

## Experiment No: 1

**AIM: Linux commands: Working with Directories:**

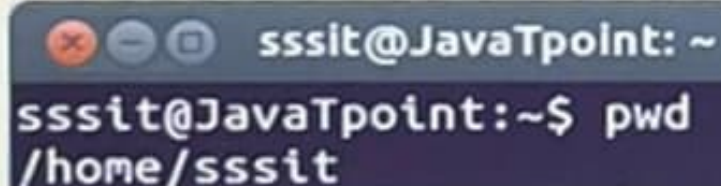
a. pwd, cd, absolute and relative paths, ls, mkdir, rmdir, touch, rm, cp, mv, echo, head, tail, cat, tac

a.

### 1. pwd(Present working directory)

In Linux pwd command displays the current working directory. It will give the whole path starting from the root to the current working directory.

Syntax: pwd

A screenshot of a Linux terminal window. The title bar shows three window control buttons (red, yellow, green) and the text 'sssit@JavaTpoint: ~'. The terminal content shows the command 'pwd' being executed, with the output '/home/sssit' displayed on the next line.

```
sssit@JavaTpoint: ~  
sssit@JavaTpoint:~$ pwd  
/home/sssit
```

### 2. cd

The "cd" command stands for 'change directory' and this command is used to change the current directory i.e; the directory in which the user is currently working.

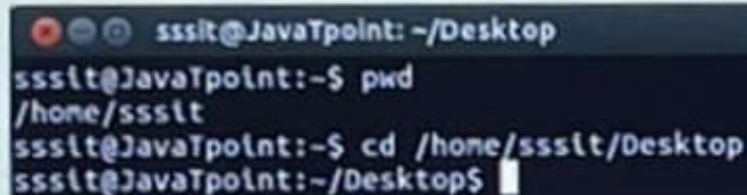
## a) Change from current directory to a new directory

**Syntax:**

**cd <dirname>**

Example:

cd Desktop

A terminal window with a dark background. The title bar shows three window control icons and the text 'sssit@JavaTpoint: ~/Desktop'. The terminal content shows the following commands and output:

```
sssit@JavaTpoint:~$ pwd
/home/sssit
sssit@JavaTpoint:~$ cd /home/sssit/Desktop
sssit@JavaTpoint:~/Desktop$
```

## b) Change directory using absolute path

An absolute path is defined as specifying the location of a file or directory from the root directory (/). It mentions the whole path starting from root.

**Syntax:**

**cd <dirname>**

Example:

cd /home/sssit/Desktop

The 'tail' command displays the last lines of a file. Its main purpose is to read the error message.

By default, it will also display the last ten lines of a file.

### Syntax:

a) tail <file name>

### Example:

tail jtp.txt

A screenshot of a Linux terminal window. The title bar shows 'sslt@JavaTpoint: ~/Desktop'. The prompt is 'sslt@JavaTpoint:~/Desktop\$'. The command 'tail jtp.txt' has been entered. The output shows the last lines of the file 'jtp.txt', which include 'r', 'g', 'h', 't', 'j', 'k', 'l', 'nnnn', and 'nnnn'. The cursor is at the end of the last line of output.

```
sslt@JavaTpoint: ~/Desktop
sslt@JavaTpoint:~/Desktop$ tail jtp.txt
r
g
h
t
j
k
l
nnnn
nnnn
sslt@JavaTpoint:~/Desktop$
```

b) Linux tail -n

The 'tail -n' option displays the specified number of lines.



### **Syntax:**

`tail -n<number> <file name> |`

### **Example:**

`tail -n 5 jtp.txt`

## **8. cat**

The 'cat' command can be used to display the content of a file. I

### **Syntax:**

a) `cat <fileName>`

### **Example:**

`cat jtp.txt`

## CD ..

g) Change from home directory and easily access to a long path

**CD ~/<dirname>**

Example

CD ~/Desktop

## 3. ls

The **ls** is the list command in Linux. It will show the full list or content of your directory. Just type **ls** and press enter key. The whole content will be shown.

Syntax:

a) **ls**



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

b) **ls -a**



List the whole files in the directory including hidden files.

**c)ls -l**

Shows List in long list format.

**d)ls -r**

It is used to print the list in reverse order.

**e)ls -R**

It will display the content of the sub-directories also.

#### **4. mkdir:**

The mkdir stands for 'make directory'. mkdir command is used to create a new directory.

**Syntax:**

**mkdir <dirname>**

Example:

mkdir created

```
sssIt@JavaTpoint: ~/Desktop
sssIt@JavaTpoint:~$ cd Desktop/
sssIt@JavaTpoint:~/Desktop$
sssIt@JavaTpoint:~/Desktop$ cat jtp.txt
this is javatpoint
you are learning linux here
thankyou
thankyou
thankyou
a
b
c
d
e
f
g
h
i
j
k
l
nnnnn
nnnnn
sssIt@JavaTpoint:~/Desktop$
```

b) To copy content of one file to another.

Syntax:

```
cat old_filename > new_filename
```

c) To concatenate content of multiple files into one.

Syntax:

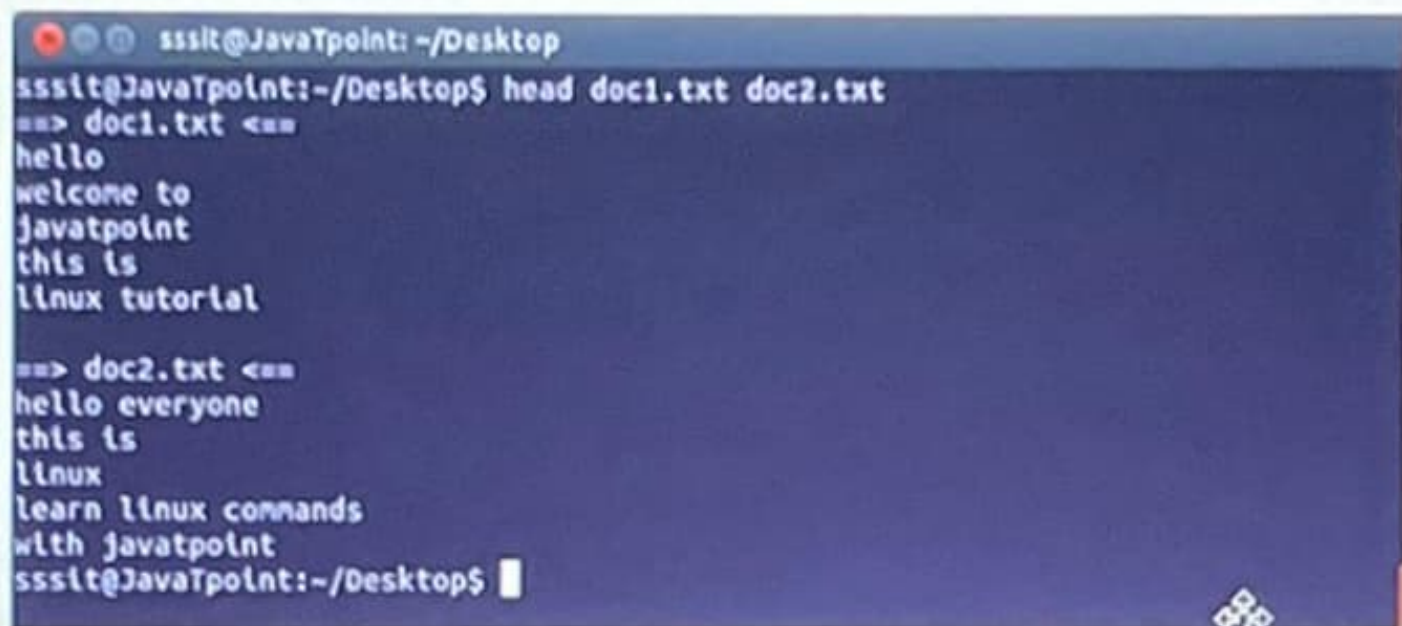
```
cat f1 f2 f3 > new_filename
```

## 9. echo

To write messages into the file.



head doc1.txt doc2.txt

A screenshot of a Linux terminal window. The title bar shows the user 'ssstt' at 'JavaTpoint' in the directory '~/Desktop'. The prompt is 'ssstt@JavaTpoint:~/Desktop\$'. The user has entered the command 'head doc1.txt doc2.txt'. The terminal shows the output for 'doc1.txt' as five lines: 'hello', 'welcome to', 'javatpoint', 'this is', and 'linux tutorial'. Then it shows the output for 'doc2.txt' as five lines: 'hello everyone', 'this is', 'linux', 'learn linux commands', and 'with javatpoint'. The prompt returns to 'ssstt@JavaTpoint:~/Desktop\$'.

```
ssstt@JavaTpoint: ~/Desktop
ssstt@JavaTpoint:~/Desktop$ head doc1.txt doc2.txt
==> doc1.txt <==
hello
welcome to
javatpoint
this is
linux tutorial

==> doc2.txt <==
hello everyone
this is
linux
learn linux commands
with javatpoint
ssstt@JavaTpoint:~/Desktop$
```

### c)Linux head -n

The 'head -n' option displays specified number of lines.

#### Syntax:

head -n <file name>

#### Example:

head -n 15 jtp.txt

## 7. tail



```

sssit@JavaTpoint: ~
sssit@JavaTpoint:~$ pwd
/home/sssit
sssit@JavaTpoint:~$ mkdir created
sssit@JavaTpoint:~$
sssit@JavaTpoint:~$ ls
created  Documents      Music      Public      Untitled Folder
Desktop  Downloads      new        sreated     Videos
Disk1    examples.desktop Pictures    Templates
sssit@JavaTpoint:~$

```

Use ls command to see the new directory.

To make multiple directories

`mkdir <dirname1> <dirname2> <dirname3> ...`

```

sssit@JavaTpoint: ~/created
sssit@JavaTpoint:~/created$ mkdir file1 file2 file3
sssit@JavaTpoint:~/created$
sssit@JavaTpoint:~/created$ ls
file1 file2 file3
sssit@JavaTpoint:~/created$

```

5. `rmdir`:

## 2. touch

touch command is used to create empty files. You can update the modification and access time of each file with the help of touch command.

### Syntax:

touch <filename>

### Example:

touch myfile1

touch myfile2

```
sssit@JavaTpoint: ~  
sssit@JavaTpoint:~$ ls  
cretecler  Disk1      Downloads  Music  Pictures  Templates  
Desktop    Documents  examples.desktop  office  Public    Videos  
sssit@JavaTpoint:~$ touch myfile1  
sssit@JavaTpoint:~$ touch myfile2  
sssit@JavaTpoint:~$ ls  
cretecler  Disk1      Downloads  Music  myfile2  Pictures  Templates  
Desktop    Documents  examples.desktop  myfile1  office    Public    Videos  
sssit@JavaTpoint:~$
```



mv command is used to move existing file or directory from one location to another. It is also used to rename a file or directory.

mv<old filename><new filename>

```
nirali@DESKTOP-EGG19F4:~/linuxprac/lin1$ pwd
/home/nirali/linuxprac/lin1
nirali@DESKTOP-EGG19F4:~/linuxprac/lin1$ mv file2.txt /home/nirali/linuxprac/lin2
nirali@DESKTOP-EGG19F4:~/linuxprac/lin1$ cd ..
nirali@DESKTOP-EGG19F4:~/linuxprac$ ls
lin1  lin2  lin3  lin4
nirali@DESKTOP-EGG19F4:~/linuxprac$ cd lin2
nirali@DESKTOP-EGG19F4:~/linuxprac/lin2$ ls
file2.txt
```

## 6. Head

The 'head' command displays the starting content of a file. By default, it displays starting 10 lines of any file.

Syntax:

Echo "message" >f1.txt|

## 10.tac

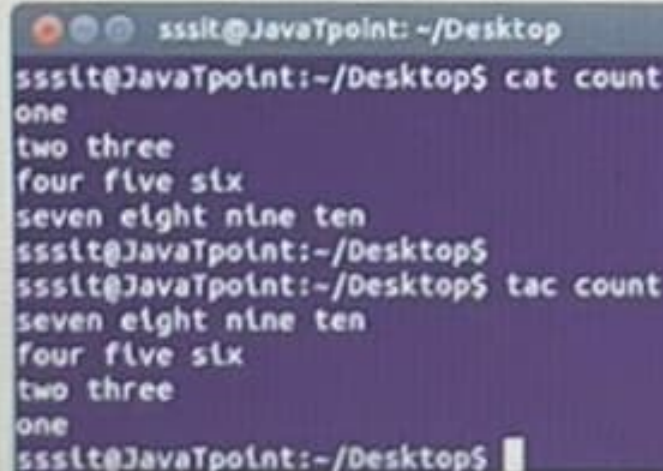
The 'tac' command is the reverse of the 'cat' command. It is also known as 'cat' backwards. It will display the file content in reverse order.

Syntax:

a) tac <file name>

Example:

tac count

A terminal window with a dark blue background and white text. The title bar at the top reads 'ssslit@JavaTpoint: ~/Desktop'. The terminal shows the following sequence of commands and output:  
1. Command: `ssslit@JavaTpoint:~/Desktop$ cat count`  
Output: `one  
two three  
four five six  
seven eight nine ten`  
2. Command: `ssslit@JavaTpoint:~/Desktop$`  
3. Command: `ssslit@JavaTpoint:~/Desktop$ tac count`  
Output: `seven eight nine ten  
four five six  
two three  
one`  
4. Command: `ssslit@JavaTpoint:~/Desktop$`  
A white cursor is visible at the end of the last command line.



```
ssstt@JavaTpoint: ~  
ssstt@JavaTpoint:~$ ls  
cretecler  Disk1      Downloads  Music      myfile2  Pictures  Templates  
Desktop    Documents  examples.desktop  myfile1  office   Public    Videos  
ssstt@JavaTpoint:~$  
ssstt@JavaTpoint:~$ rm myfile1  
ssstt@JavaTpoint:~$  
ssstt@JavaTpoint:~$ ls  
cretecler  Disk1      Downloads  Music      office   Public    Videos  
Desktop    Documents  examples.desktop  myfile2  Pictures  Templates  
ssstt@JavaTpoint:~$
```

#### 4. cp -r

'cp' means copy. 'cp' command is used to copy a file or a directory.

To copy a file into the same directory syntax will be,

a) **cp -r <existing file name> <new file name>**

```
nirali@DESKTOP-EGG19F4:~/linuxprac$ ls  
lin1  lin2  lin3  
nirali@DESKTOP-EGG19F4:~/linuxprac$ cp -r lin1 lin4  
nirali@DESKTOP-EGG19F4:~/linuxprac$ ls  
lin1  lin2  lin3  lin4  
nirali@DESKTOP-EGG19F4:~/linuxprac$ cd lin1  
nirali@DESKTOP-EGG19F4:~/linuxprac/lin1$ cp -r file1.txt file2.txt  
nirali@DESKTOP-EGG19F4:~/linuxprac/lin1$ ls  
file1.txt  file2.txt
```

#### 5. mv

## Syntax:

a) head <file name>

## Example:

head jtp.txt

A terminal window with a dark background. The title bar shows 'ssstt@JavaTpoint: ~/Desktop'. The command 'ssstt@JavaTpoint:~/Desktop\$ head jtp.txt' has been entered. The output consists of five lines: 'this is javatpoint', 'you are learning linux here', 'thankyou', 'thankyou', and 'thankyou'. Below these, the letters 'a', 'b', 'c', 'd', and 'e' are listed vertically. The prompt 'ssstt@JavaTpoint:~/Desktop\$' is visible at the bottom with a cursor.

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

## Head command for multiple files

If we'll write two file names then it will display first ten lines (in this case file has five lines only) of each file separated by a heading.

## Syntax:

b) head <file name> <file name>

## Example:



If you would like to create two files simultaneously, then your command will be:

**Syntax:**

```
touch <filename> <filename> ...
```

**Example:**

```
touch myfile1 myfile2
```

### **3. rm**

I

The 'rm' means remove. This command is used to remove a file. The command line doesn't have a recycle bin or trash unlike other GUI's to recover the files. Hence, be very much careful while using this command. Once you have deleted a file, it is removed permanently.

**Syntax:**

```
a) rm -r <filename>
```

**Example:**

```
rm myfile1
```



This command is used to remove an empty directory.

**Syntax:**

**rm -d <dirname>**

Example: rm -d created

```
ssslit@JavaTpoint: ~  
ssslit@JavaTpoint:~/created$ ls  
file1 file2  
ssslit@JavaTpoint:~/created$ rm -d file1  
ssslit@JavaTpoint:~/created$ ls  
file2
```

This command is used to remove a directory along with its subdirectories.

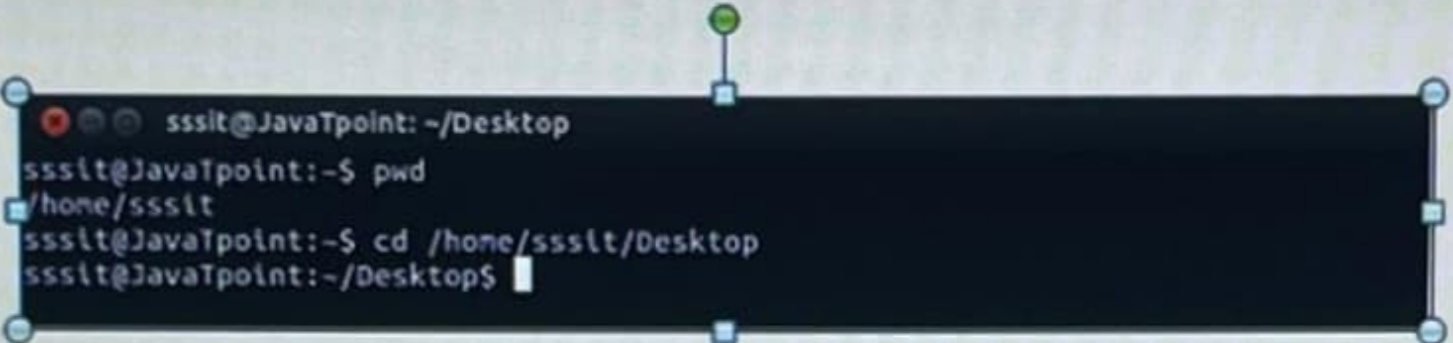
**Syntax:**

**rm -r <dirname>**

Example: rm -r file

```
the_programmer@theprogrammer-virtual-machine:~/fyit-a/Sign$ ls  
Boys file Girls  
the_programmer@theprogrammer-virtual-machine:~/fyit-a/Sign$ rm -r file  
the_programmer@theprogrammer-virtual-machine:~/fyit-a/Sign$ ls  
Boys Girls  
the_programmer@theprogrammer-virtual-machine:~/fyit-a/Sign$
```





```
ssstt@JavaTpoint: ~/Desktop
ssstt@JavaTpoint:~$ pwd
/home/ssstt
ssstt@JavaTpoint:~$ cd /home/ssstt/Desktop
ssstt@JavaTpoint:~/Desktop$
```

### c) Change directory using relative path

Relative path is defined as the path related to the present working directory (pwd). It starts at your current directory and **never starts with a /**.

**Syntax:**

**cd <dirname>**

**Example:**

**cd Desktop**

### d) Change to root directory

**CD /**

### e) Change to current directory

**CD .**

### f) Change to parent directory