

Experiment No : 6

AIM

Familiarization of LINUX Commands.

CO2

Perform system administration task.

Procedure

1. grep

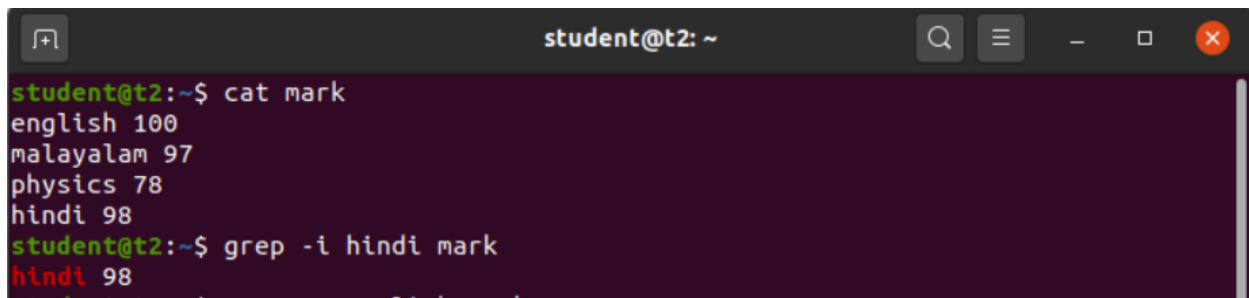
grep command is used to filter and display contents accordingly which makes our search easy.

1.1) grep -i [pattern] [filename]

used to search the pattern and display contents accordingly, -i implies the search is insensitive.

```
$grep -i hindi mark
```

Output

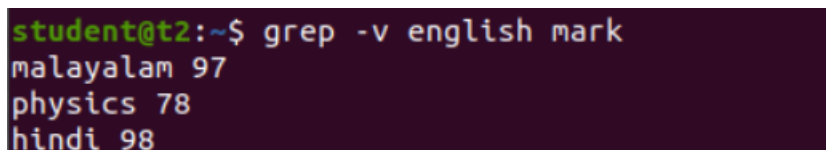
A terminal window titled 'student@t2: ~' with search, menu, and window control icons. It shows the command 'cat mark' and its output: 'english 100', 'malayalam 97', 'physics 78', 'hindi 98'. Then it shows the command 'grep -i hindi mark' and its output: 'hindi 98'.

1.2) grep -v [pattern] [filename]

performs inverted search which implies displaying all the contents which does not have the specified pattern.

```
$ grep -v english mark
```

Output

A terminal window showing the command 'grep -v english mark' and its output: 'malayalam 97', 'physics 78', 'hindi 98'.

1.3) `grep -A1 [pattern] [filename]`

displays the searched content along with a line after it.

`$ grep -A1 physics mark`

Output

```
student@t2:~$ grep -A1 physics mark
physics 78
hindi 98
```

1.4) `grep -B1 [pattern] [filename]`

displays the searched content along with a line before it.

`$ grep -B1 physics mark`

Output

```
student@t2:~$ grep -B1 physics mark
malayalam 97
physics 78
```

1.5) `grep -C1 [pattern] [filename]`

displays the searched content along with a line before and after it. It works as a combination of `-A1` and `-B1` option of `grep` command.

`$ grep -C1 malayalam mark`

Output

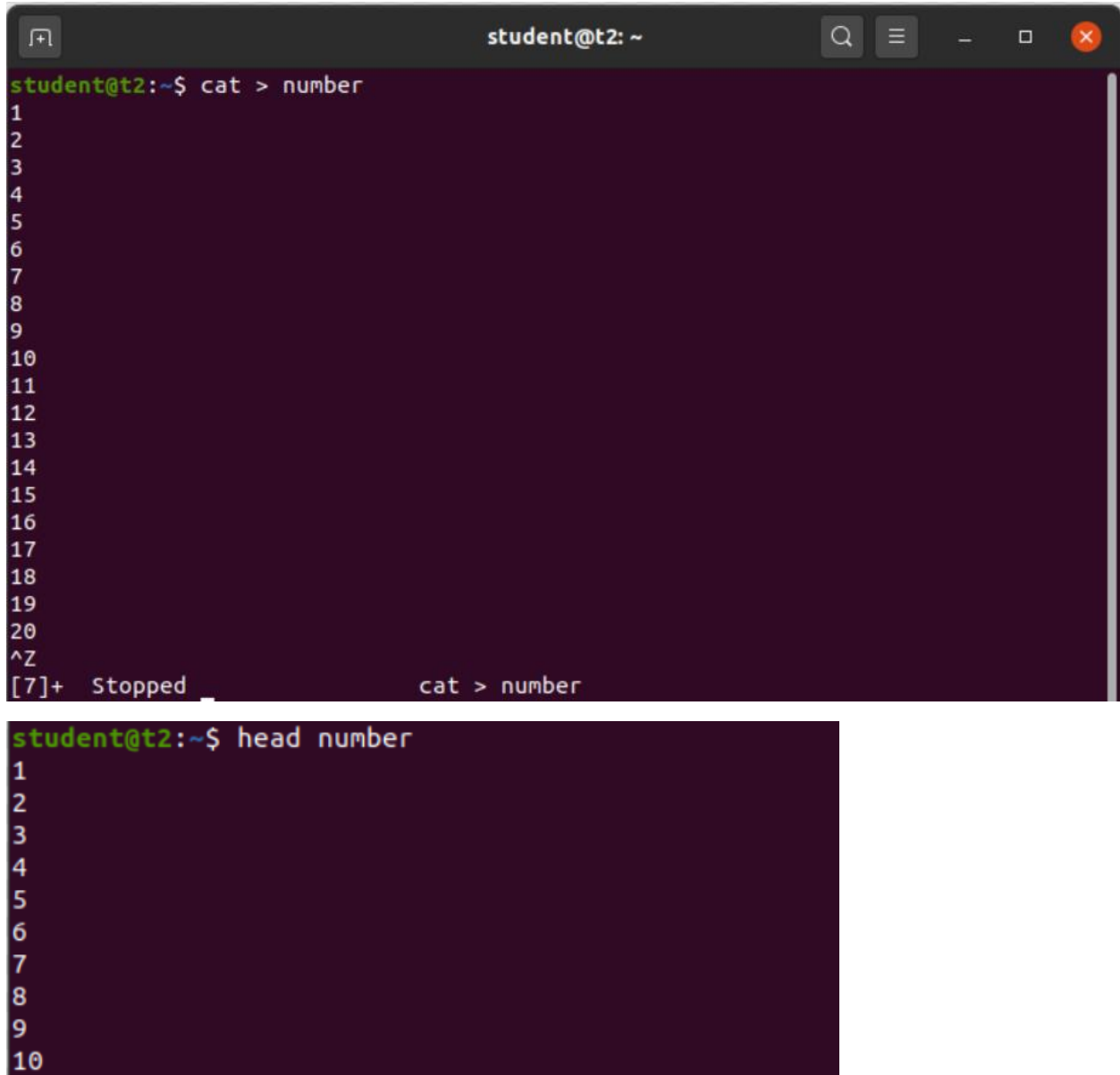
```
student@t2:~$ grep -C1 malayalam mark
english 100
malayalam 97
physics 78
student@t2:~$
```

2. `head`

displays the top contents of a file. By default, it displays first 10 lines of file.

`$ head number`

Output



```
student@t2: ~  
student@t2:~$ cat > number  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
^Z  
[7]+ Stopped cat > number  
  
student@t2:~$ head number  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10
```

2.1) `head -[number of lines] [filename]`

displays only the number of lines specified by the user.

\$ `head -5 number`

Output

```
student@t2:~$ head -5 number
1
2
3
4
5
student@t2:~$
```

3. tail

displays the bottom contents of a file. By default, it displays last 10 lines of file.

\$ tail number

Output

```
student@t2: ~
student@t2:~$ tail number
11
12
13
14
15
16
17
18
19
20
```

3.1) tail -[number of lines] [filename]

displays only the number of lines specified by the user.

\$ tail -5 number

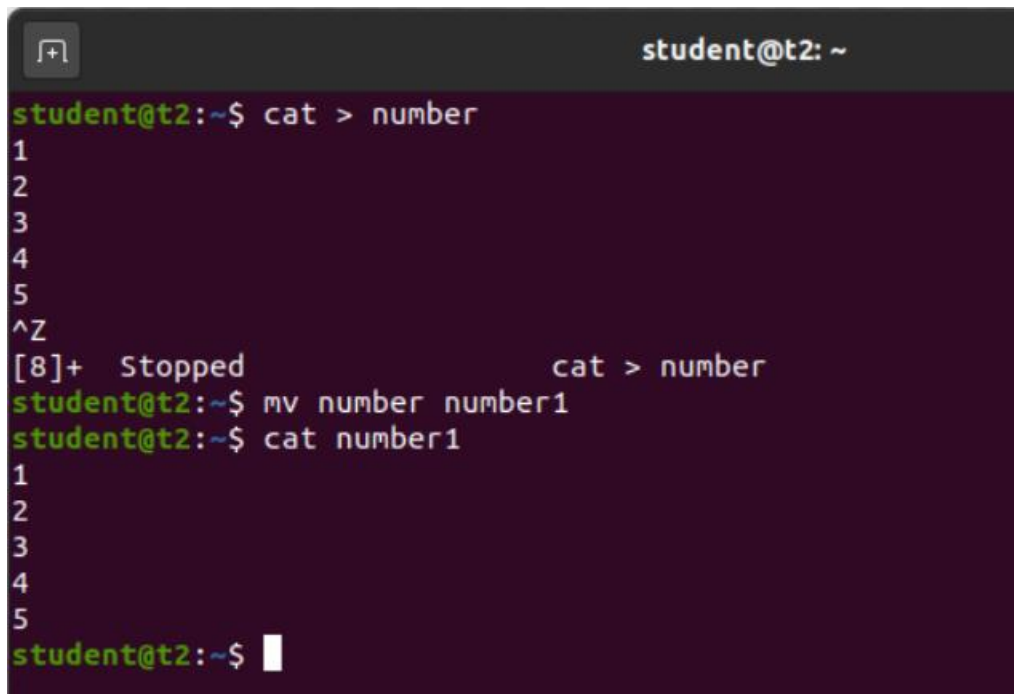
Output

```
student@t2:~$ tail -5 number
16
17
18
19
20
```

4) mv [source file] [copy file]

To move the contents of file 1 to file 2 by overwriting the contents of file and replacing it with new name.

```
$ mv number number1
```

OutputA terminal window titled 'student@t2: ~' showing a series of commands and their outputs. The user enters 'cat > number', which outputs the numbers 1 through 5 followed by a control character '^Z'. The user then enters '[8]+ Stopped', which outputs 'cat > number'. Next, the user enters 'mv number number1', and finally 'cat number1', which outputs the numbers 1 through 5. The prompt '\$' is visible at the end of the last line.

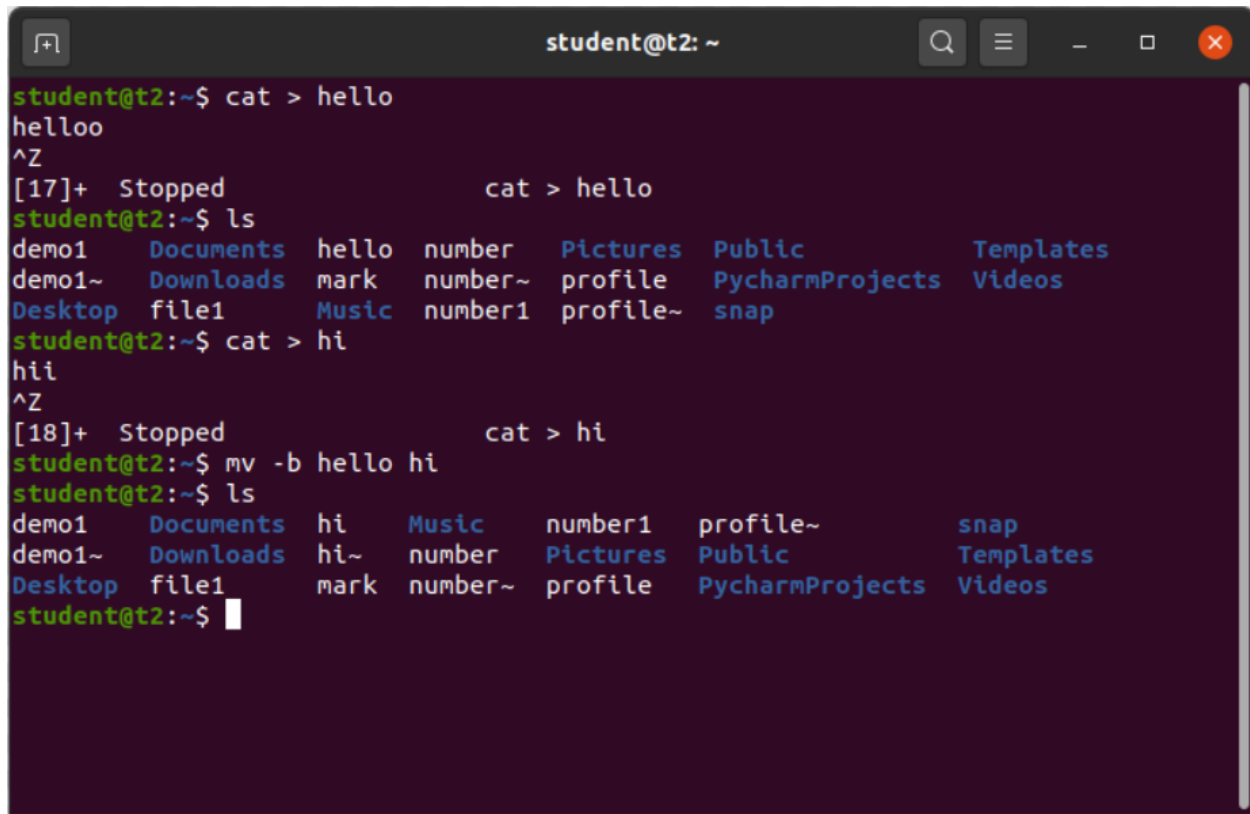
```
student@t2:~$ cat > number
1
2
3
4
5
^Z
[8]+  Stopped                  cat > number
student@t2:~$ mv number number1
student@t2:~$ cat number1
1
2
3
4
5
student@t2:~$
```

4.1) mv -b [source file] [copy file]

To move the contents of file 1 to file 2 while keeping a backup of original file.

```
$ mv -b hello hi
```

Output



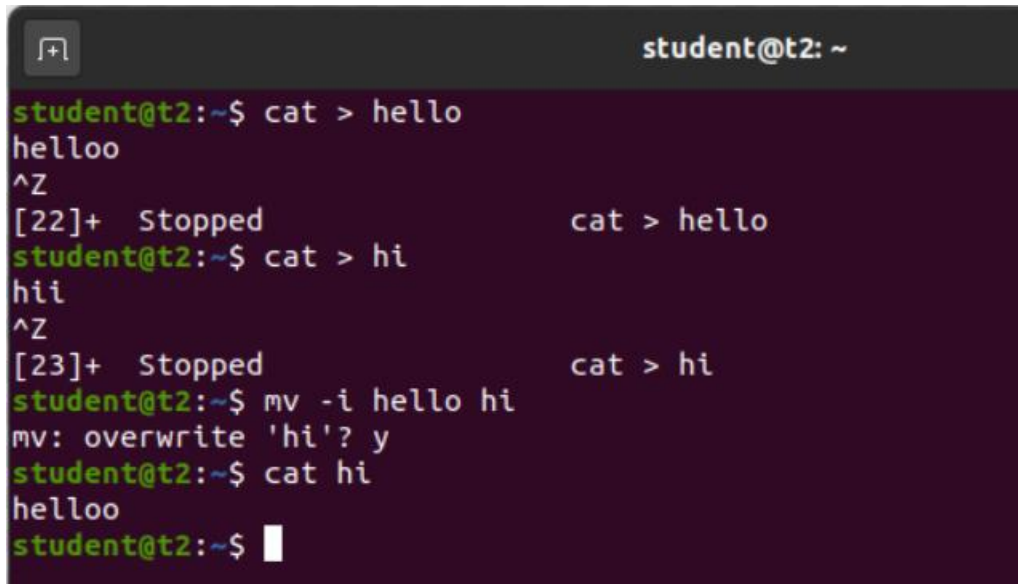
```
student@t2: ~  
student@t2:~$ cat > hello  
helloo  
^Z  
[17]+  Stopped                  cat > hello  
student@t2:~$ ls  
demo1  Documents  hello  number  Pictures  Public  Templates  
demo1~ Downloads  mark   number~ profile  PycharmProjects  Videos  
Desktop file1    Music  number1 profile~ snap  
student@t2:~$ cat > hi  
hii  
^Z  
[18]+  Stopped                  cat > hi  
student@t2:~$ mv -b hello hi  
student@t2:~$ ls  
demo1  Documents  hi  Music  number1  profile~  snap  
demo1~ Downloads  hi~  number  Pictures  Public  Templates  
Desktop file1    mark  number~  profile  PycharmProjects  Videos  
student@t2:~$
```

4.2) mv -i [source file] [copy file]

To move the contents of file 1 to file 2 by overwriting the original file.

```
$ mv -i hello hi
```

Output



```
student@t2: ~  
student@t2:~$ cat > hello  
helloo  
^Z  
[22]+  Stopped                  cat > hello  
student@t2:~$ cat > hi  
hii  
^Z  
[23]+  Stopped                  cat > hi  
student@t2:~$ mv -i hello hi  
mv: overwrite 'hi'? y  
student@t2:~$ cat hi  
helloo  
student@t2:~$
```

Result

The program has been executed and output has been verified.