

PRACTICAL 1

AIM: Demonstration of Basic Linux commands

Command shell: A program that interprets commands is Command shell.

Shell Script: Allows a user to execute commands by typing them manually at a terminal, or automatically in programs called shell scripts.

A shell is not an operating system .It is a way to interface with the operating system and run Commands.

BASH (Bourne Again Shell)

- Bash is a shell written as a free replacement to the standard Bourne Shell(/bin/sh) originally written by Steve Bourne for UNIX systems
- It has all of the features of the original Bourne Shell,plus additions that make it easier to program with and use from the command line.
- Since it is Free Software, it has been adopted as the default shell on most Linux systems.

BASIC LINUX COMMANDS:

1. pwd : Print Working Directory

DESCRIPTION:

Pwd prints the full path name of the current working directory.

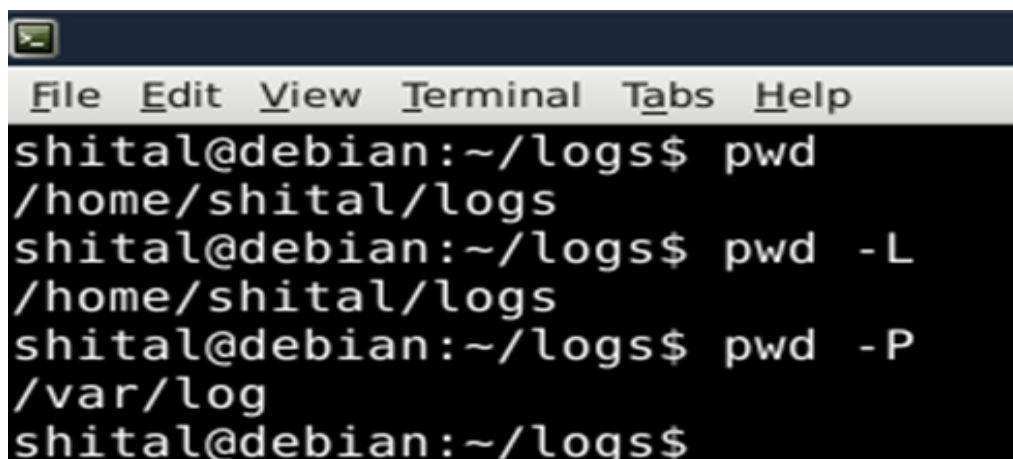
SYNTAX:

pwd

EXAMPLE:

\$pwd

/home/directory_name



```
File Edit View Terminal Tabs Help
shital@debian:~/logs$ pwd
/home/shital/logs
shital@debian:~/logs$ pwd -L
/home/shital/logs
shital@debian:~/logs$ pwd -P
/var/log
shital@debian:~/logs$
```

2. **cd:** Change Directory

DESCRIPTION:

It allows you to change your working directory. You use it to move around within the hierarchy your file system

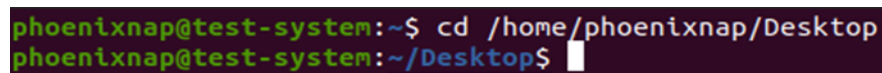
SYNTAX:

`cd directory_name`

EXAMPLE:

To change into “work directory” in “documents” you need to write as follows.

Input: `$cd/documents/work`

A terminal window with a dark purple background. The first line shows the prompt 'phoenixnap@test-system:~\$' followed by the command 'cd /home/phoenixnap/Desktop'. The second line shows the prompt 'phoenixnap@test-system:~/Desktop\$' with a white cursor. A red underline is drawn under the path '~/Desktop', and a red arrow points from the underline to the text.

3. **cd..**

DESCRIPTION:

Move up one directory.

SYNTAX:

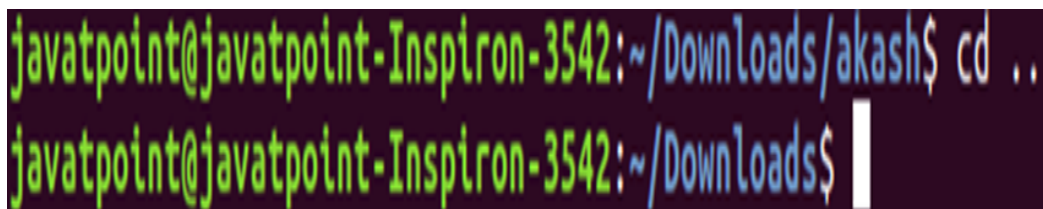
`cd..`

EXAMPLE:

If you are in work directory and want to go to documents then write

`cd..`

You will end up in /documents.

A terminal window with a dark purple background. The first line shows the prompt 'jvatpoint@jvatpoint-Inspiron-3542:~/Downloads/akash\$' followed by the command 'cd ..'. The second line shows the prompt 'jvatpoint@jvatpoint-Inspiron-3542:~/Downloads\$' with a white cursor.

4. ls : list all the Files and Directories

DESCRIPTION:

List all files and folders in the current directory in the column format.

SYNTAX:

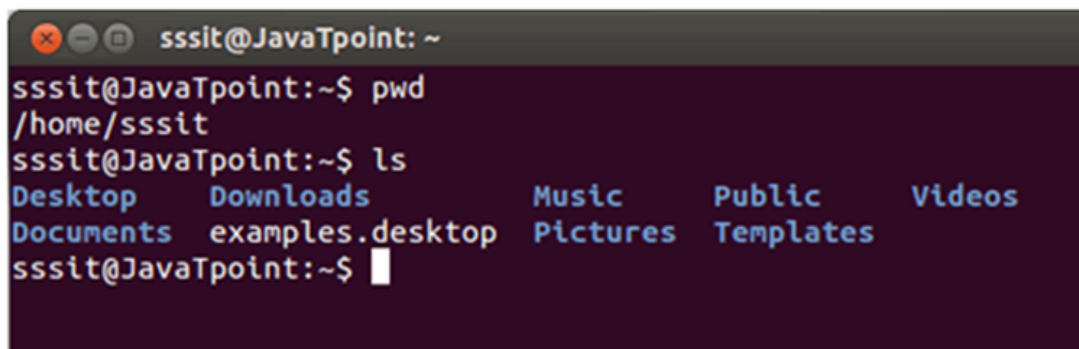
`ls[options]`

EXAMPLE:

Using various options

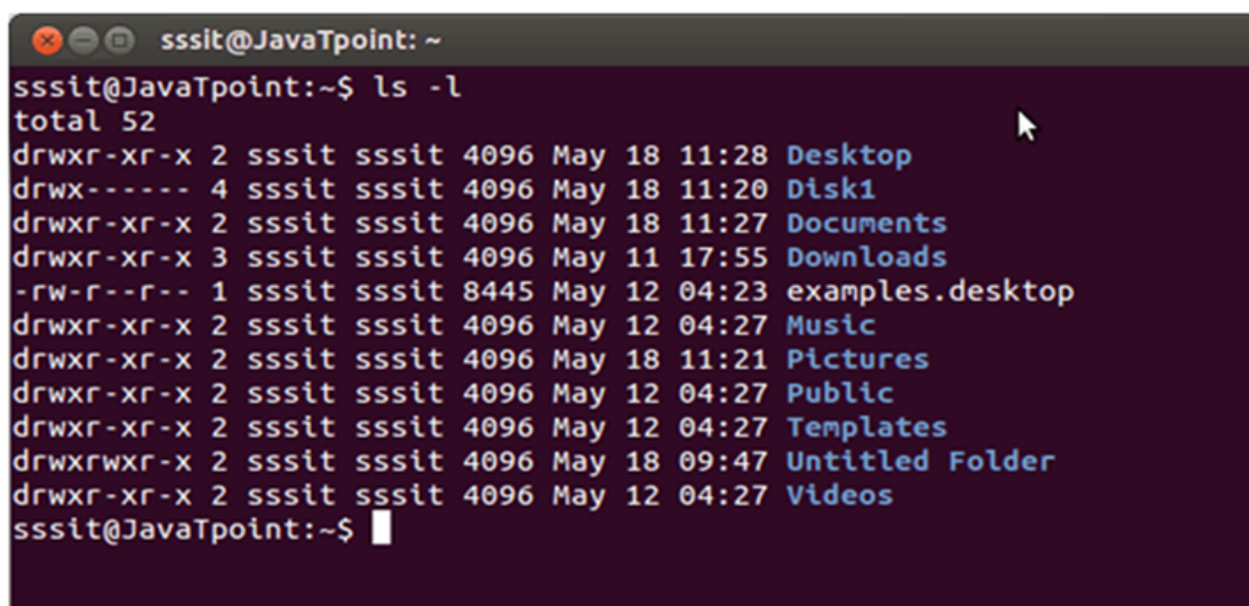
- Lists the total files in the directory and sub directories, the names of the files in the current directory, their permissions, the number of sub directories in directories listed, the size of the file, and the date of last modification.

`ls-l`



```
sssit@JavaTpoint: ~  
sssit@JavaTpoint:~$ pwd  
/home/sssit  
sssit@JavaTpoint:~$ ls  
Desktop      Downloads      Music      Public      Videos  
Documents    examples.desktop  Pictures    Templates  
sssit@JavaTpoint:~$
```

- List all files including hidden files
 - `ls-a`



```
sssit@JavaTpoint: ~  
sssit@JavaTpoint:~$ ls -l  
total 52  
drwxr-xr-x 2 sssit sssit 4096 May 18 11:28 Desktop  
drwx----- 4 sssit sssit 4096 May 18 11:20 Disk1  
drwxr-xr-x 2 sssit sssit 4096 May 18 11:27 Documents  
drwxr-xr-x 3 sssit sssit 4096 May 11 17:55 Downloads  
-rw-r--r-- 1 sssit sssit 8445 May 12 04:23 examples.desktop  
drwxr-xr-x 2 sssit sssit 4096 May 12 04:27 Music  
drwxr-xr-x 2 sssit sssit 4096 May 18 11:21 Pictures  
drwxr-xr-x 2 sssit sssit 4096 May 12 04:27 Public  
drwxr-xr-x 2 sssit sssit 4096 May 12 04:27 Templates  
drwxrwxr-x 2 sssit sssit 4096 May 18 09:47 Untitled Folder  
drwxr-xr-x 2 sssit sssit 4096 May 12 04:27 Videos  
sssit@JavaTpoint:~$
```

5. cat

DESCRIPTION:

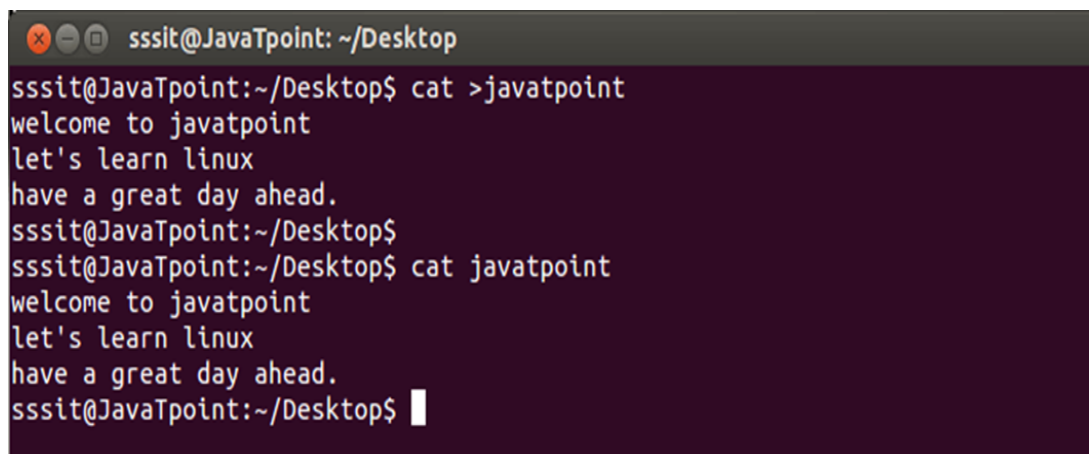
cat stands for "catenate". It Reads Data From Files, and outputs their contents. It is the simplest way to To display the contents of a file at the command line.

SYNTAX:

cat filename

EXAMPLES:

- Print the contents of files *mytext.txt* and *yourtext.txt*
 - *cat mytext.txt yourtext.txt*
- Print the cpu information using *cat* command
 - *cat /proc/cpuinfo*
- Print the memory information using *cat* command
 - *cat/proc/meminfo*



```
sssit@JavaTpoint: ~/Desktop
sssit@JavaTpoint:~/Desktop$ cat >javatpoint
welcome to javatpoint
let's learn linux
have a great day ahead.
sssit@JavaTpoint:~/Desktop$
sssit@JavaTpoint:~/Desktop$ cat javatpoint
welcome to javatpoint
let's learn linux
have a great day ahead.
sssit@JavaTpoint:~/Desktop$
```

6. head

DESCRIPTION:

head, by default, prints the first 10 lines of each FILE to standard output. With more than one FILE, it precedes each set of output with a header identifying the file name.

If no FILE is specified, or when FILE is specified as a dash ("-"), *head* reads from standard input.

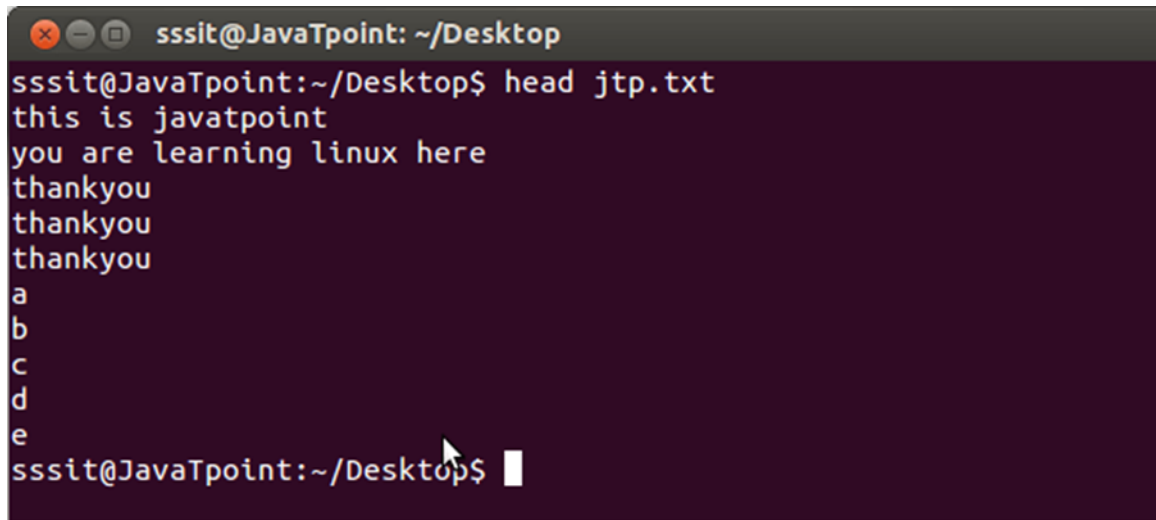
SYNTAX:

head[option]...[file/directory]

EXAMPLE:

Display the first ten lines of myfile.txt.

headmyfile.txt

A terminal window with a dark purple background. The title bar shows 'sssit@JavaTpoint: ~/Desktop'. The command 'head jtp.txt' has been executed, and the output is displayed line by line: 'this is javatpoint', 'you are learning linux here', 'thankyou', 'thankyou', 'thankyou', 'a', 'b', 'c', 'd', 'e'. The prompt 'sssit@JavaTpoint:~/Desktop\$' is visible at the bottom with a cursor.

```
sssit@JavaTpoint: ~/Desktop
sssit@JavaTpoint:~/Desktop$ head jtp.txt
this is javatpoint
you are learning linux here
thankyou
thankyou
thankyou
a
b
c
d
e
sssit@JavaTpoint:~/Desktop$
```

7. tail

DESCRIPTION:

tail is a command which prints the last few number of lines (10 lines by default) of a certain file, then terminates.

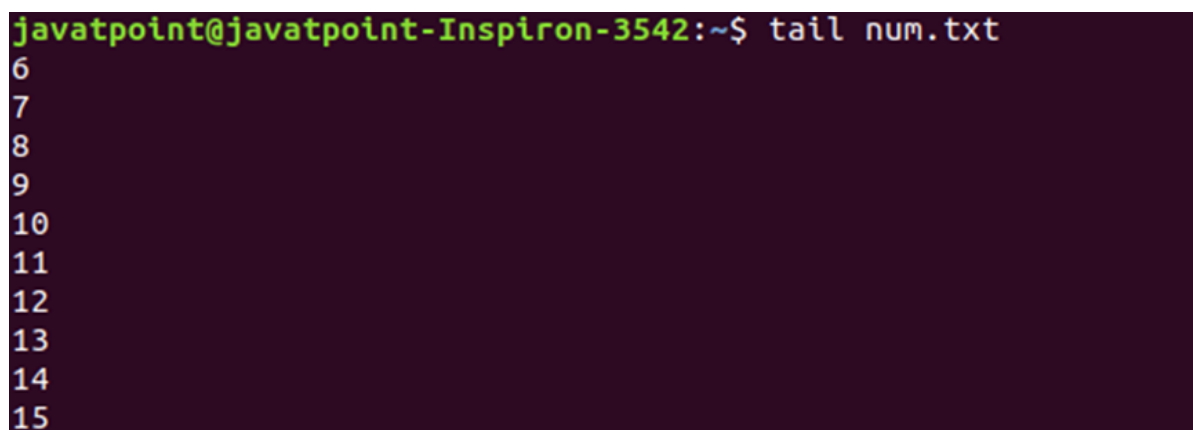
SYNTAX:

tail[option]...[file/directory]

EXAMPLE:

Output the last 100 lines of the file myfile.txt.

tail myfile.txt-n 100

A terminal window with a dark purple background. The title bar shows 'javatpoint@javatpoint-Inspiron-3542:~\$'. The command 'tail num.txt' has been executed, and the output shows line numbers 6 through 15.

```
javatpoint@javatpoint-Inspiron-3542:~$ tail num.txt
6
7
8
9
10
11
12
13
14
15
```

8. mv :Moving(and Renaming) Files

DESCRIPTION:

The `mv` command lets you move a file from one directory location to another. It also lets you rename a file(there is no separate `rename` command).

SYNTAX:

`mv[option]sourcedirectory`

EXAMPLE:

- Moves the file `myfile.txt` to the directory `destination directory`.

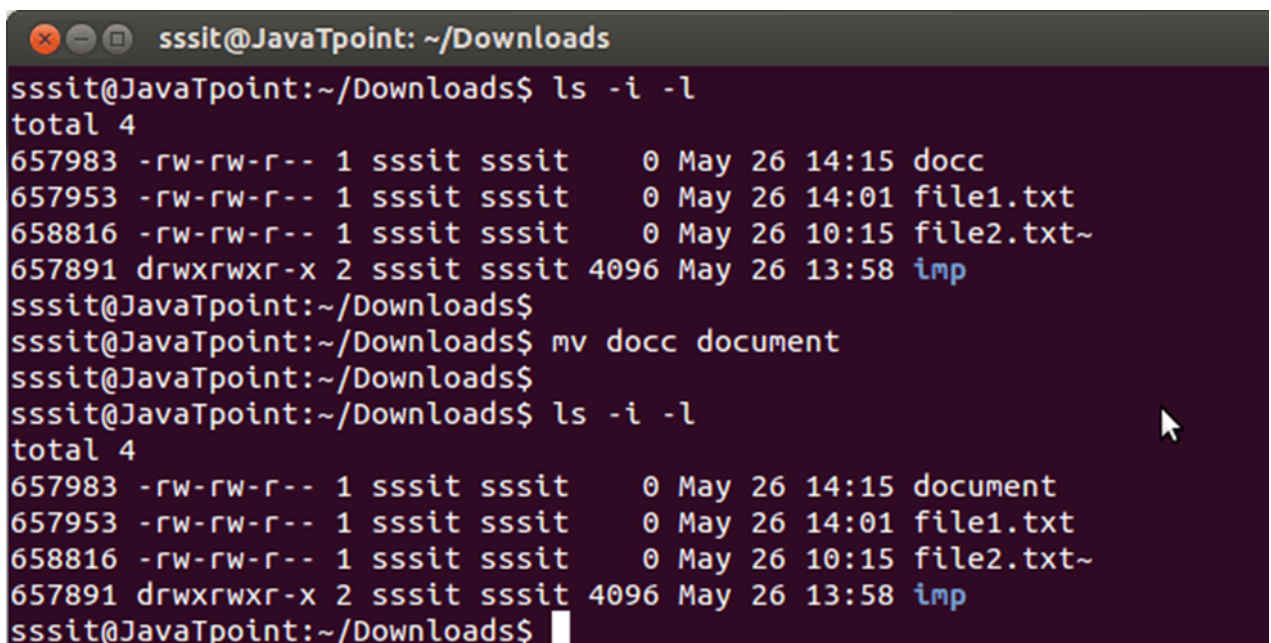
`mv myfile.txt destination_directory`

- Move the file `myfile.txt` into the parent directory.

`mv myfile.txt../`

- In this case, if `JOE1_expenses` does not exist, it will be created with the exact content of `joe_expenses`, and `joe_expenses` will disappear. If `JOE1_expenses` already exists, its content will be replaced with that of `joe_expenses` (and `joe_expenses` will still disappear).

`mv joe_expenses JOE1_expenses`

A terminal window titled 'sssit@JavaTpoint: ~/Downloads' showing a sequence of commands and their output. The user first runs 'ls -i -l' to list files with their inode numbers. Then, they run 'mv docc document' to rename the file 'docc' to 'document'. Finally, they run 'ls -i -l' again to verify the change. The output shows that 'docc' has been replaced by 'document' with the same inode number (657983).

```
sssit@JavaTpoint: ~/Downloads
sssit@JavaTpoint:~/Downloads$ ls -i -l
total 4
657983 -rw-rw-r-- 1 sssit sssit    0 May 26 14:15 docc
657953 -rw-rw-r-- 1 sssit sssit    0 May 26 14:01 file1.txt
658816 -rw-rw-r-- 1 sssit sssit    0 May 26 10:15 file2.txt~
657891 drwxrwxr-x 2 sssit sssit 4096 May 26 13:58 imp
sssit@JavaTpoint:~/Downloads$
sssit@JavaTpoint:~/Downloads$ mv docc document
sssit@JavaTpoint:~/Downloads$
sssit@JavaTpoint:~/Downloads$ ls -i -l
total 4
657983 -rw-rw-r-- 1 sssit sssit    0 May 26 14:15 document
657953 -rw-rw-r-- 1 sssit sssit    0 May 26 14:01 file1.txt
658816 -rw-rw-r-- 1 sssit sssit    0 May 26 10:15 file2.txt~
657891 drwxrwxr-x 2 sssit sssit 4096 May 26 13:58 imp
sssit@JavaTpoint:~/Downloads$
```

9. **mkdir** : Make Directory

DESCRIPTION:

If the specified directory does not already exist, **mkdir** creates it. More than one directory may be specified when calling **mkdir**.

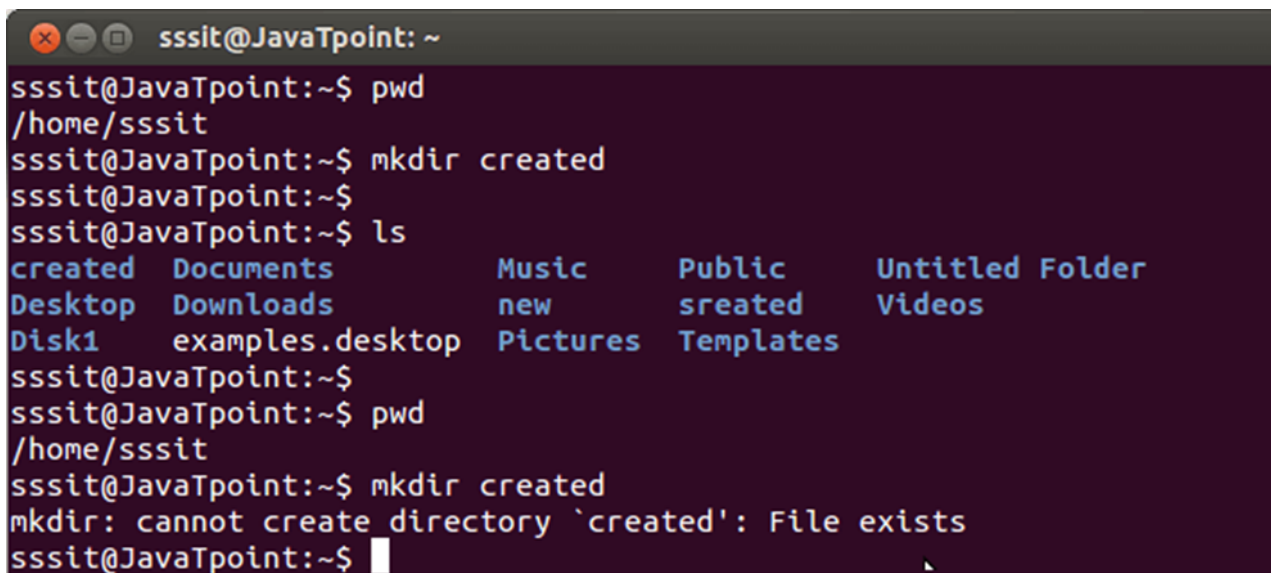
SYNTAX:

mkdir[option]directory

EXAMPLE:

Create a directory named *work*.

mkdir work

A terminal window titled 'sssit@JavaTpoint: ~' showing the execution of the 'mkdir' command. The user runs 'pwd' and gets '/home/sssit'. Then they run 'mkdir created' and 'ls', which shows a directory listing including 'created'. Finally, they run 'mkdir created' again, resulting in an error message: 'mkdir: cannot create directory `created': File exists'.

10. **cp** : Copy Files

DESCRIPTION:

The **cp** command is used to make copies of files and directories.

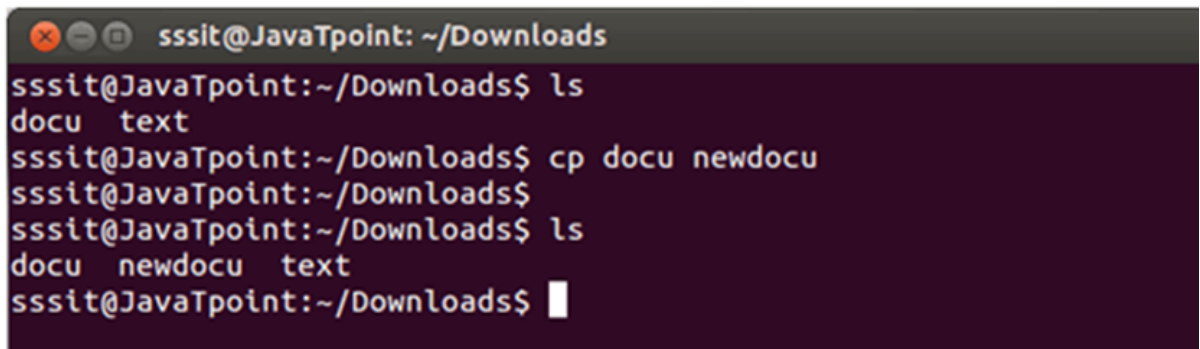
SYNTAX:

cp[option]sourcedirectory

EXAMPLE:

Creates a copy of the file in the currently working directory named origfile. The copy will be named newfile, and will be located in the working directory.

```
cp origfile newfile
```

A terminal window with a dark background and light text. The title bar shows 'sssit@JavaTpoint: ~/Downloads'. The terminal content shows a sequence of commands and their outputs: 'ls' lists 'docu' and 'text'; 'cp docu newdocu' is executed; a second 'ls' shows 'docu', 'newdocu', and 'text'.

```
sssit@JavaTpoint: ~/Downloads
sssit@JavaTpoint:~/Downloads$ ls
docu  text
sssit@JavaTpoint:~/Downloads$ cp docu newdocu
sssit@JavaTpoint:~/Downloads$
sssit@JavaTpoint:~/Downloads$ ls
docu  newdocu  text
sssit@JavaTpoint:~/Downloads$
```