

1) Meaning of different user prompt

\$, #, % symbols indicate the user account type you are logged in to.

- Dollar sign (\$) means you are a normal user.
- hash (#) means you are the system administrator (root).
- In the C shell, the prompt ends with a percentage sign (%).

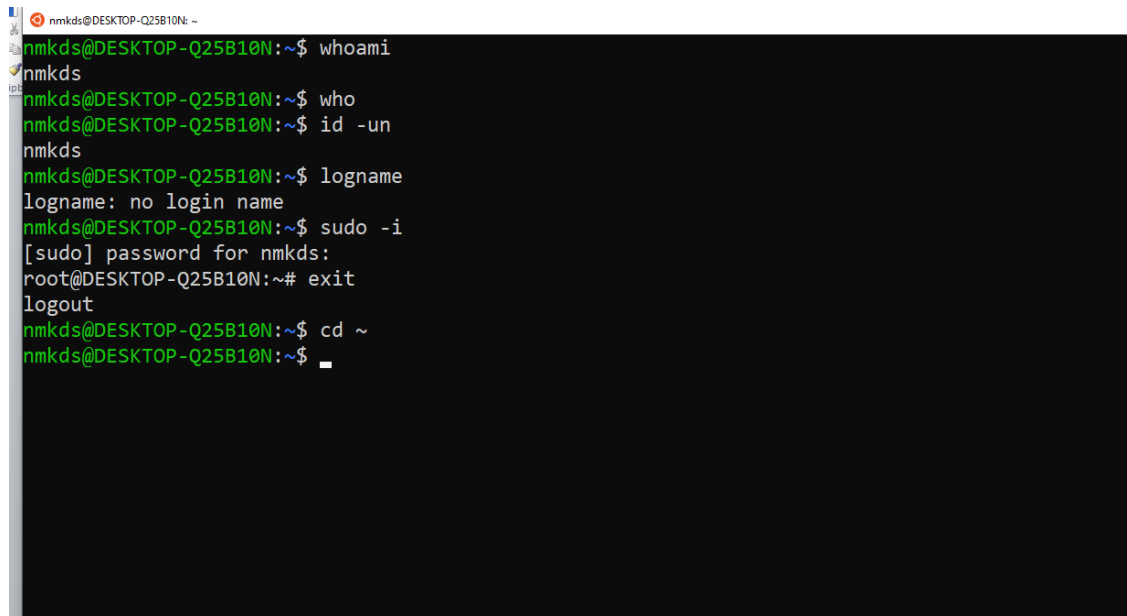
2) How to switch between normal user and root

a) From root to user
type “exit” and enter

b) To go to root
`sudo -i`

c) To go to home directory
type “cd ~”

2) To get username , logname is not displaying

A terminal window with a black background and green text. The prompt is 'nmkds@DESKTOP-Q25B10N: ~'. The user enters 'whoami' and 'who', both returning 'nmkds'. Then 'id -un' is entered, also returning 'nmkds'. Next, 'logname' is entered, returning 'logname: no login name'. Then 'sudo -i' is entered, followed by a password prompt '[sudo] password for nmkds:'. After the password is entered, the prompt changes to 'root@DESKTOP-Q25B10N:~#'. The user enters 'exit', returning to the 'nmkds' prompt. Then 'cd ~' is entered, and the prompt remains 'nmkds@DESKTOP-Q25B10N:~\$'.

```
nmkds@DESKTOP-Q25B10N: ~  
nmkds@DESKTOP-Q25B10N:~$ whoami  
nmkds  
nmkds@DESKTOP-Q25B10N:~$ who  
nmkds@DESKTOP-Q25B10N:~$ id -un  
nmkds  
nmkds@DESKTOP-Q25B10N:~$ logname  
logname: no login name  
nmkds@DESKTOP-Q25B10N:~$ sudo -i  
[sudo] password for nmkds:  
root@DESKTOP-Q25B10N:~# exit  
logout  
nmkds@DESKTOP-Q25B10N:~$ cd ~  
nmkds@DESKTOP-Q25B10N:~$
```

```
nmkds@DESKTOP-Q25B10N: ~  
nmkds@DESKTOP-Q25B10N:~$ pwd  
/home/nmkds  
nmkds@DESKTOP-Q25B10N:~$ ls -l  
total 24  
drwxr-xr-x 1 nmkds nmkds 4096 Oct 6 14:47 Assignment  
drwxr-xr-x 1 nmkds nmkds 4096 Oct 6 14:45 Lab  
-rw-r--r-- 1 nmkds nmkds 34 Oct 6 13:43 Start  
drwxr-xr-x 1 nmkds nmkds 4096 Oct 6 13:46 a  
drwxr-xr-x 1 nmkds nmkds 4096 Oct 6 13:45 b  
drwxr-xr-x 1 nmkds nmkds 4096 Oct 6 13:45 c  
drwxr-xr-x 1 nmkds nmkds 4096 Oct 4 20:46 cv  
drwxr-xr-x 1 nmkds nmkds 4096 Oct 5 18:02 edac  
-rw-r--r-- 1 nmkds nmkds 0 Oct 4 23:58 file  
-rw-r--r-- 1 nmkds nmkds 0 Oct 4 23:58 file.txt  
-rw-r--r-- 1 nmkds nmkds 15 Oct 6 09:44 test.txt  
-rw-r--r-- 1 nmkds nmkds 24576 Oct 4 23:58 typescript  
nmkds@DESKTOP-Q25B10N:~$ ls -a  
.  
..  
bash_logout  
bashrc  
bash_history  
landscape  
motd_shown  
ns.txt.swp  
profile  
sudo_as_admin_successful  
Assignment  
Lab  
Start  
a  
b  
c  
cv  
edac  
file  
file.txt  
test.txt  
typescript
```

Cat command

Cat(concatenate) command is very frequently used in Linux. It reads data from the file and gives their content as output. It helps us to create, view, concatenate files. So let us see some frequently used cat commands.

1) To view a single file

Command:

```
$cat filename
```

Output

It will show content of given filename

2) To view multiple files

Command:

```
$cat file1 file2
```

3) To view contents of a file preceding with line numbers.

Command:

```
$cat -n filename
```

Output

It will show content with line number

example:-cat-n file.txt

1)This is file

2)A unique array

4) Create a file

Command:

```
$ cat >newfile
```

Output

Will create and a file named newfile

5) Copy the contents of one file to another file.

Command:

```
$cat [filename-whose-content-is-to-be-copied] > [destination-filename]
```

Output

The content will be copied in destination file

6) Cat command can suppress repeated empty lines in output

Command:

```
$cat -s file.txt
```

Will suppress repeated empty lines in output

7) Cat command can append the contents of one file to the end of another file.

Command:

```
$cat file1 >> file2
```

Output

Will append the contents of one file to the end of another file

8) Cat command can display content in reverse order using tac command.

Command:

```
$tac filename
```

Output

Will display content in reverse order

9) Cat command can highlight the end of line.

Command:

```
$cat -E "filename"
```

Output

Will highlight the end of line

10) If you want to use the -v, -E and -T option together, then instead of writing -vET in the command, you can just use the -A command line option.

Command

```
$cat -A "filename"
```

11) Cat command to open dashed files.

Command:

```
$cat -- "-dashfile"
```

Output

Will display the content of -dashfile

12) Cat command if the file has a lot of content and can't fit in the terminal.

Command:

```
$cat "filename" | more
```

Output

Will show that much content, which could fit in terminal and will ask to show more.

12) Cat command to merge the contents of multiple files.

```
$cat "filename1" "filename2" "filename3" > "merged_filename"
```

Output

Will merge the contents of file in respective order and will insert that content in "merged_filename".

13) Cat command to display the content of all text files in the folder.

Command:

```
$cat *.txt
```

Output

Will show the content of all text files present in the folder.

```
nmkds@DESKTOP-Q25B10N: ~  
nmkds@DESKTOP-Q25B10N:~$ echo "Create a file type data and display it"  
Create a file type data and display it  
nmkds@DESKTOP-Q25B10N:~$ cat file1.txt  
Now type into  
File some data  
and come out  
nmkds@DESKTOP-Q25B10N:~$ echo " It showed data of already created file"  
 It showed data of already created file  
nmkds@DESKTOP-Q25B10N:~$ cat >newfile.txt  
Hi i m creating a new file and writing in it  
to stop  
press control D  
nmkds@DESKTOP-Q25B10N:~$ echo "display data for new file"  
display data for new file  
nmkds@DESKTOP-Q25B10N:~$ cat newfile.txt  
Hi i m creating a new file and writing in it  
to stop  
press control D  
nmkds@DESKTOP-Q25B10N:~$ echo "Combine 2 files or display data togethr"  
echoCombine 2 files or display data togethr: command not found  
nmkds@DESKTOP-Q25B10N:~$ cat file1.txt newfile.txt  
Now type into  
File some data  
and come out  
Hi i m creating a new file and writing in it  
to stop  
press control D  
nmkds@DESKTOP-Q25B10N:~$
```

Seq command

seq command in Linux is used to generate numbers from *FIRST* to *LAST* in steps of *INCREMENT*. It is a very useful command where we had to generate list of numbers in while, for, until loop.

Syntax:

```
seq [OPTION]... LAST  
or  
seq [OPTION]... FIRST LAST  
or  
seq [OPTION]... FIRST INCREMENT LAST
```

```
nmkds@DESKTOP-Q25B10N: ~  
nmkds@DESKTOP-Q25B10N:~$ seq 5  
1  
2  
3  
4  
5  
nmkds@DESKTOP-Q25B10N:~$ seq 1 4 20  
1  
5  
9  
13  
17  
nmkds@DESKTOP-Q25B10N:~$ echo "U observe there is a gap in numbers by 4"  
U observe there is a gap in numbers by 4  
nmkds@DESKTOP-Q25B10N:~$ seq -w 10  
01  
02  
03  
04  
05  
06  
07  
08  
09  
10
```

Options:

- **seq LAST** : When only one argument is given then it produces numbers from 1 to *LAST* in step increment of 1. If the *LAST* is less than 1, then it produces no output.

```
File Edit View Search Terminal Help  
naman@root:~$ seq 10  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
naman@root:~$
```

- **seq FIRST LAST** : When two arguments are given then it produces numbers from *FIRST* till *LAST* in step increment of 1. If *LAST* is less than *FIRST*, then it produces no output.

```
File Edit View Search Terminal Help
naman@root:~$ seq 3 9
3
4
5
6
7
8
9
naman@root:~$
```

- **seq FIRST INCREMENT LAST** : When three arguments are given then it produces numbers from *FIRST* till *LAST* in step of *INCREMENT*. If *LAST* is less than *FIRST*, then it produces no output.

```
File Edit View Search Terminal Help
naman@root:~$ seq 3 7 30
3
10
17
24
naman@root:~$
```

- **seq -f "FORMAT" FIRST INCREMENT LAST** : This command is used to generate sequence in a formatted manner. *FIRST* and *INCREMENT* are optional.

```
File Edit View Search Terminal Help
naman@root:~$ seq -f "GFG%02g" 4
GFG01
GFG02
GFG03
GFG04
naman@root:~$ seq -f "GFG%02g" 2 4
GFG02
GFG03
GFG04
naman@root:~$ seq -f "GFG%02g" 10 10 40
GFG10
GFG20
GFG30
GFG40
naman@root:~$
```


- **seq -s "STRING" FIRST INCREMENT LAST** : This command is used to *STRING* to separate numbers. By default this value is equal to "\n". *FIRST* and *INCREMENT* are optional.

Recommended: Please try your approach on [{IDE}](#) first, before moving on to the solution.

```
File Edit View Search Terminal Help
naman@root:~$ seq -s " " 10
1 2 3 4 5 6 7 8 9 10
naman@root:~$ seq -s " " 5 10
5 6 7 8 9 10
naman@root:~$ seq -s " " 5 4 15
5 9 13
naman@root:~$
```

- **seq -w FIRST INCREMENT LAST** : This command is used to equalize width by padding with leading zeroes. *FIRST* and *INCREMENT* are optional.

```
File Edit View Search Terminal Help
naman@root:~$ seq -w 10
01
02
03
04
05
06
07
08
09
10
naman@root:~$ seq -w 99 101
099
100
101
naman@root:~$ seq -w 1 10 50
01
11
21
31
41
naman@root:~$
```