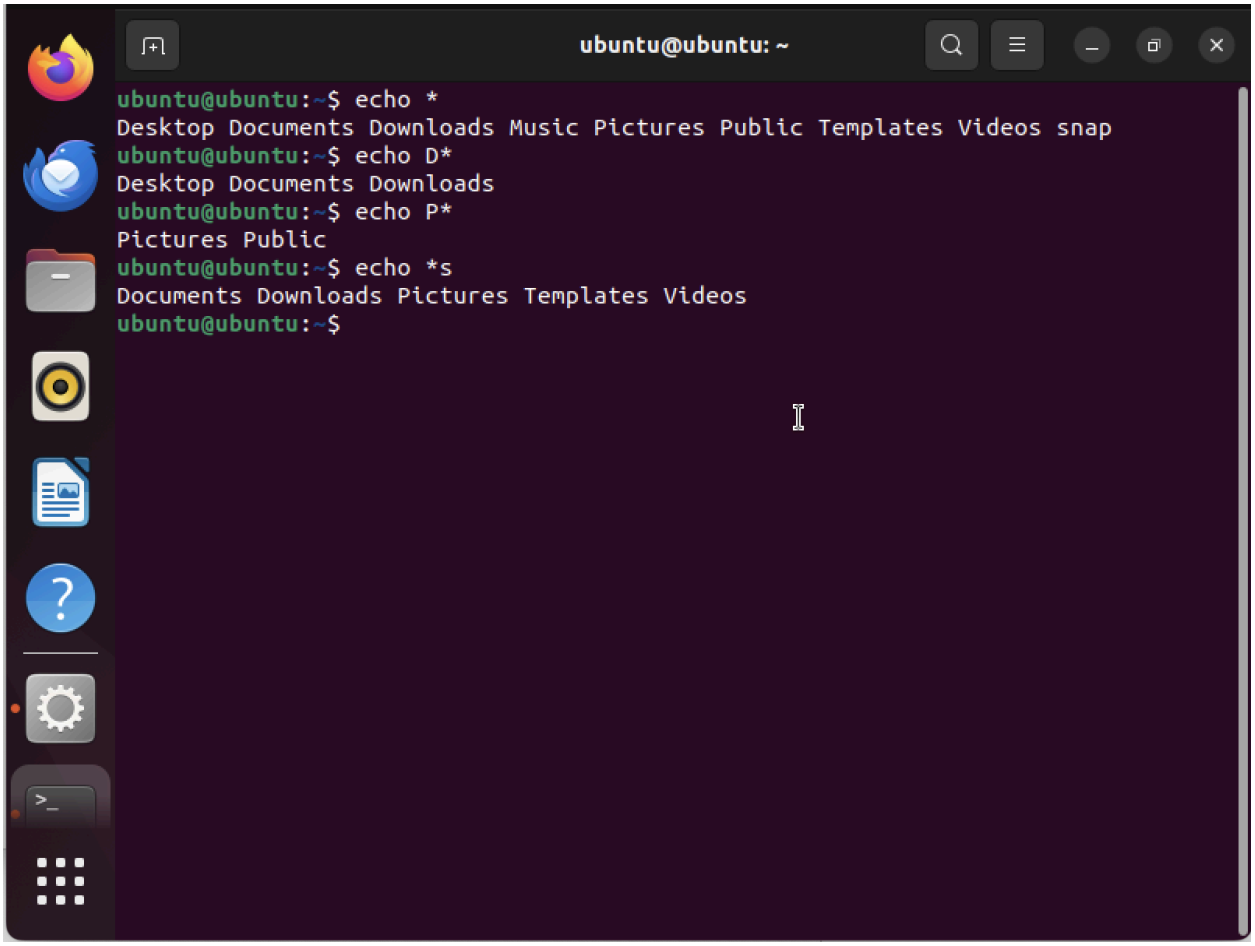
Think of the first example, **D***, as "match all filenames in the current directory that begin with a capital d character and have zero or more of any other character after the **D**".



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:

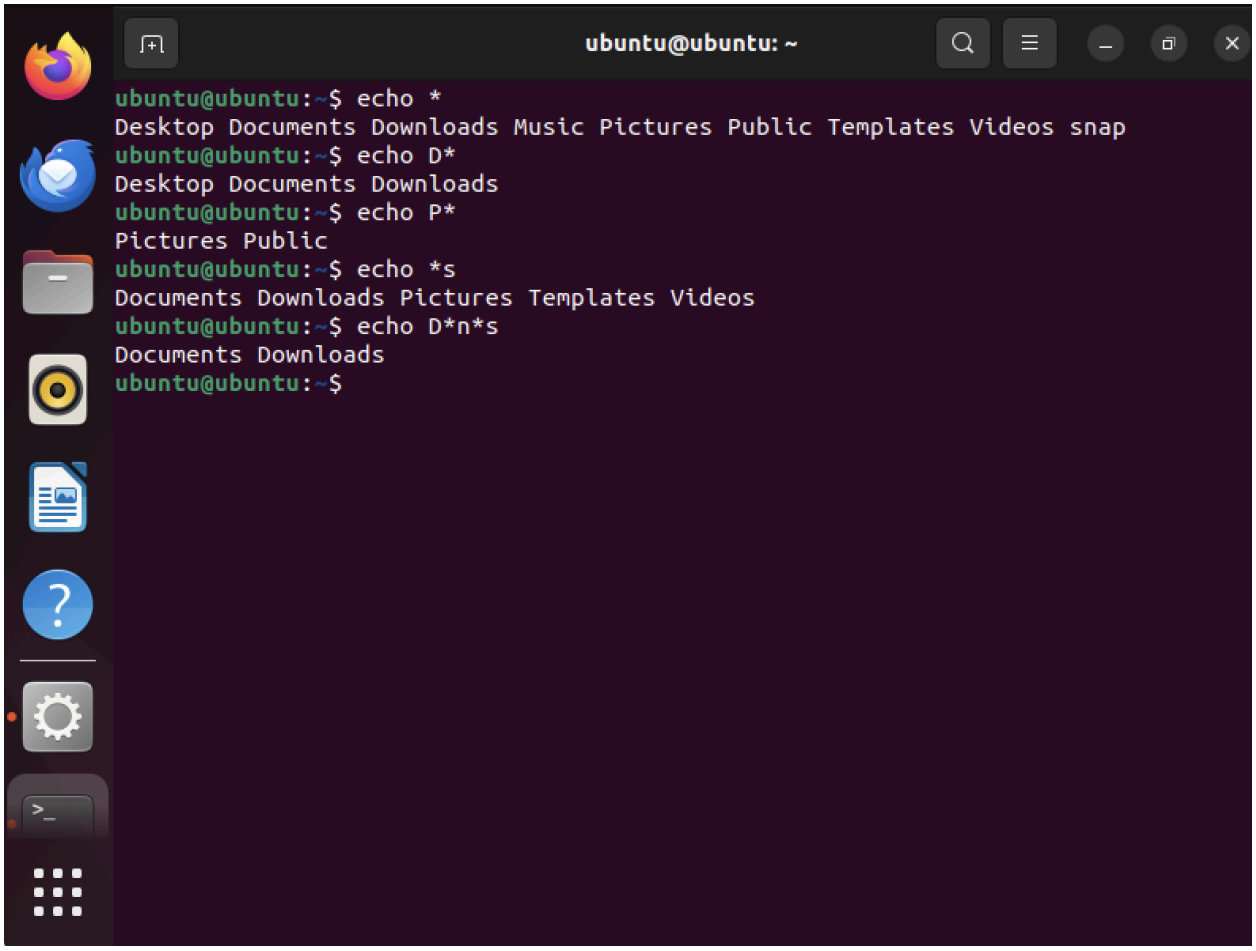
```
echo *s
```



Step 4:

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

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



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.

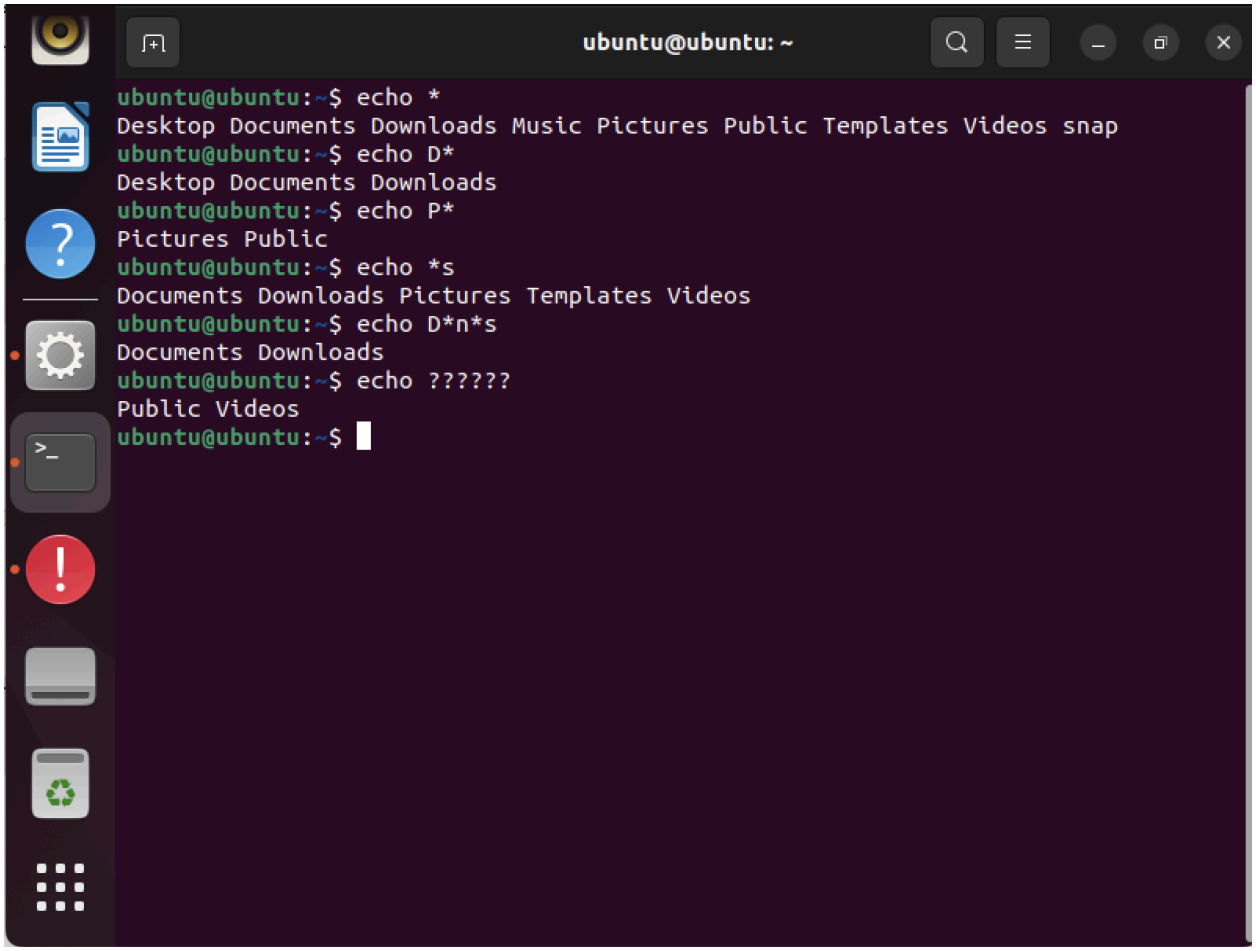
Since each question mark matches one unknown character, typing six of them will match six-character filenames.

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

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

Important

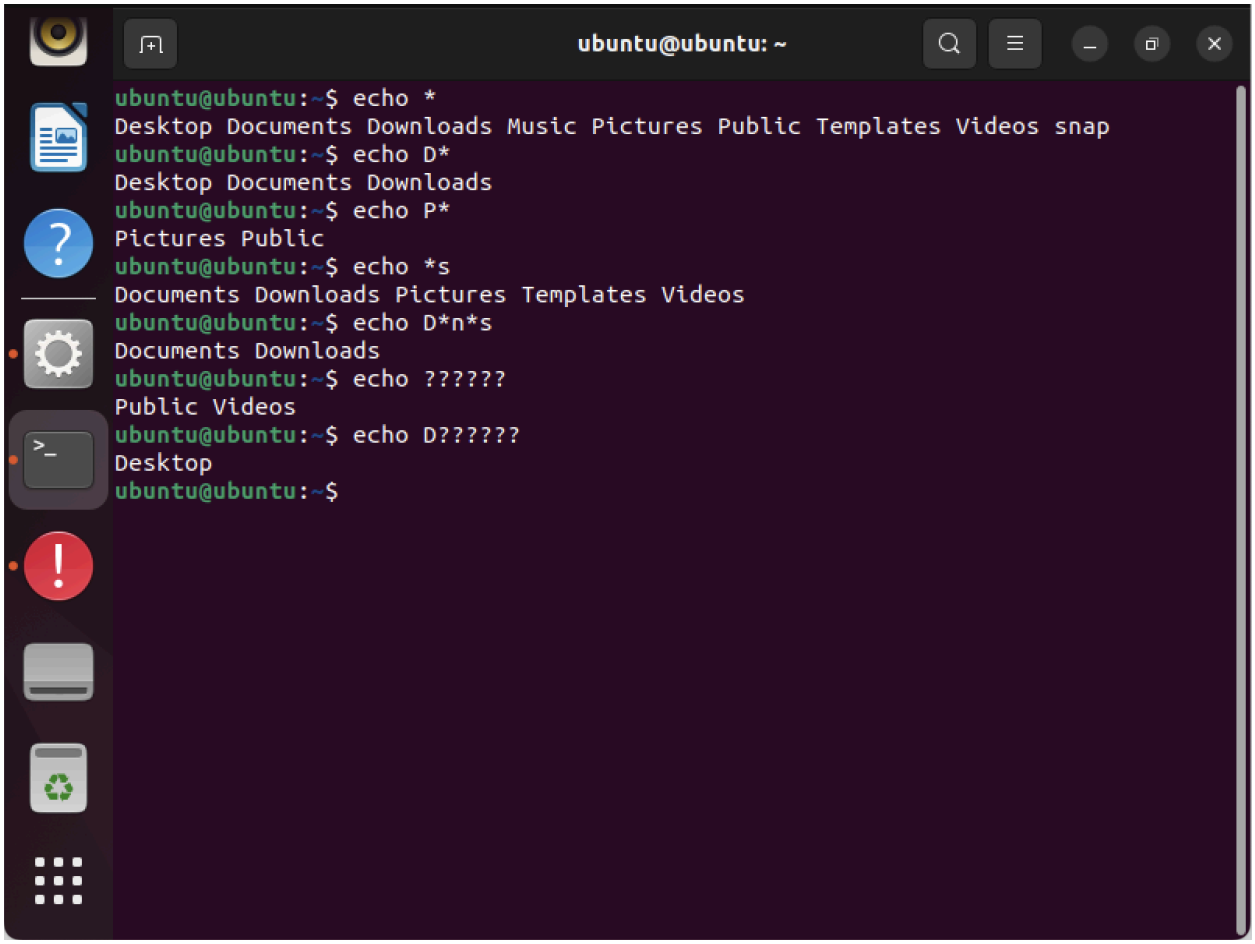
Each ? character must match exactly one character in a filename-no more and no less than one character.



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????????
```

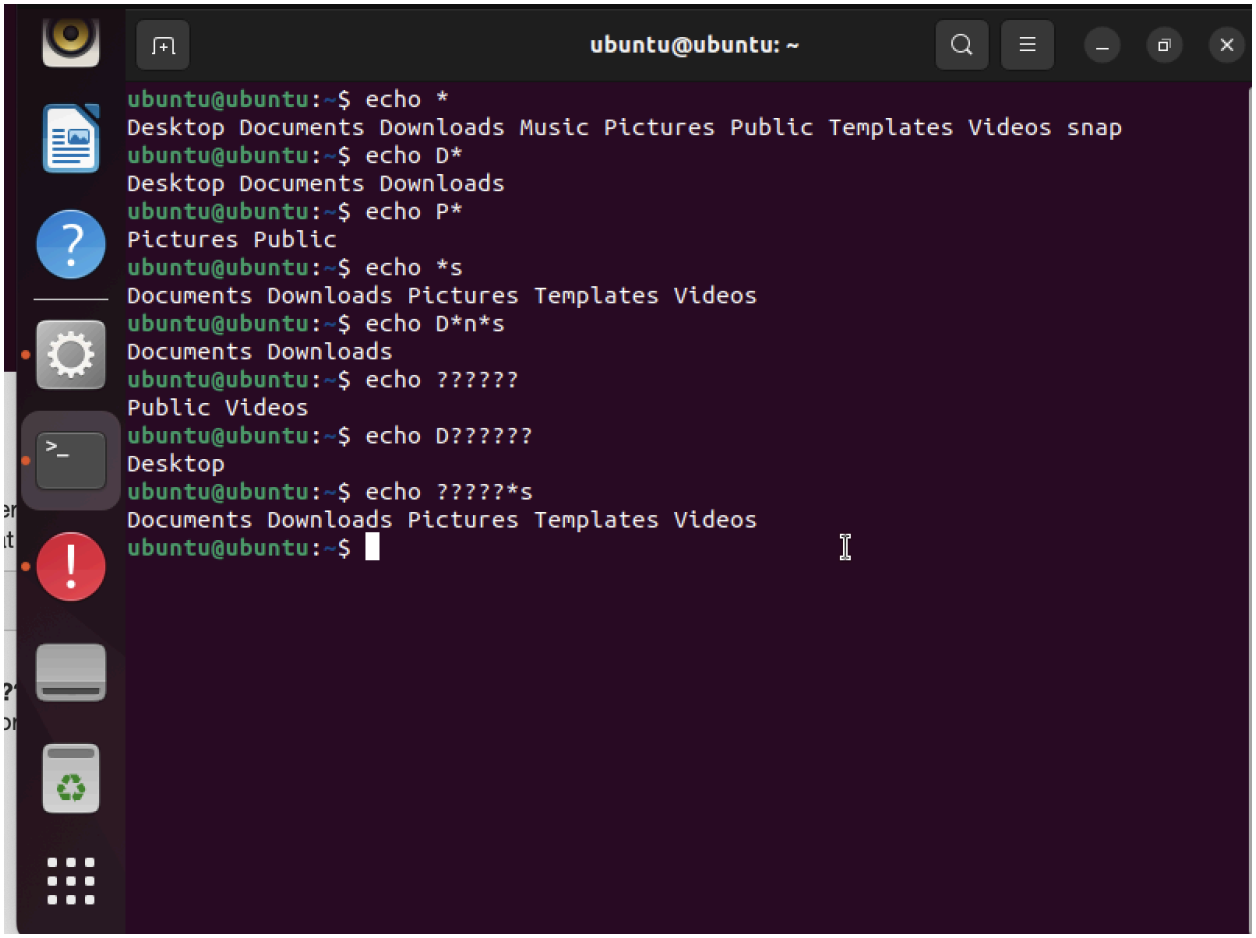


Step 7:

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".



Step 8:

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

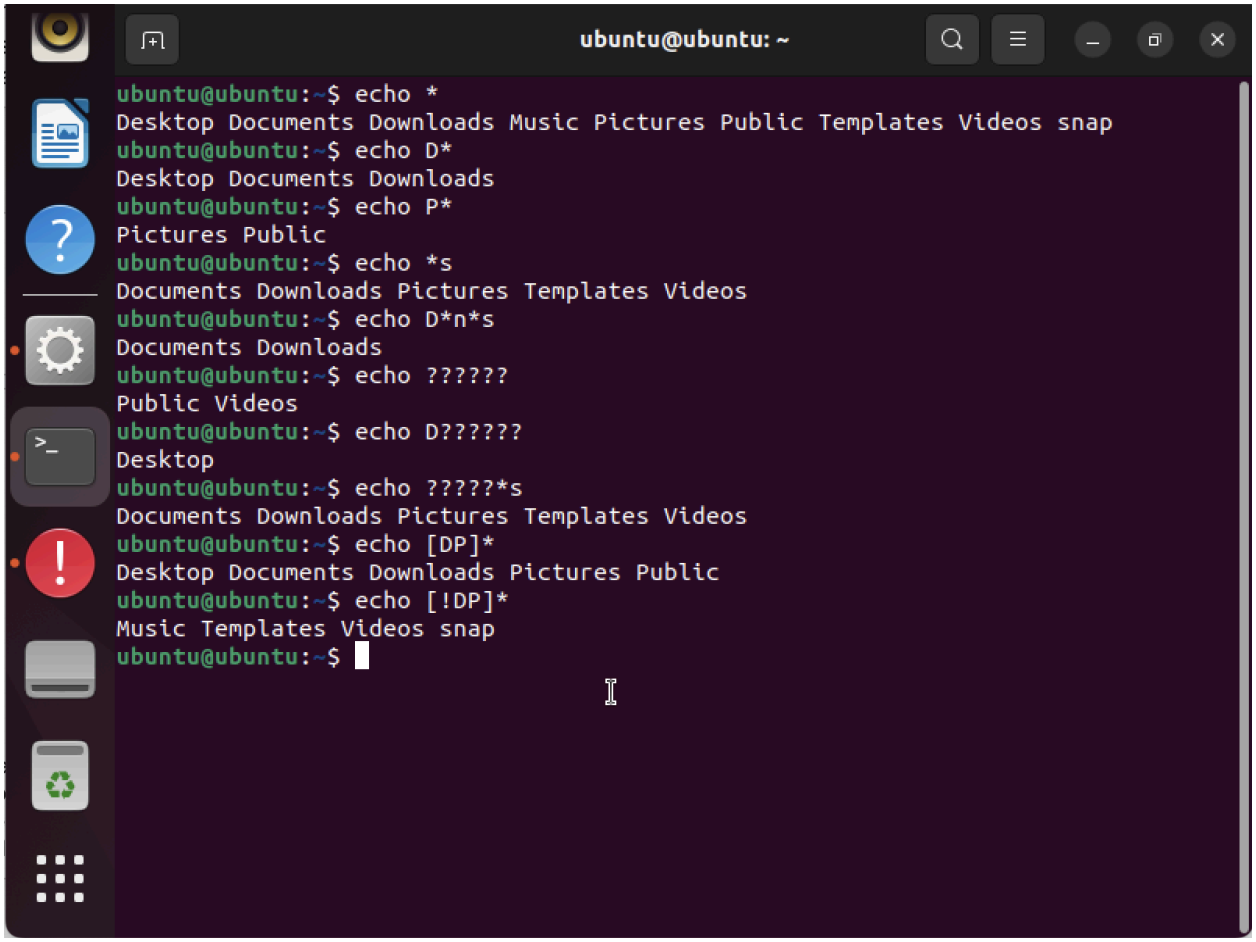
This glob uses a pair of square **brackets [ ]** to specify which one character will be allowed. The allowed characters can be specified as a range, a list, or by what is known as a character class.

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]*
```



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`:

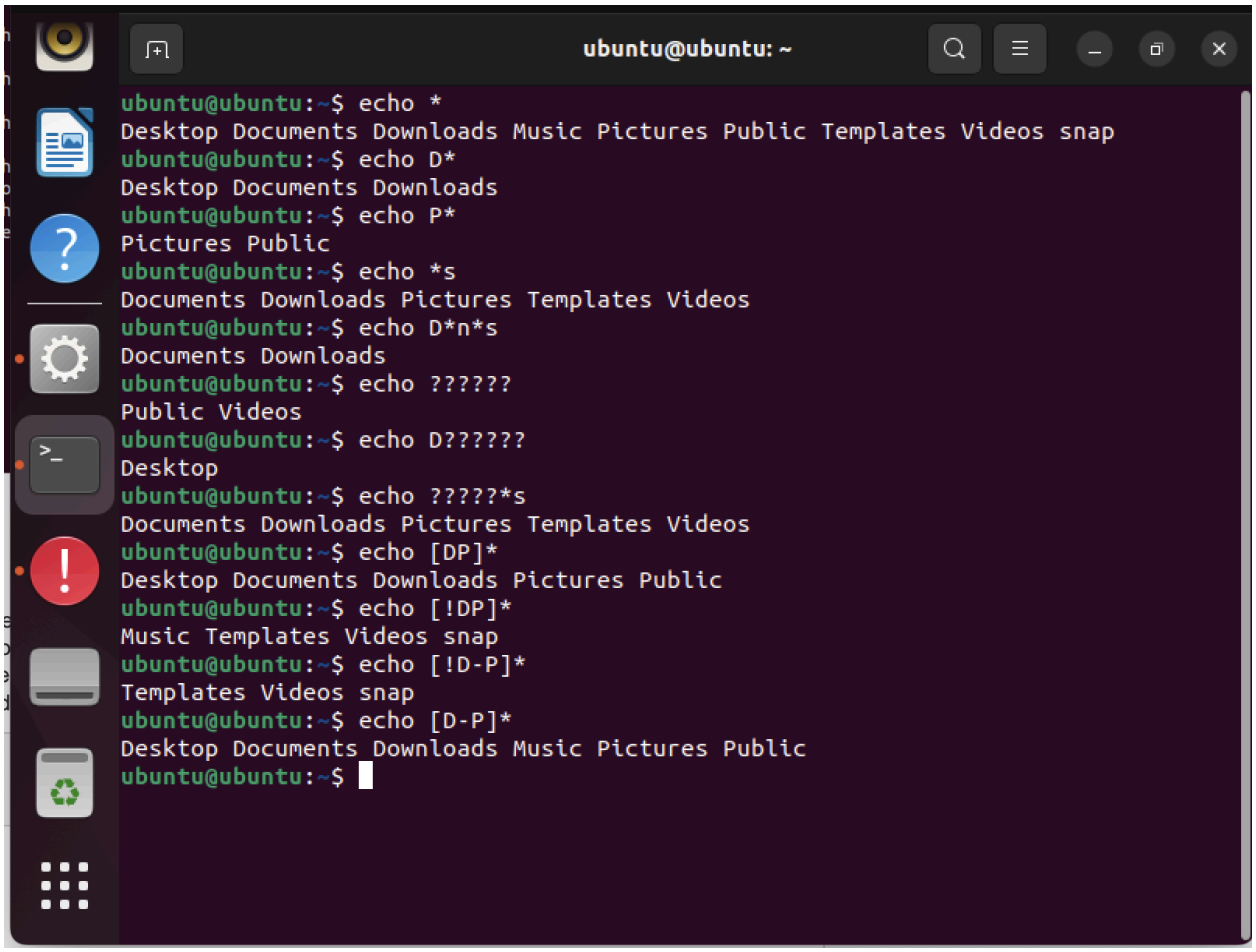
```
echo [D-P]*

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**.



Step 10:

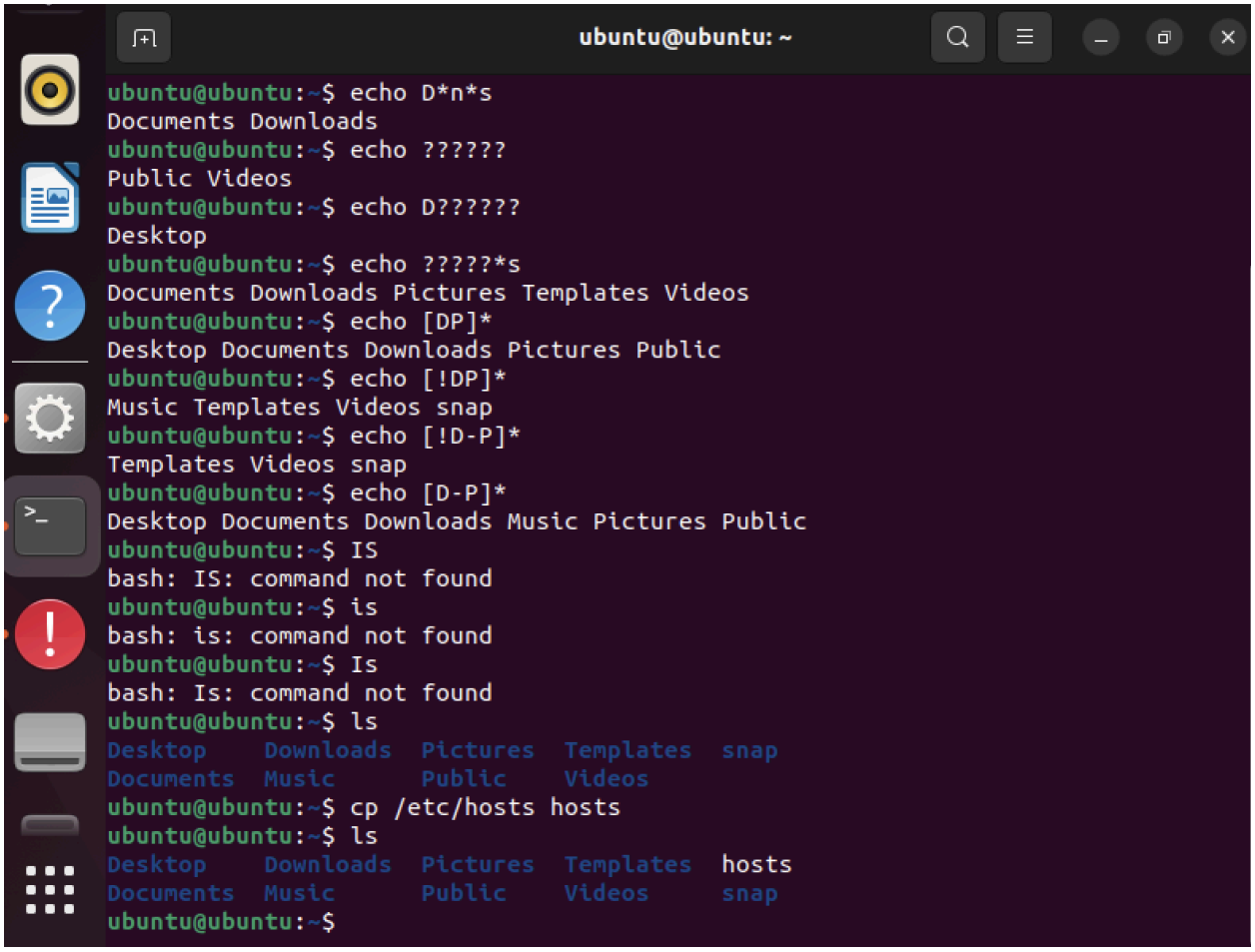
Make a copy of the /etc/hosts file and place it in the current directory. Then, list the contents of the current directory before and after the copy:

```
ls

cp /etc/hosts hosts

ls
```

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



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.

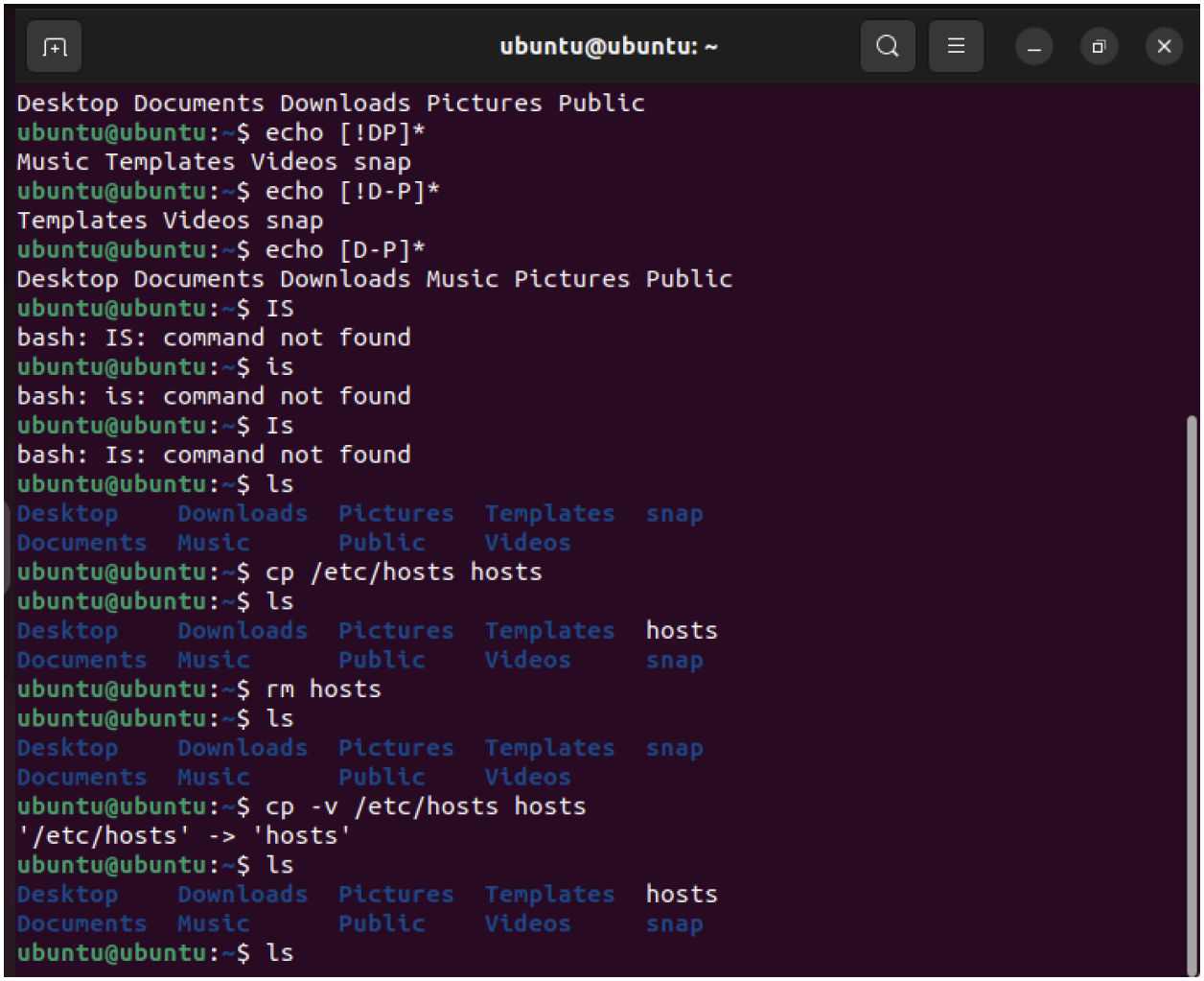
Note that the `-v` switch displays the source and target when the `cp` command is executed:

Source

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

Target

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



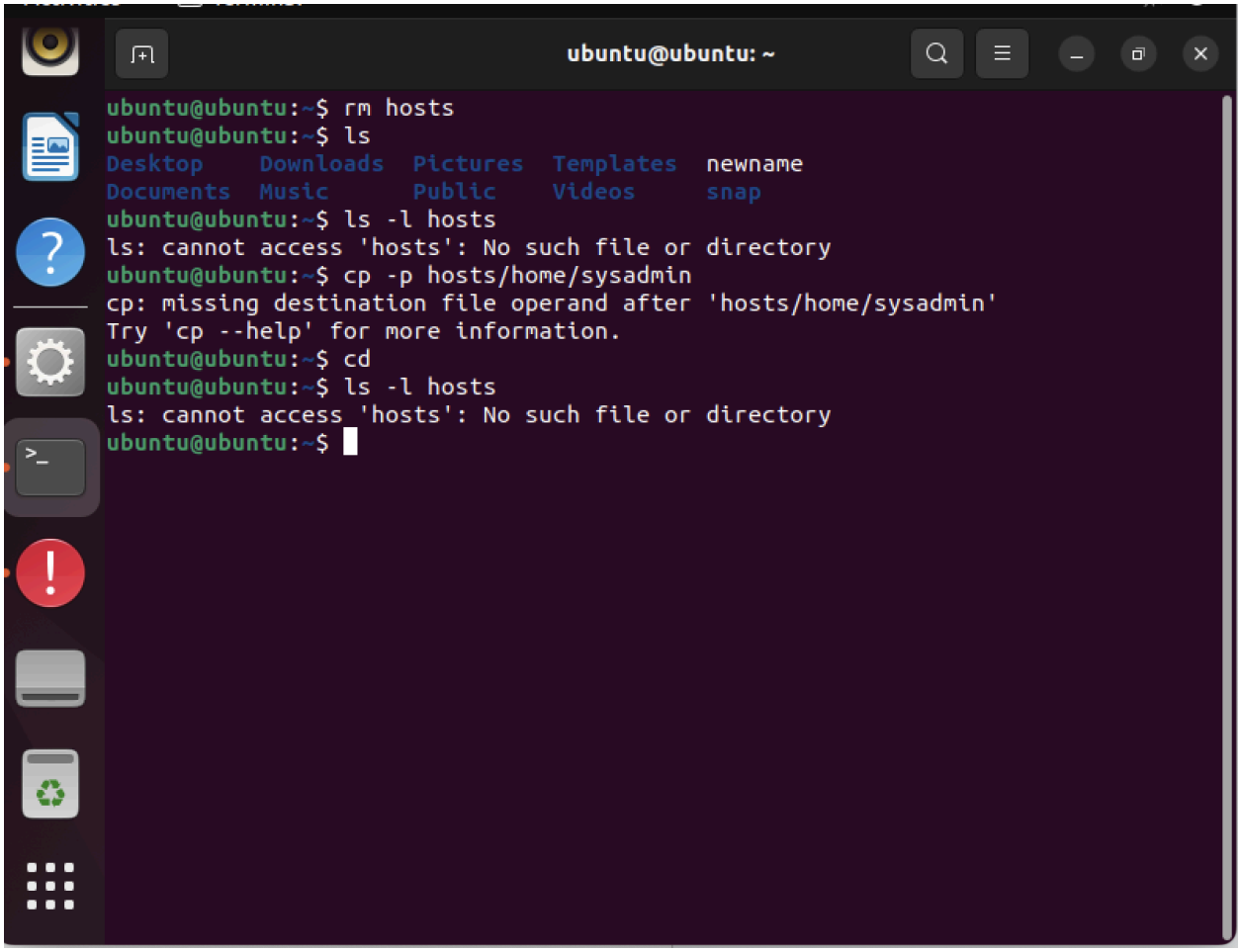
Step 12:

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.



Step 13:

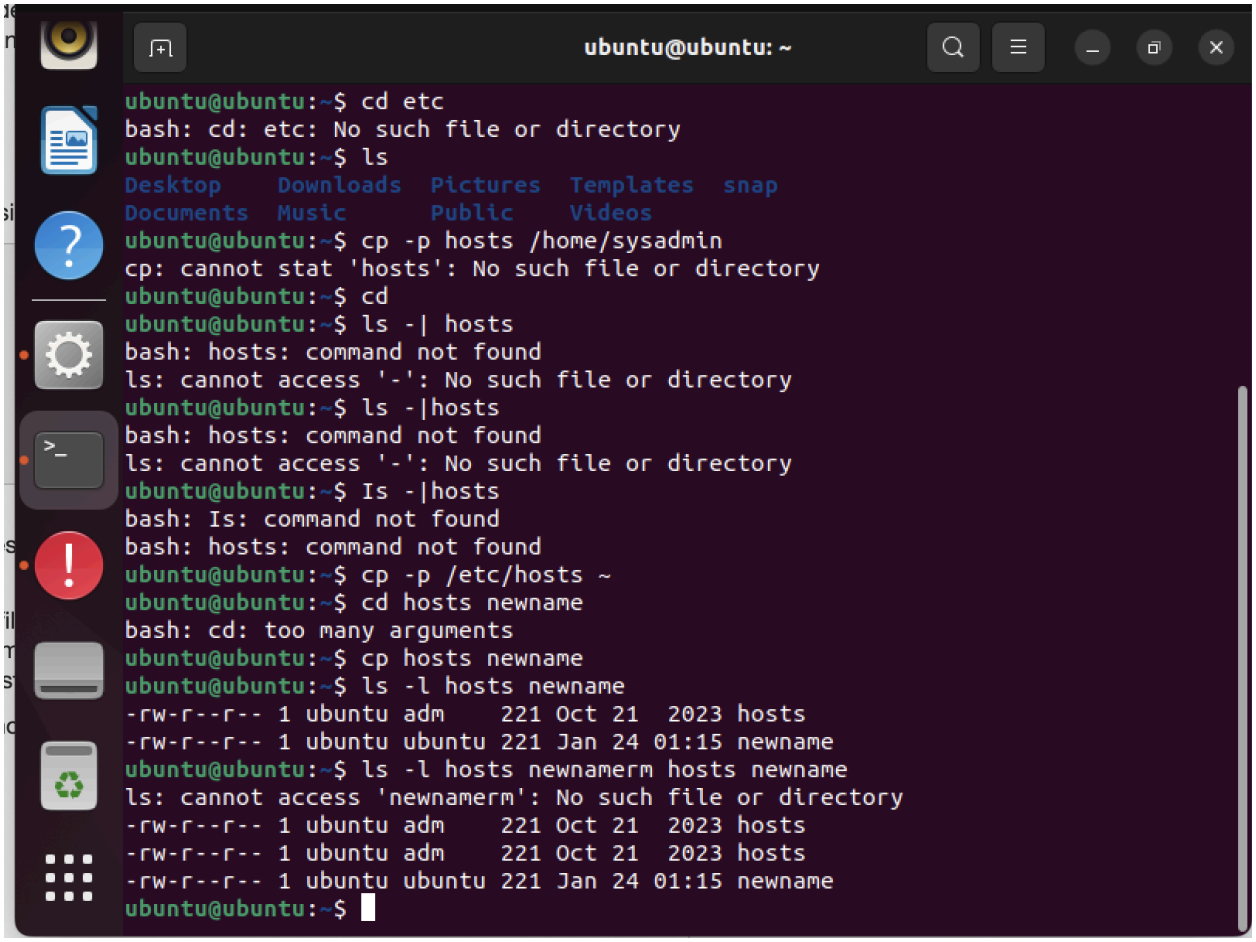
Type the following commands to copy using a different target name:

```
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.



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.

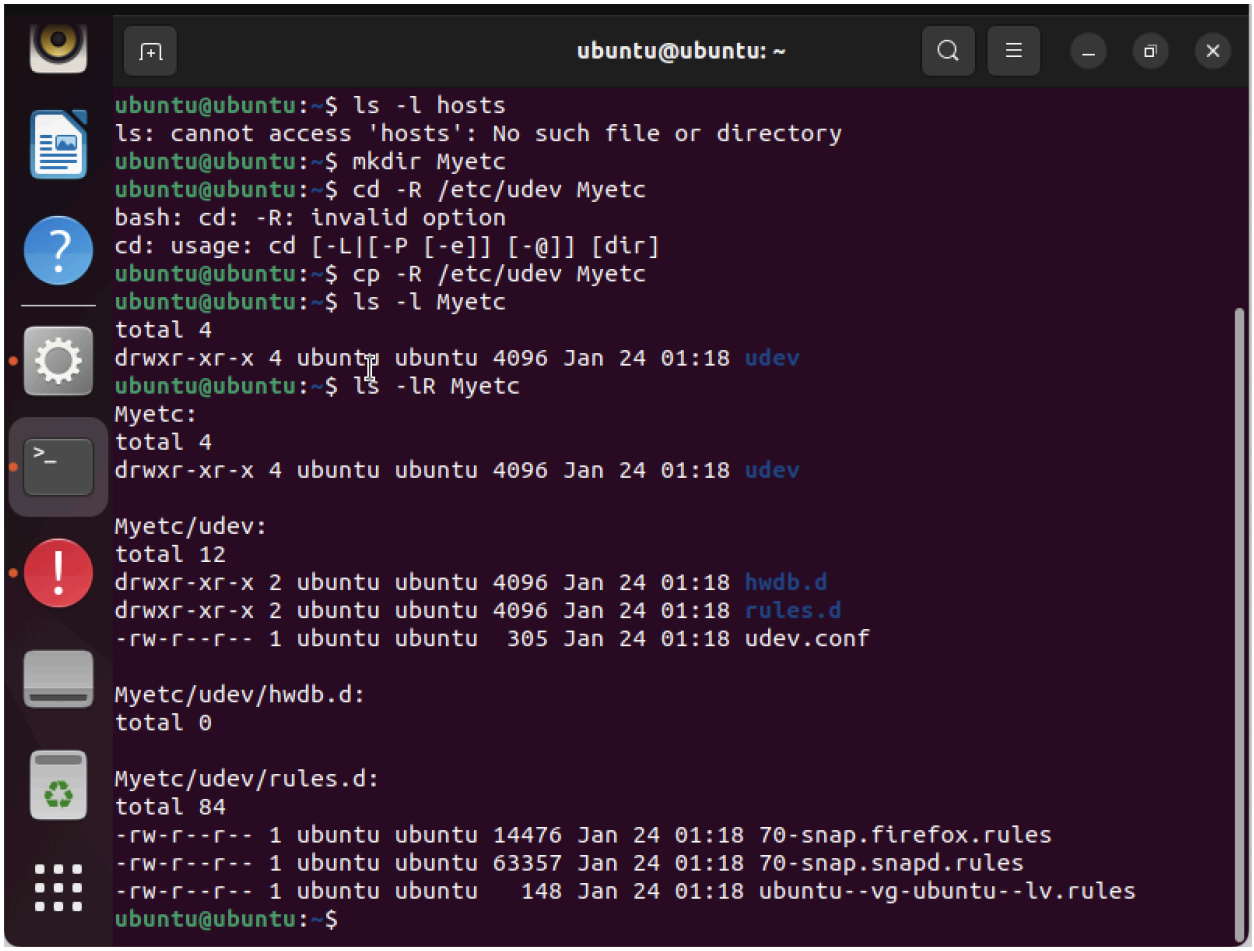
Options are available for the mkdir command to set security, permissions and other attributes of a new directory. Once the directory has been copied, the ls command is used to list the contents of the directory with both the long and recursive options.

```
mkdir Myetc

cp –R /etc/udev Myetc

ls –l Myetc
```

```
ls –lR Myetc
```



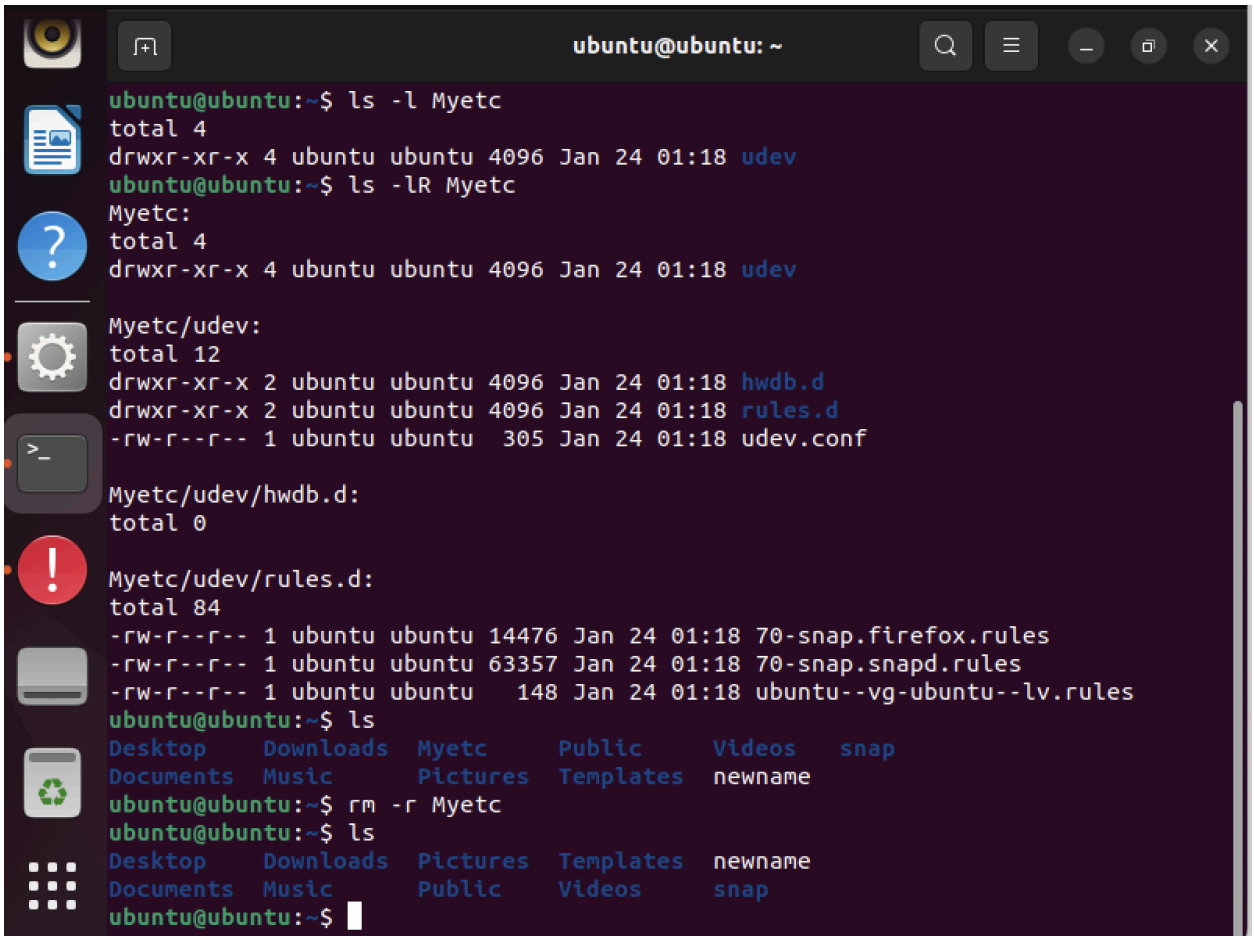
Step 15:

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.



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:

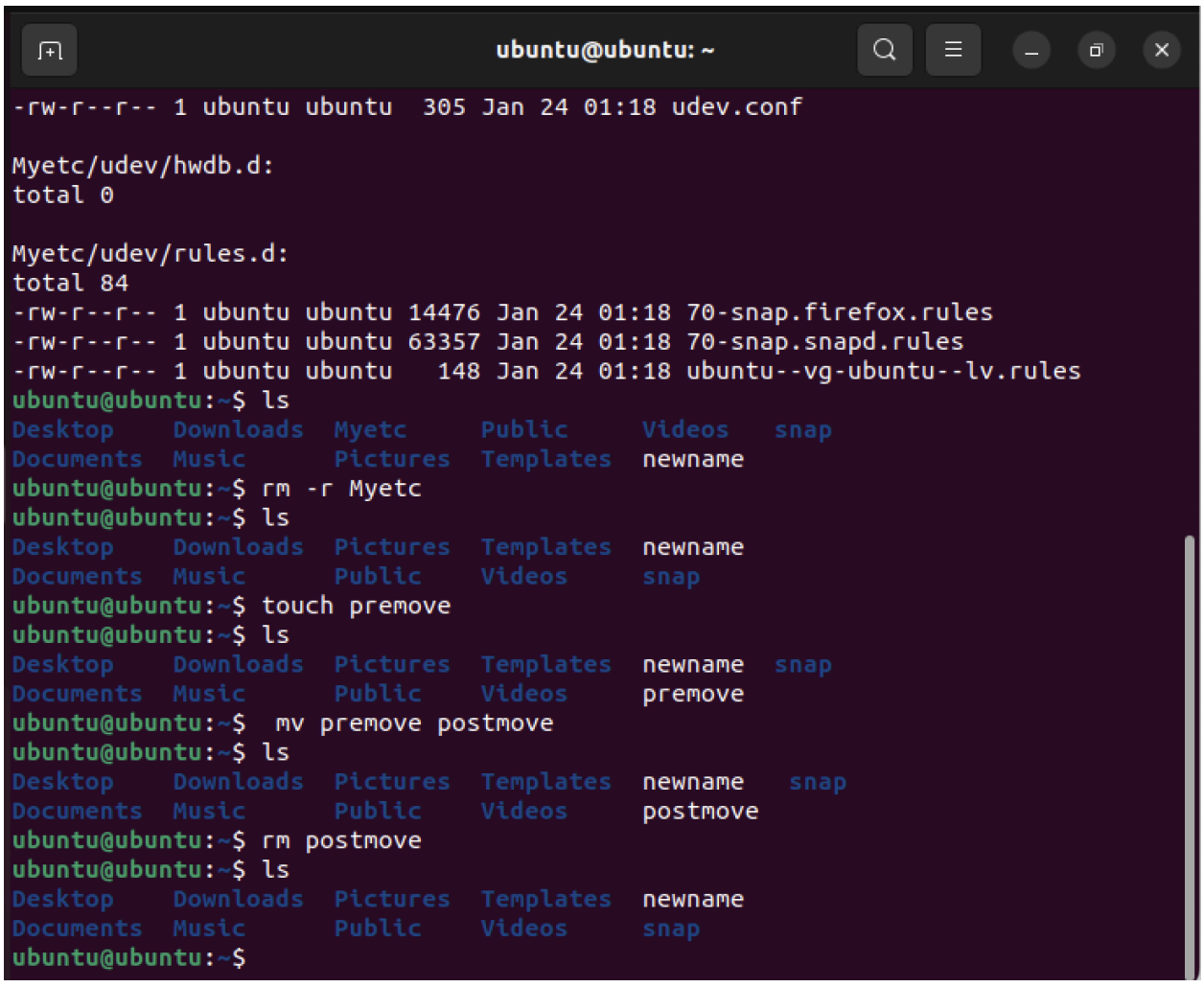
```
touch premove

ls

mv premove postmove
```

```
ls
rm postmove
```

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



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.