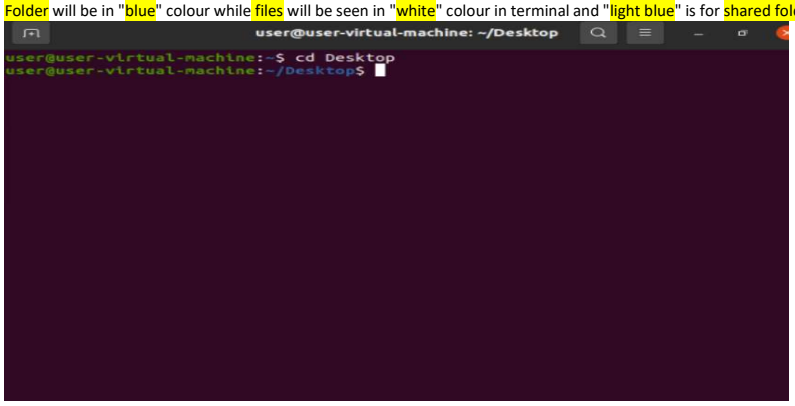
UBUNTU

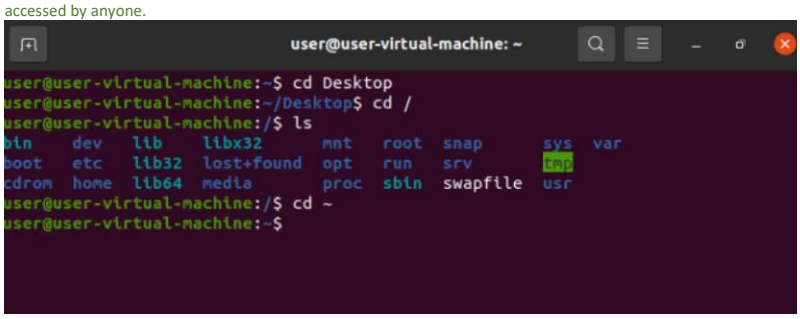
26 September 2021 17:17

1. **cd:**
   The "**ls**" is list command. "cd" command used to change directory using "**cd**" I can go to any directory(location).

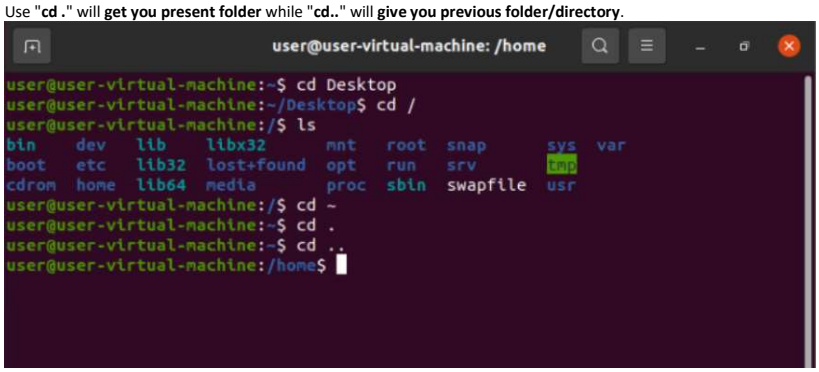
Folder will be in "blue" colour while files will be seen in "white" colour in terminal and "light blue" is for shared folder.



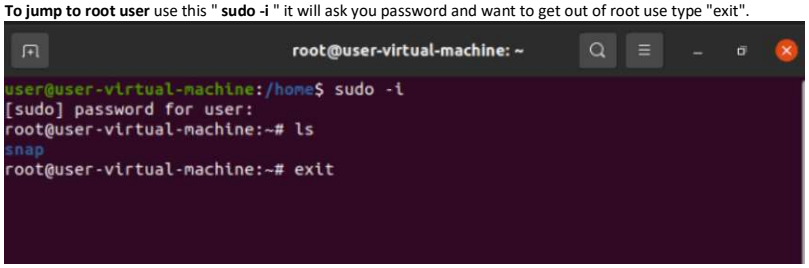
The '**cd ~** and **cd /**':
Use "**cd ~**" will give send you in **local home directory** which is denoted by "~" in terminal while use "**cd /**" will give you **root home directory** which is denoted by "/" root directories can be seen by anyone but can't be accessed by anyone.



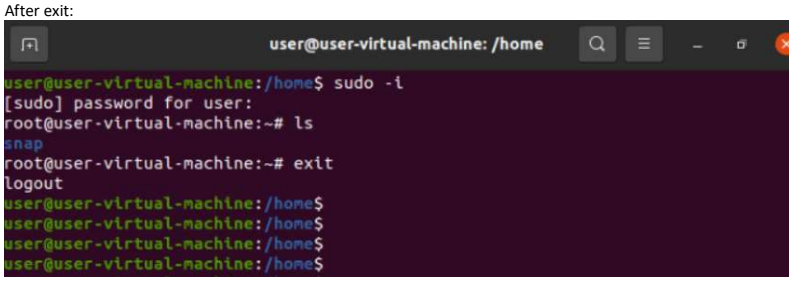
'cd.' And '**cd ..**' :
Use "**cd .**" will **get you present folder** while "**cd..**" will **give you previous folder/directory**.



To jump to root user use this " **sudo -i** " it will ask you password and want to get out of root use type "exit".



After exit:



"**pwd** " will tell me my **present working directory** .

```
user@user-virtual-machine:~$ cd ..
user@user-virtual-machine:/home$
user@user-virtual-machine:/home$ pwd
/home
user@user-virtual-machine:/home$
```

Use "**echo 'something'** " to **print something** in terminal.

```
user@user-virtual-machine:/home$ echo 'something'
something
user@user-virtual-machine:/home$
```

Clear command:

Before clear

```
user@user-virtual-machine: /home
user@user-virtual-machine:/home$ sudo -i
[sudo] password for user:
root@user-virtual-machine:~# ls
snap
root@user-virtual-machine:~# exit
logout
user@user-virtual-machine:/home$
user@user-virtual-machine:/home$
user@user-virtual-machine:/home$
user@user-virtual-machine:/home$
user@user-virtual-machine:/home$
user@user-virtual-machine:/home$
user@user-virtual-machine:/home$
user@user-virtual-machine:/home$ echo 'something'
something
user@user-virtual-machine:/home$ clear
```

After clear command:

```
user@user-virtual-machine: /home
user@user-virtual-machine:/home$
```

The ls command:

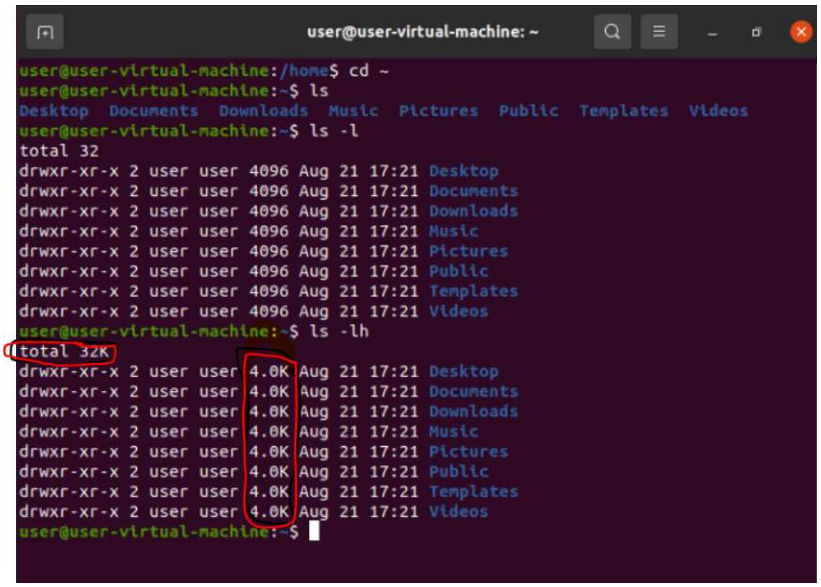
1.Use "**ls -a**" to **show all folder and file even hidden folder**, In linux if a folder/file starts with "**.**" it is a **hidden folder**.

```
user@user-virtual-machine: /home
user@user-virtual-machine:/home$ ls
user
user@user-virtual-machine:/home$ ls -a
.  ..  user
user@user-virtual-machine:/home$ ls -A
user
user@user-virtual-machine:/home$
```

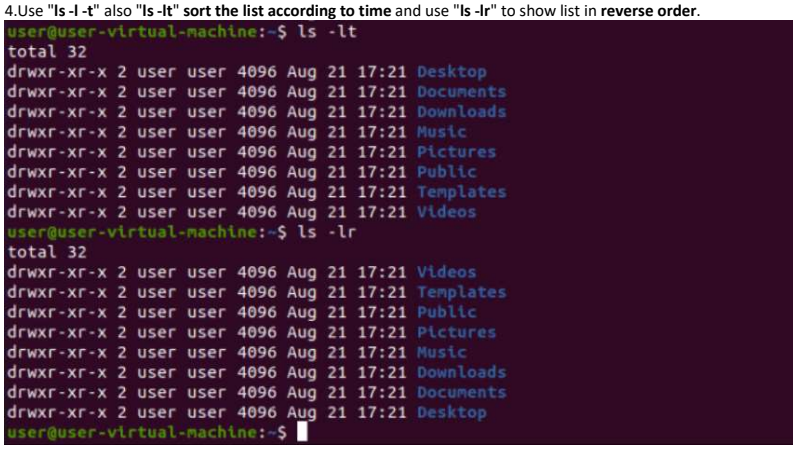
2.Use "**ls -l**" to shows all folder/ file in **list format**, with all info about files and the accessibility like 'drwrwxr-x'.

```
user@user-virtual-machine: ~
user@user-virtual-machine:/home$ cd ~
user@user-virtual-machine:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
user@user-virtual-machine:~$ ls -l
total 32
drwxr-xr-x 2 user user 4096 Aug 21 17:21 Desktop
drwxr-xr-x 2 user user 4096 Aug 21 17:21 Documents
drwxr-xr-x 2 user user 4096 Aug 21 17:21 Downloads
drwxr-xr-x 2 user user 4096 Aug 21 17:21 Music
drwxr-xr-x 2 user user 4096 Aug 21 17:21 Pictures
drwxr-xr-x 2 user user 4096 Aug 21 17:21 Public
drwxr-xr-x 2 user user 4096 Aug 21 17:21 Templates
drwxr-xr-x 2 user user 4096 Aug 21 17:21 Videos
user@user-virtual-machine:~$ ls
```

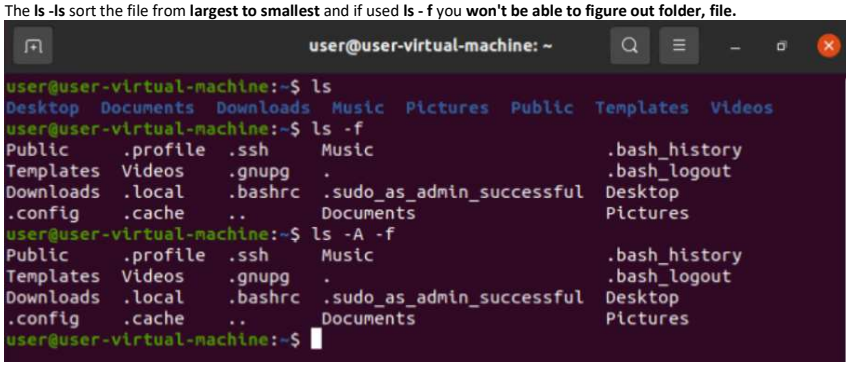
3.Use "**ls -lh**" to show all things that shows in "ls -l" but in **human readable format**.



4.Use "**ls -l -t**" also "**ls -lt**" **sort the list according to time** and use "**ls -lr**" to show list in **reverse order**.



The **ls -ls** sort the file from **largest to smallest** and if used **ls - f** you **won't be able to figure out folder, file.**



Use "**cat file_name**" to display the content inside the file or "cat*" to print the content of every file present in that directory.

```
user@user-virtual-machine: ~
user@user-virtual-machine:~$ cat filename
hello
this is filename
hello
user@user-virtual-machine:~$ cat *
cat: Desktop: Is a directory
cat: Documents: Is a directory
cat: Downloads: Is a directory
hello
hello
this is filename
hello
cat: Music: Is a directory
cat: Pictures: Is a directory
cat: Public: Is a directory
cat: Templates: Is a directory
cat: Videos: Is a directory
user@user-virtual-machine:~$
```

Use "**cat -n filename**" to display the same thing as " cat file_name" but with **numbers in beginning**.

```
user@user-virtual-machine:~$ cat -n filename
     1  hello
     2  this is filename
     3  hello
```

Use **echo 'some_data' > file_name** to **create a file in a directory**.

```
user@user-virtual-machine:~$ echo 'somedata' > f2
user@user-virtual-machine:~$ cat f2
somedata
```

Use "**echo 'append_data' >> filename**" to **append give data into directory** or **create a file with given data in-case file is not created**.

```
user@user-virtual-machine:~$ echo 'append_data' >> filename
user@user-virtual-machine:~$ cat filename
append_data
user@user-virtual-machine:~$ echo 'append_data2' >> filename
user@user-virtual-machine:~$ cat filename
append_data
append_data2
user@user-virtual-machine:~$
```

Use "**cat -s filename > filename**" to **clear common data in the file**.

```
user@user-virtual-machine:~$ cat -s filename
hello

this is filename

    hello
user@user-virtual-machine:~$ cat -s filename > filename
```

cat f1 > filename to **paste data of f1 to filename by erasing available data in filename.**

```
user@user-virtual-machine:~$ cat f1
hello
user@user-virtual-machine:~$ cat filename
append_data
append_data2
user@user-virtual-machine:~$ cat f1 > filename
user@user-virtual-machine:~$ cat filename
hello
user@user-virtual-machine:~$ cat f1
hello
user@user-virtual-machine:~$
```

cat f1 f2 > filename

```
user@user-virtual-machine:~$ cat f2
somedata
user@user-virtual-machine:~$ cat f1 filename
hello
hello
user@user-virtual-machine:~$ cat f1 f2 > filename
user@user-virtual-machine:~$ cat filename
hello
somedata
user@user-virtual-machine:~$ cat f1 f2
hello
somedata
user@user-virtual-machine:~$
```

cat f2 >> filename to **append data of f2 to filename**

```
user@user-virtual-machine:~$ cat f2 >> filename
user@user-virtual-machine:~$ cat filename
hello
somedata
somedata
user@user-virtual-machine:~$
```

How to print n hello in terminal using shell commands?

--> echo $k (echo variable)

echo Hello$k{1..10} **({ 1..10} it prints number from 1 to 10)**

```
user@user-virtual-machine:~$ echo $k

user@user-virtual-machine:~$ echo Hello$k{1..10}
Hello Hello Hello Hello Hello Hello Hello Hello Hello Hello
```

What if you can to print hello in vertical?

--> printf 'Hello\n%.0s' {1..10} (we use %s bcoz wanted to print string and 0 in %0.s to reduce decimal value to 0)