

OSL ASSIGNMENT 1A

Name: Amar R. Dalvi

Batch: L-10

Roll No.: 33218

Title: Study of basic Linux commands

1. echo:

The echo command is used to display a line of text.

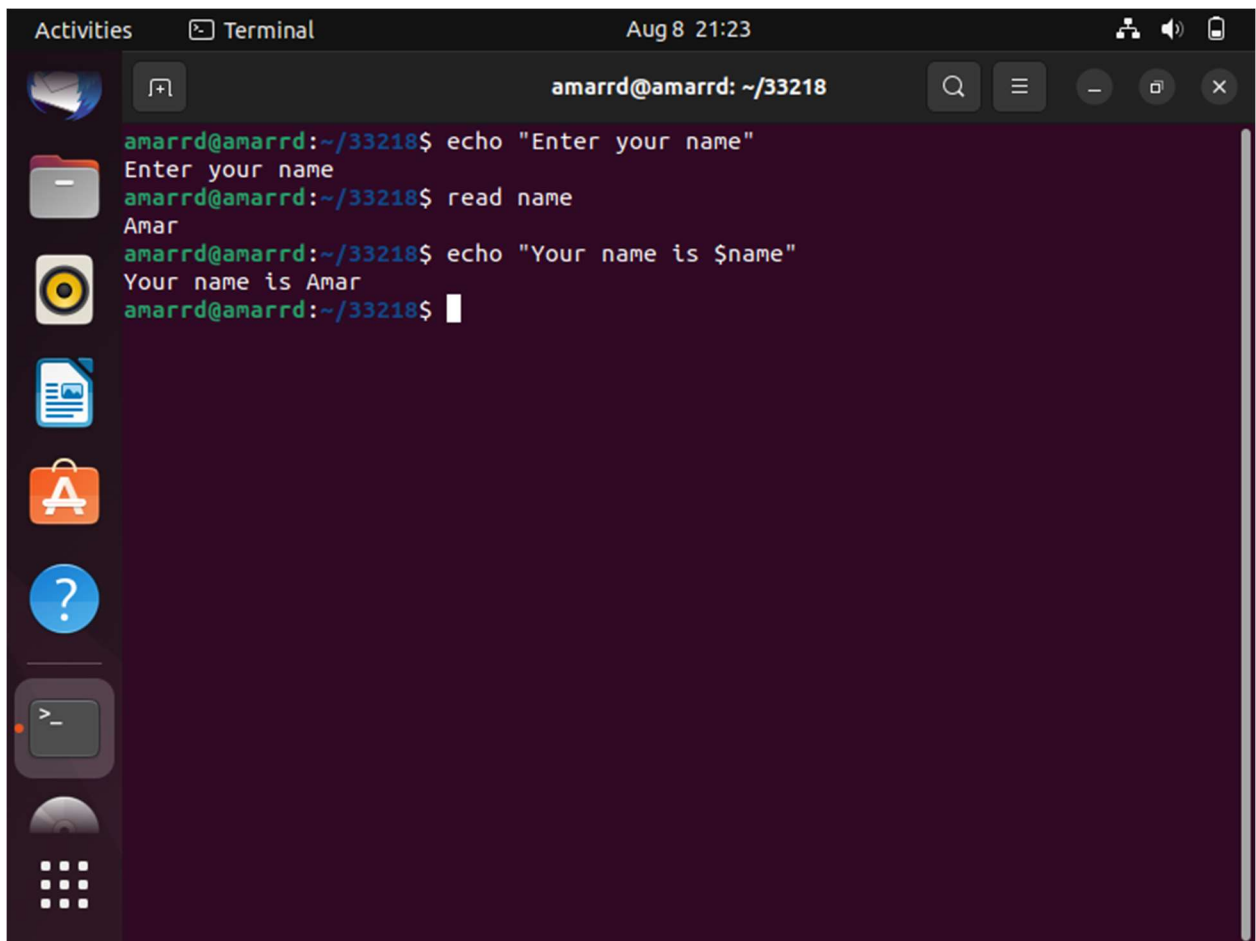
Syntax: echo [SHORT-OPTION] ... [STRING]...

Description(option):

- n do not output the trailing newline
- e enable interpretation of backslash escapes
- E disable interpretation of backslash escapes (default)

If -e is in effect, the following sequences are recognized:

- \\ backslash
- \a alert (BEL)
- \b backspace
- \c produce no further output
- \e escape
- \f form feed
- \n new line
- \r carriage return
- \t horizontal tab
- \v vertical tab



The image shows a terminal window titled "Terminal" with a timestamp of "Aug 8 21:23". The prompt is "amarrrd@amarrrd: ~/33218". The user enters the command `echo "Enter your name"`, which outputs "Enter your name". Then, the user enters `read name`, which prompts for input. The user types "Amar", and the terminal outputs "Amar". Finally, the user enters `echo "Your name is $name"`, which outputs "Your name is Amar". The terminal window has a dark purple background and a sidebar on the left with various application icons.

```
amarrrd@amarrrd:~/33218$ echo "Enter your name"
Enter your name
amarrrd@amarrrd:~/33218$ read name
Amar
amarrrd@amarrrd:~/33218$ echo "Your name is $name"
Your name is Amar
amarrrd@amarrrd:~/33218$
```

2. ls

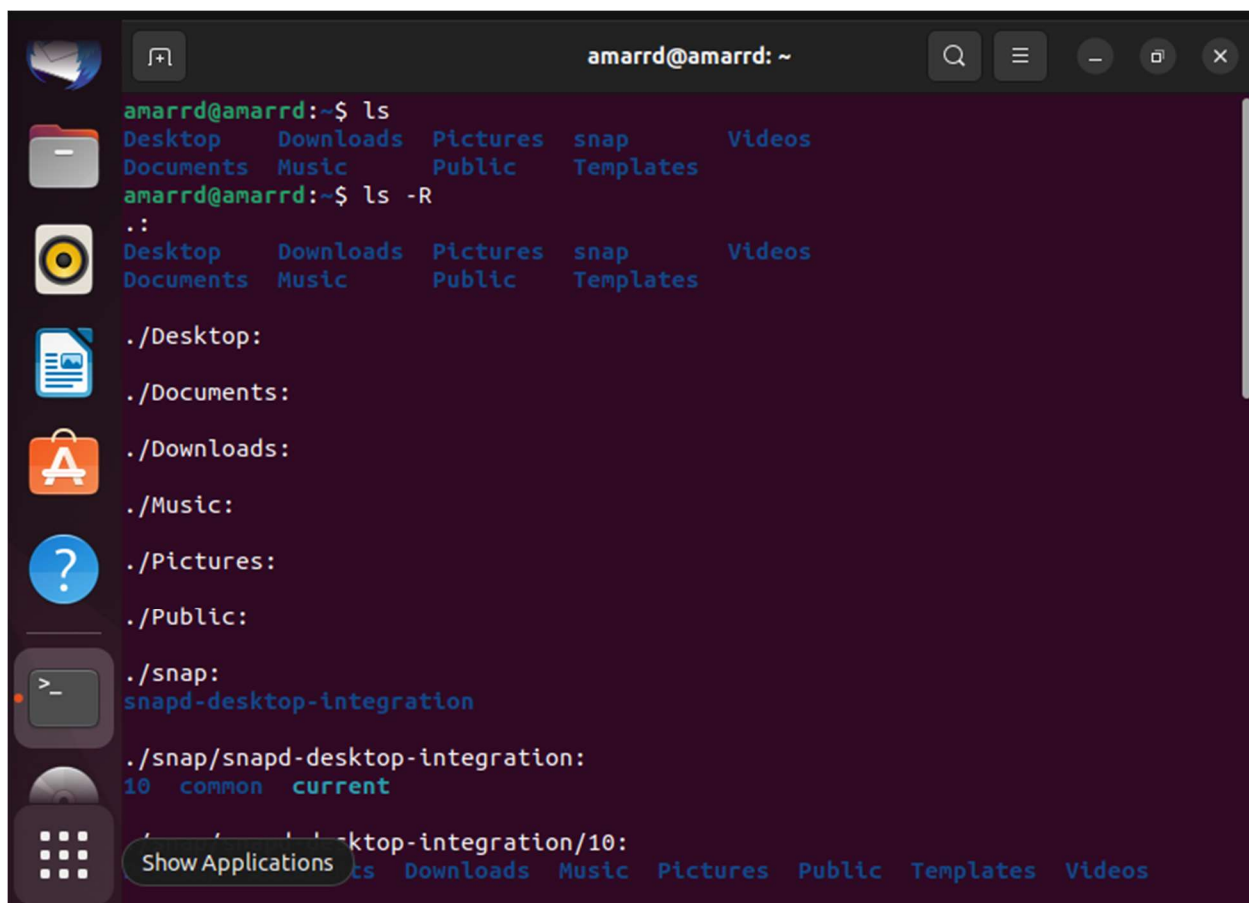
The ls command is used to list directory contents.

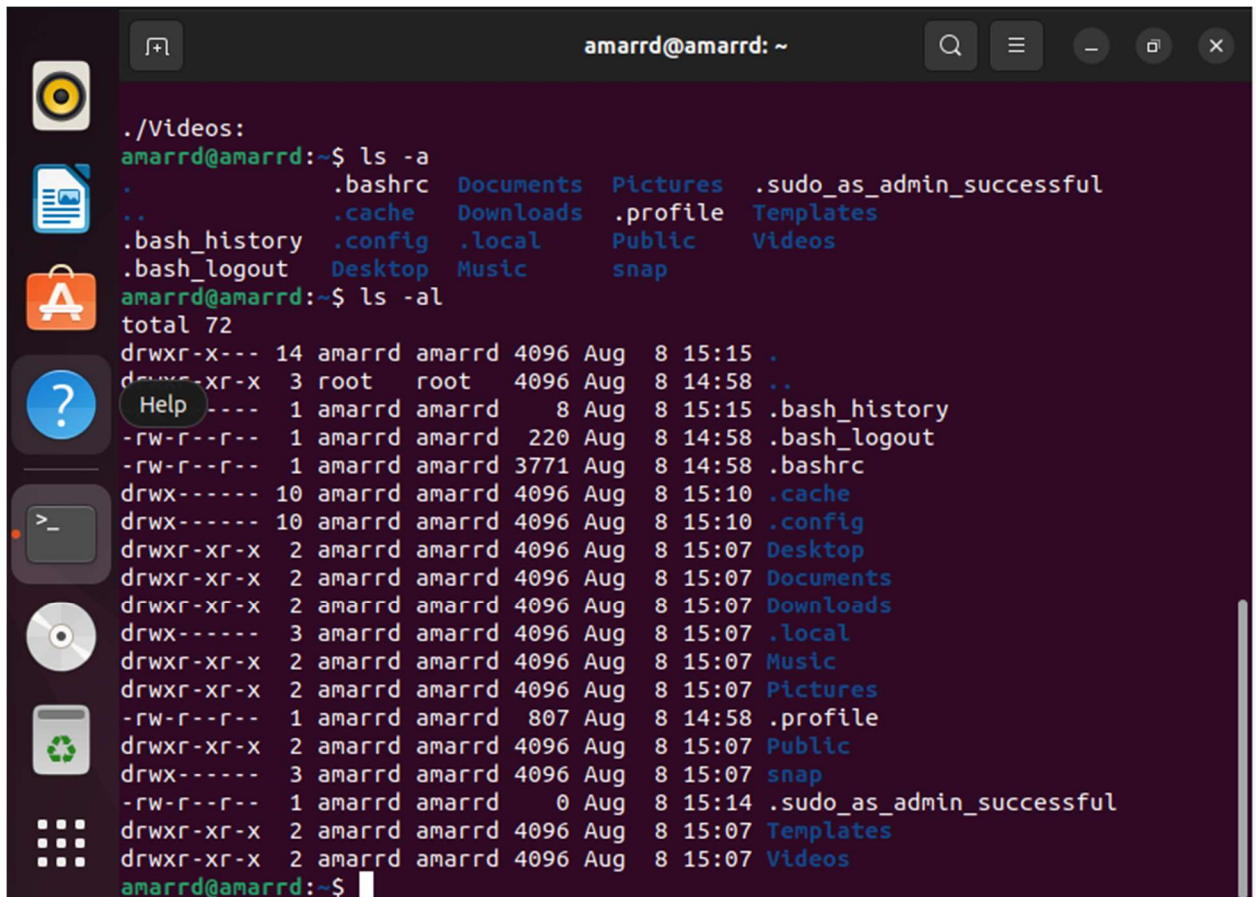
Syntax:

ls [OPTION]... [FILE]...

Description(option):

- i print index number of each file
- l use long listing format
- t sort by modification time, newest first
- 1 list one file per line.





The image shows a terminal window titled 'amarrrd@amarrrd: ~'. The left sidebar contains icons for various applications: a yellow circle with a black dot, a blue document icon, an orange shopping bag icon, a blue question mark icon, a terminal icon, a CD icon, a recycling icon, and a 3x3 grid icon. The terminal content shows the user navigating to the '/Videos' directory and running two 'ls' commands. The first command is 'ls -a', which lists hidden files and standard directories. The second command is 'ls -al', which provides a detailed listing including permissions, owner, group, size, and date for each file and directory.

```
amarrrd@amarrrd: ~  
amarrrd@amarrrd:~/Videos:  
amarrrd@amarrrd:~$ ls -a  
.  
..  
.bash_history  
.bash_logout  
.bashrc  
.cache  
.config  
.local  
Desktop  
Music  
Pictures  
Public  
snap  
.sudo_as_admin_successful  
Templates  
Videos  
amarrrd@amarrrd:~$ ls -al  
total 72  
drwxr-x--- 14 amarrd amarrd 4096 Aug  8 15:15 .  
drwxr-xr-x  3 root   root   4096 Aug  8 14:58 ..  
-rwxr-xr-x  1 amarrd amarrd    8 Aug  8 15:15 .bash_history  
-rw-r--r--  1 amarrd amarrd  220 Aug  8 14:58 .bash_logout  
-rw-r--r--  1 amarrd amarrd 3771 Aug  8 14:58 .bashrc  
drwx----- 10 amarrd amarrd 4096 Aug  8 15:10 .cache  
drwx----- 10 amarrd amarrd 4096 Aug  8 15:10 .config  
drwxr-xr-x  2 amarrd amarrd 4096 Aug  8 15:07 Desktop  
drwxr-xr-x  2 amarrd amarrd 4096 Aug  8 15:07 Documents  
drwxr-xr-x  2 amarrd amarrd 4096 Aug  8 15:07 Downloads  
drwx-----  3 amarrd amarrd 4096 Aug  8 15:07 .local  
drwxr-xr-x  2 amarrd amarrd 4096 Aug  8 15:07 Music  
drwxr-xr-x  2 amarrd amarrd 4096 Aug  8 15:07 Pictures  
-rw-r--r--  1 amarrd amarrd   807 Aug  8 14:58 .profile  
drwxr-xr-x  2 amarrd amarrd 4096 Aug  8 15:07 Public  
drwx-----  3 amarrd amarrd 4096 Aug  8 15:07 snap  
-rw-r--r--  1 amarrd amarrd    0 Aug  8 15:14 .sudo_as_admin_successful  
drwxr-xr-x  2 amarrd amarrd 4096 Aug  8 15:07 Templates  
drwxr-xr-x  2 amarrd amarrd 4096 Aug  8 15:07 Videos  
amarrrd@amarrrd:~$
```

3. cat

The cat command is used to Concatenate files and print on the standard output.

Syntax:

cat [OPTION] [FILE]

Description(option):

-E: display \$ at end of each line

-n: number all output lines

-b: number nonempty output lines, overrides -n

```
amarrrd@amarrrd: ~/33218/33218
amarrrd@amarrrd:~/33218$ cat -n demo.sh
cat: demo.sh: No such file or directory
amarrrd@amarrrd:~/33218$ cd 33218
amarrrd@amarrrd:~/33218/33218$ cat -n demo.sh
 1  #!/bin/bash
 2
 3  echo "Enter name: "
 4  read fname sname
 5  echo "Name: $fname $sname"
 6
 7  read -p'username:' user_val
 8  echo "username: $user_val"
 9
10  echo "Enter name:"
11  read -a names
12  #print names
13  echo "Name array has: ${names[0]}, ${names[1]}, ${names[2]}"
14
15  echo "Enter password:"
16  read -s password
17  echo "password: $password"
18
19  #testing variables
20
21  days=7
22  guest="Amar"
23  echo "$guest checked in $days ago"
24
amarrrd@amarrrd:~/33218/33218$
```

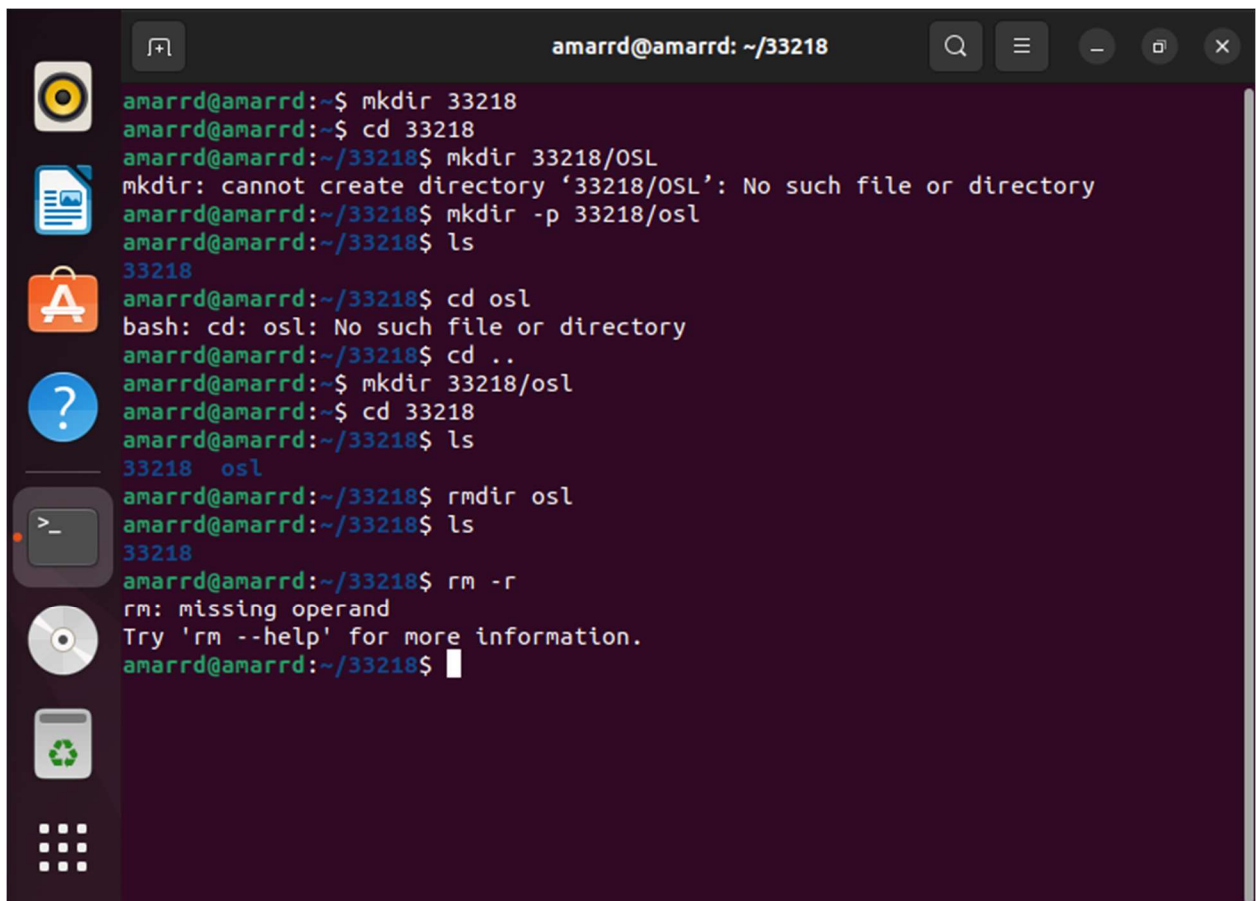
```
amarrrd@amarrrd: ~/33218/33218
 9
10  echo "Enter name:"
11  read -a names
12  #print names
13  echo "Name array has: ${names[0]}, ${names[1]}, ${names[2]}"
14
15  echo "Enter password:"
16  read -s password
17  echo "password: $password"
18
19  #testing variables
20
21  days=7
22  guest="Amar"
23  echo "$guest checked in $days ago"
24
amarrrd@amarrrd:~/33218/33218$ bash demo.sh
Enter name:
Amar Dalvi
Name: Amar Dalvi
username:amarrrd
username: amarrrd
Enter name:
Yash Amar Vaishnavi
Name array has: Yash, Amar, Vaishnavi
Enter password:
password: amar
Amar checked in 7 ago
amarrrd@amarrrd:~/33218/33218$
```

4. mkdir

Use mkdir command to make a new directory — if you type mkdir Music it will create a directory called Music.

6. rmdir

If you need to delete a directory, use the rmdir command. However, rmdir only allows you to delete empty directories.

A terminal window titled 'amarrrd@amarrrd: ~/33218' with standard window controls. The terminal shows a series of commands and their outputs. The user creates a directory '33218', enters it, and attempts to create '33218/OSL', which fails. They then use 'mkdir -p 33218/osl' to create it successfully. After listing the contents, they attempt to change to 'osl' but fail. They then change to the parent directory and create '33218/osl' again. After listing, they use 'rmdir osl' to delete it. Finally, they attempt to remove the directory with 'rm -r' but receive an error for a missing operand.

```
amarrrd@amarrrd:~$ mkdir 33218
amarrrd@amarrrd:~$ cd 33218
amarrrd@amarrrd:~/33218$ mkdir 33218/OSL
mkdir: cannot create directory '33218/OSL': No such file or directory
amarrrd@amarrrd:~/33218$ mkdir -p 33218/osl
amarrrd@amarrrd:~/33218$ ls
33218
amarrrd@amarrrd:~/33218$ cd osl
bash: cd: osl: No such file or directory
amarrrd@amarrrd:~/33218$ cd ..
amarrrd@amarrrd:~$ mkdir 33218/osl
amarrrd@amarrrd:~$ cd 33218
amarrrd@amarrrd:~/33218$ ls
33218  osl
amarrrd@amarrrd:~/33218$ rmdir osl
amarrrd@amarrrd:~/33218$ ls
33218
amarrrd@amarrrd:~/33218$ rm -r
rm: missing operand
Try 'rm --help' for more information.
amarrrd@amarrrd:~/33218$
```

7. touch

The touch command is used to create, change, and modify timestamps of a file

Syntax:

```
touch [OPTION] [file name]
```

Description(option):

-c: used to check whether a file is created or not.

-m: used to change the modification time.

8. locate

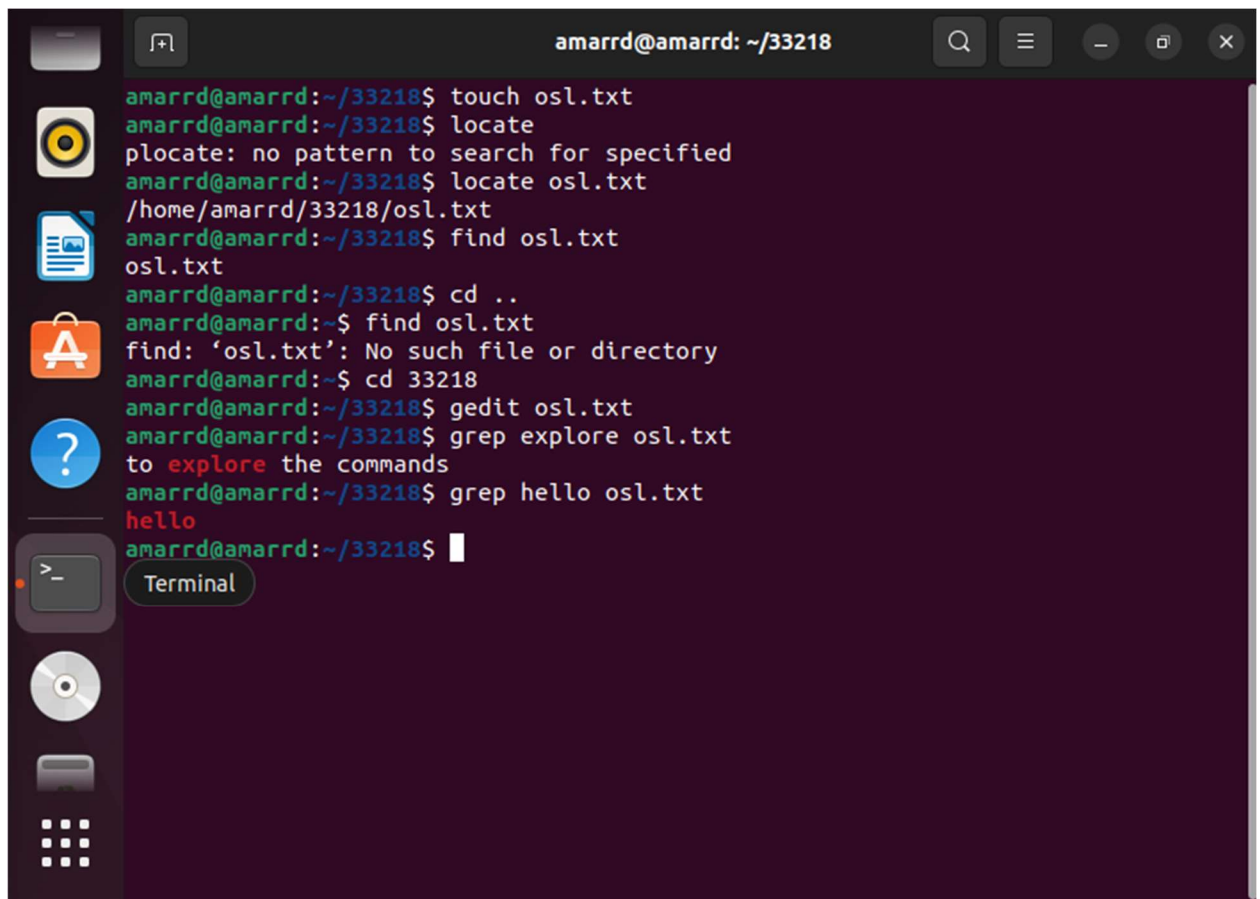
You can use this command to locate a file, just like the search command in Windows. What's more, using the -i argument along with this command will make it caseinsensitive, so you can search for a file even if you don't remember its

9. find

Similar to the locate command, using find also searches for files and directories. The difference is, you use the find command to locate files within a given directory. As an example, find /home/ -name notes.txt command will search for a file called notes.txt within the home directory

10.grep command

Another basic Linux command that is undoubtedly helpful for everyday use is grep. It lets you search through all the text in a given file



```
amarrrd@amarrrd: ~/33218
amarrrd@amarrrd:~/33218$ touch osl.txt
amarrrd@amarrrd:~/33218$ locate
plocate: no pattern to search for specified
amarrrd@amarrrd:~/33218$ locate osl.txt
/home/amarrrd/33218/osl.txt
amarrrd@amarrrd:~/33218$ find osl.txt
osl.txt
amarrrd@amarrrd:~/33218$ cd ..
amarrrd@amarrrd:~$ find osl.txt
find: 'osl.txt': No such file or directory
amarrrd@amarrrd:~$ cd 33218
amarrrd@amarrrd:~/33218$ gedit osl.txt
amarrrd@amarrrd:~/33218$ grep explore osl.txt
to explore the commands
amarrrd@amarrrd:~/33218$ grep hello osl.txt
hello
amarrrd@amarrrd:~/33218$
```

11. df command

Use df command to get a report on the system's disk space usage, shown in percentage and KBs. If you want to see the report in megabytes, type df -m.

12. du command

If you want to check how much space a file or a directory takes, the du (Disk Usage) command is the answer. However, the disk usage summary will show disk block numbers instead of the usual size format. If you want to see it in bytes, kilobytes, and megabytes, add the -h argument to the command line.

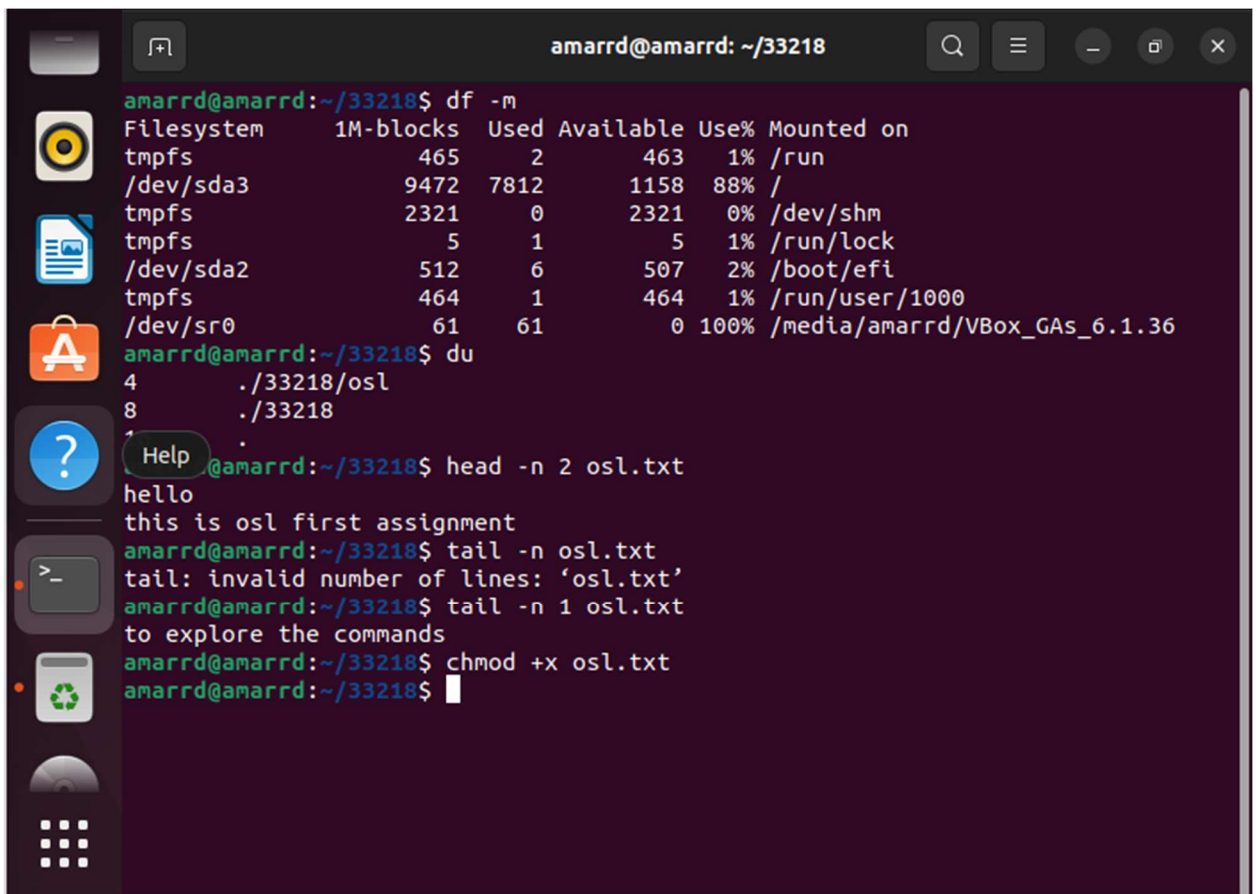
13. head command

The head command is used to view the first lines of any text file. By default, it will show the first ten lines, but you can change this number to your liking.

For example, if you only want to show the first five lines, type `head -n 5 filename.ext`.

14.tail command

This one has a similar function to the head command, but instead of showing the first lines, the tail command will display the last ten lines of a text file. For example, `tail -n filename.ext`.



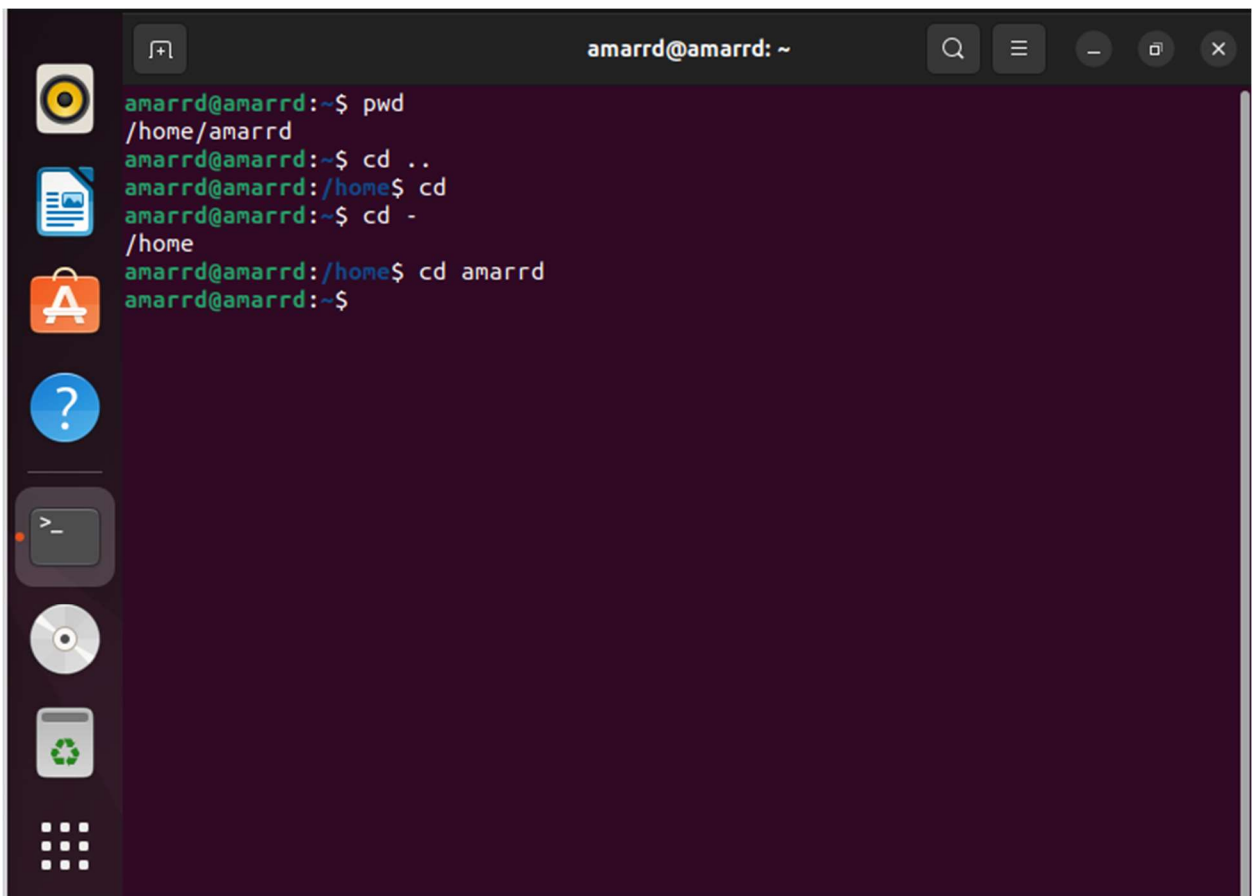
```
amarrrd@amarrrd: ~/33218
amarrrd@amarrrd:~/33218$ df -h
Filesystem      1M-blocks    Used Available Use% Mounted on
tmpfs            465          2        463   1% /run
/dev/sda3       9472       7812       1158  88% /
tmpfs           2321          0       2321   0% /dev/shm
tmpfs            5            1          5   1% /run/lock
/dev/sda2        512          6        507   2% /boot/efi
tmpfs           464          1        464   1% /run/user/1000
/dev/sr0         61          61          0 100% /media/amarrrd/VBox_GAs_6.1.36
amarrrd@amarrrd:~/33218$ du
4      ./33218/osl
8      ./33218
amarrrd@amarrrd:~/33218$ head -n 2 osl.txt
hello
this is osl first assignment
amarrrd@amarrrd:~/33218$ tail -n osl.txt
tail: invalid number of lines: 'osl.txt'
amarrrd@amarrrd:~/33218$ tail -n 1 osl.txt
to explore the commands
amarrrd@amarrrd:~/33218$ chmod +x osl.txt
amarrrd@amarrrd:~/33218$
```

15. pwd command

Use the `pwd` command to find out the path of the current working directory (folder) you're in. The command will return an absolute (full) path, which is basically a path of all the directories that starts with a forward slash (/). An example of an absolute path is `/home/username`

16. cd command

To navigate through the Linux files and directories, use the `cd` command. It requires either the full path or the name of the directory, depending on the current working directory that you're in.



```
amarrrd@amarrrd: ~  
amarrrd@amarrrd:~$ pwd  
/home/amarrrd  
amarrrd@amarrrd:~$ cd ..  
amarrrd@amarrrd:/home$ cd  
amarrrd@amarrrd:~$ cd -  
/home  
amarrrd@amarrrd:/home$ cd amarrd  
amarrrd@amarrrd:~$
```

The image shows a terminal window titled "amarrrd@amarrrd: ~". The terminal output shows the following sequence of commands and their results:

- `amarrrd@amarrrd:~$ pwd` returns `/home/amarrrd`
- `amarrrd@amarrrd:~$ cd ..` changes the directory to `/home`
- `amarrrd@amarrrd:/home$ cd` changes the directory back to `~`
- `amarrrd@amarrrd:~$ cd -` returns `/home`
- `amarrrd@amarrrd:/home$ cd amarrd` changes the directory to `~`
- `amarrrd@amarrrd:~$` shows the prompt after the final command.

