OSL ASSIGNMENT 1A

Name: Amar R. Dalvi

Batch: L-10

Roll No.: 33218

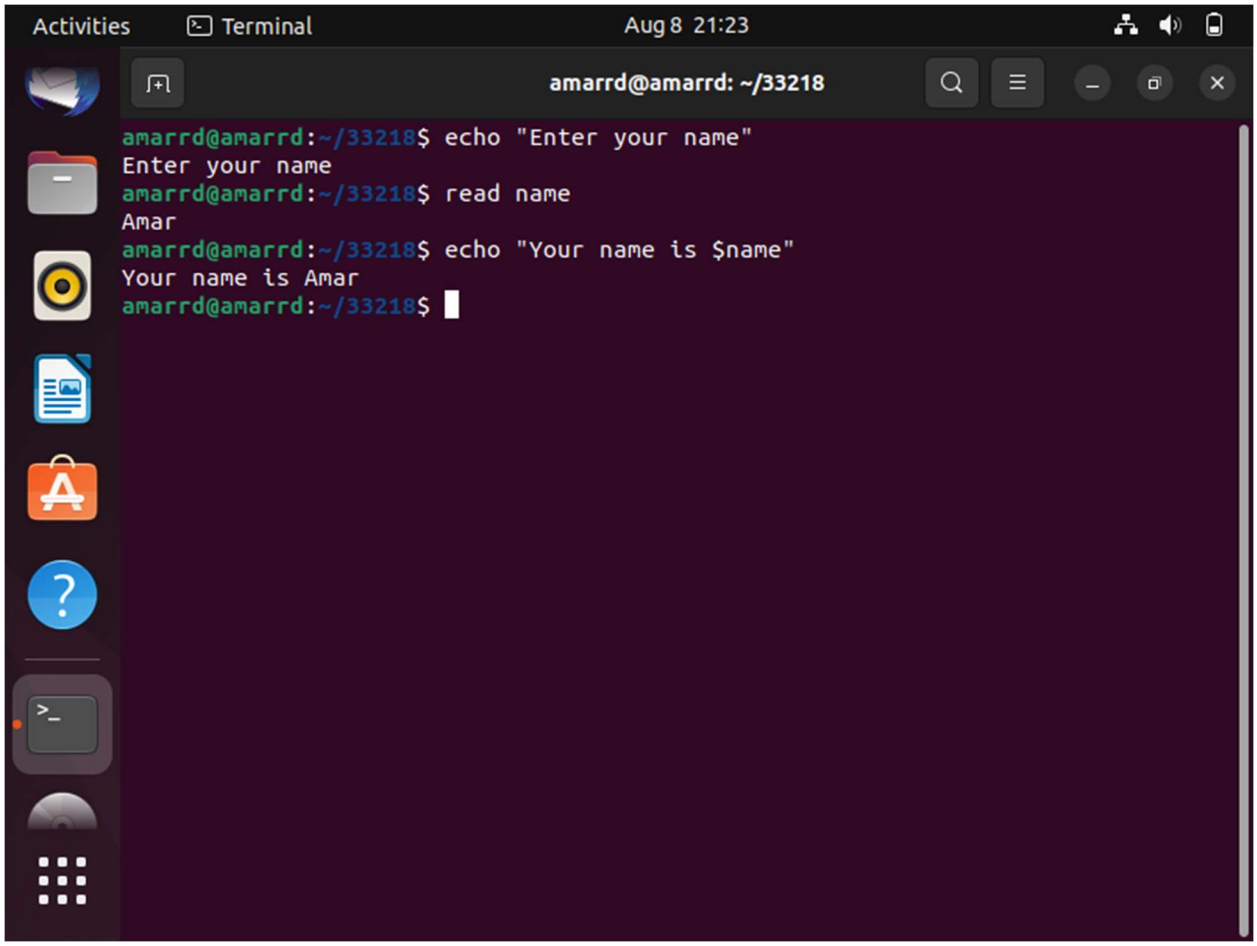
Title: Study of basic Linux commands

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)

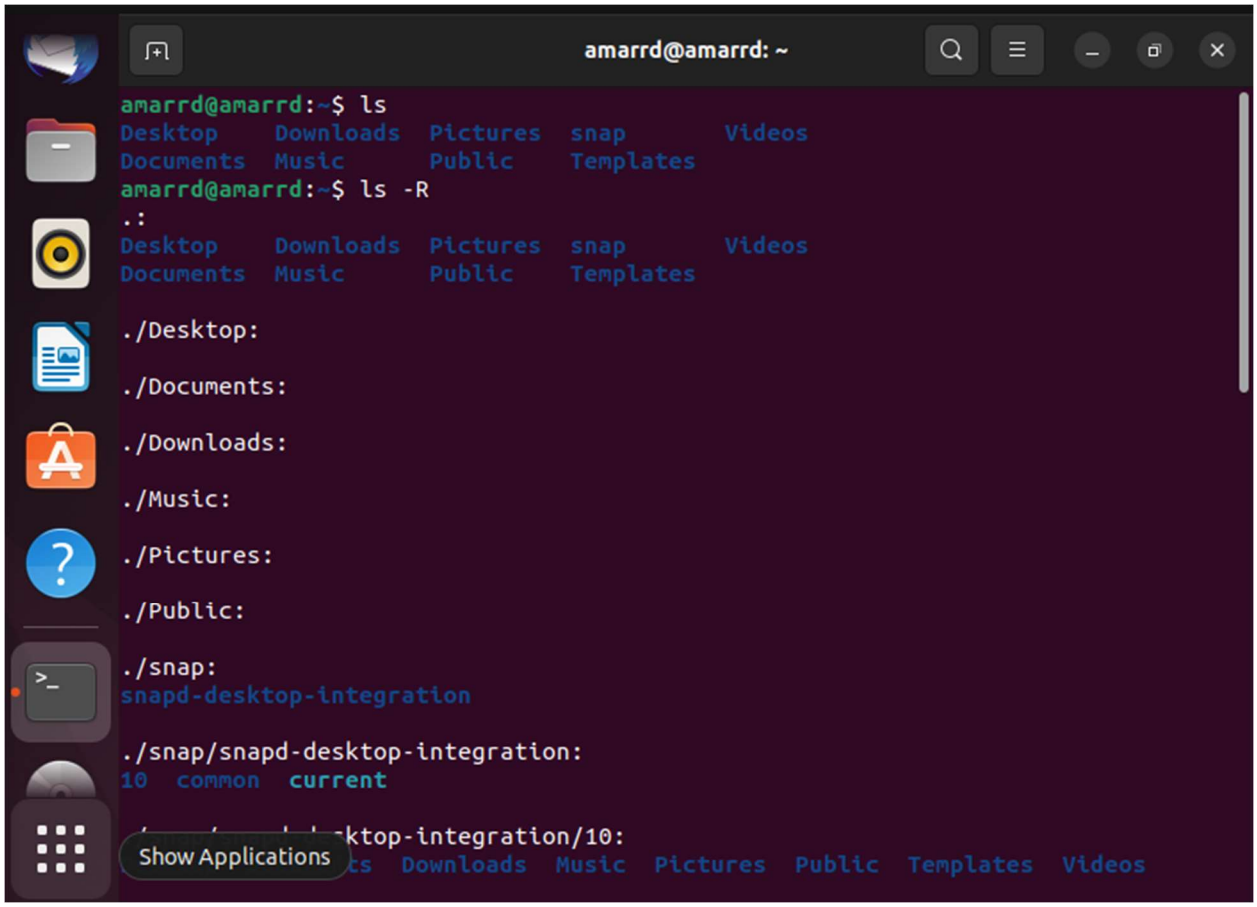
   If -e is in effect, the following sequences are recognized:
   \\ backslash
   \a alert (BEL)
   \b backspace
   \c produce no further output
   \e escape
   \f form feed
   \n new line
   \r carriage return
   \t horizontal tab
   \v vertical tab

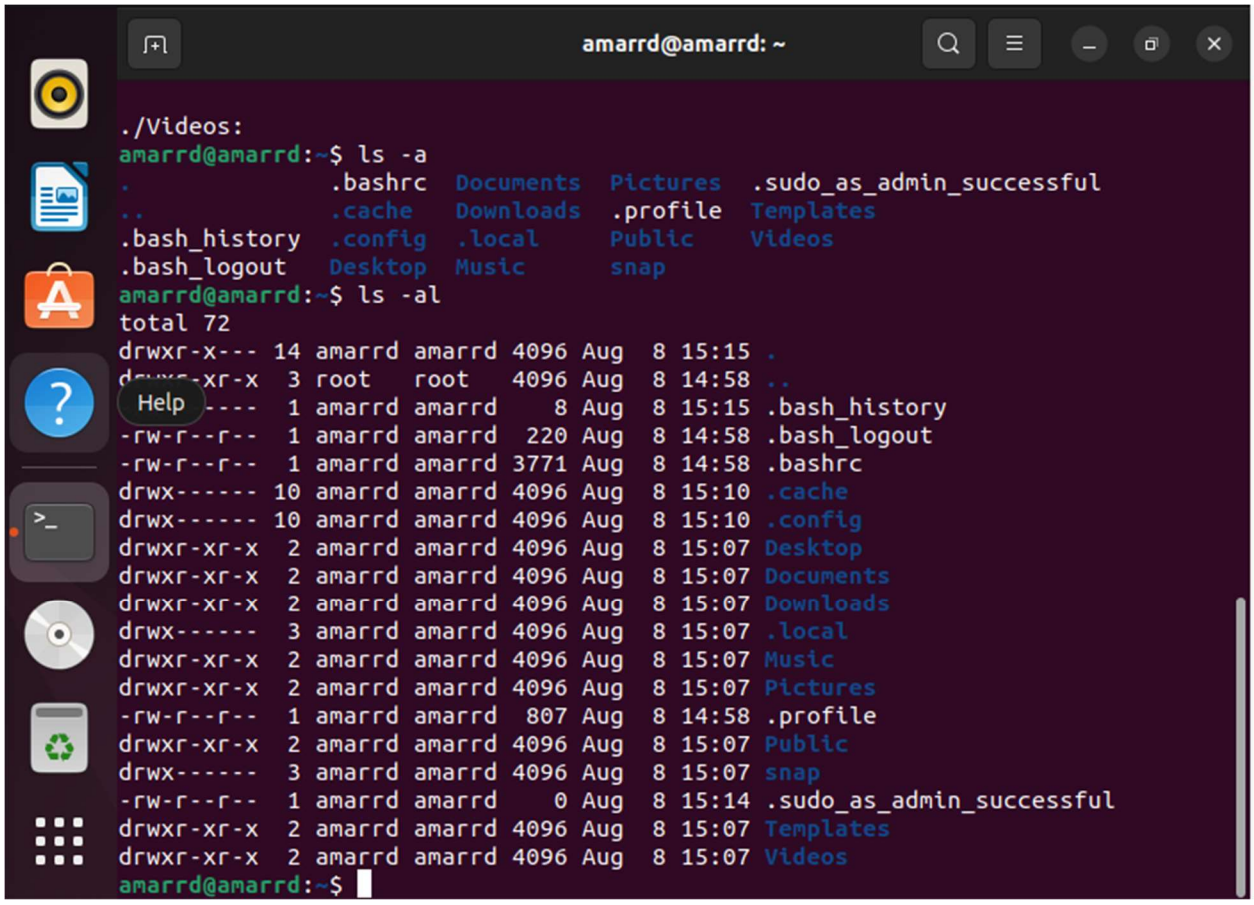


2. ls

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.

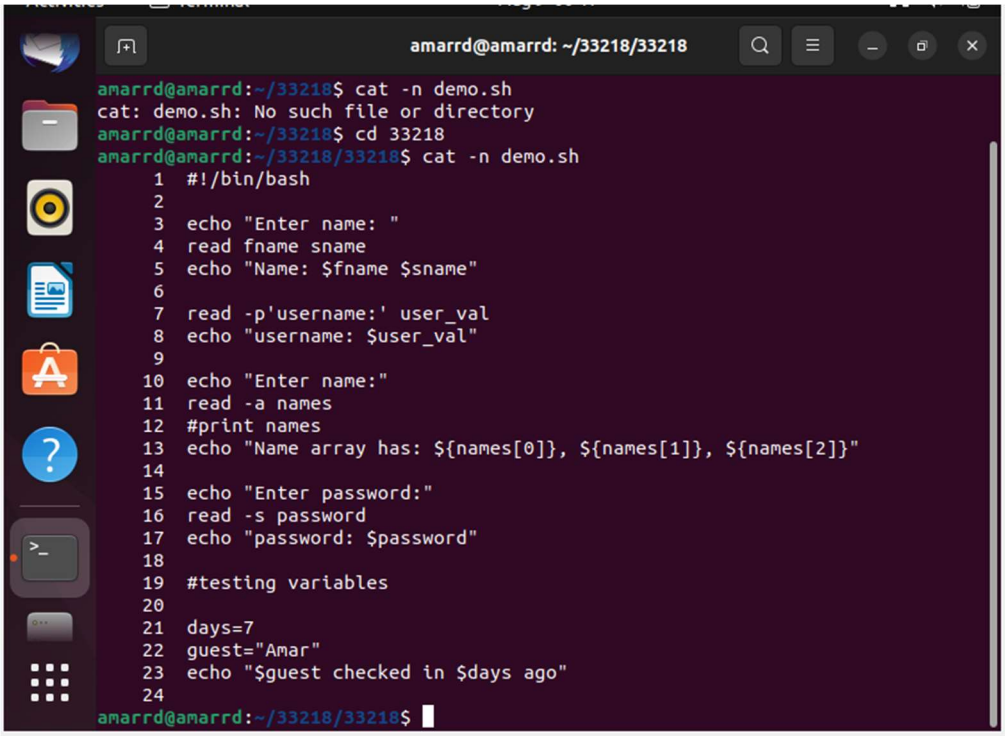

amarrd@amarrd: ~
amarrd@amarrd:~$ ls
Desktop Downloads Pictures snap Videos
Documents Music Public Templates
amarrd@amarrd:~$ ls -R
.:
Desktop Downloads Pictures snap Videos
Documents Music Public Templates
./Desktop:
./Documents:
./Downloads:
./Music:
./Pictures:
./Public:
./snap:
snapd-desktop-integration
./snap/snapd-desktop-integration:
10 common current
Downloads Music Pictures Public Templates Videos
Show Applications

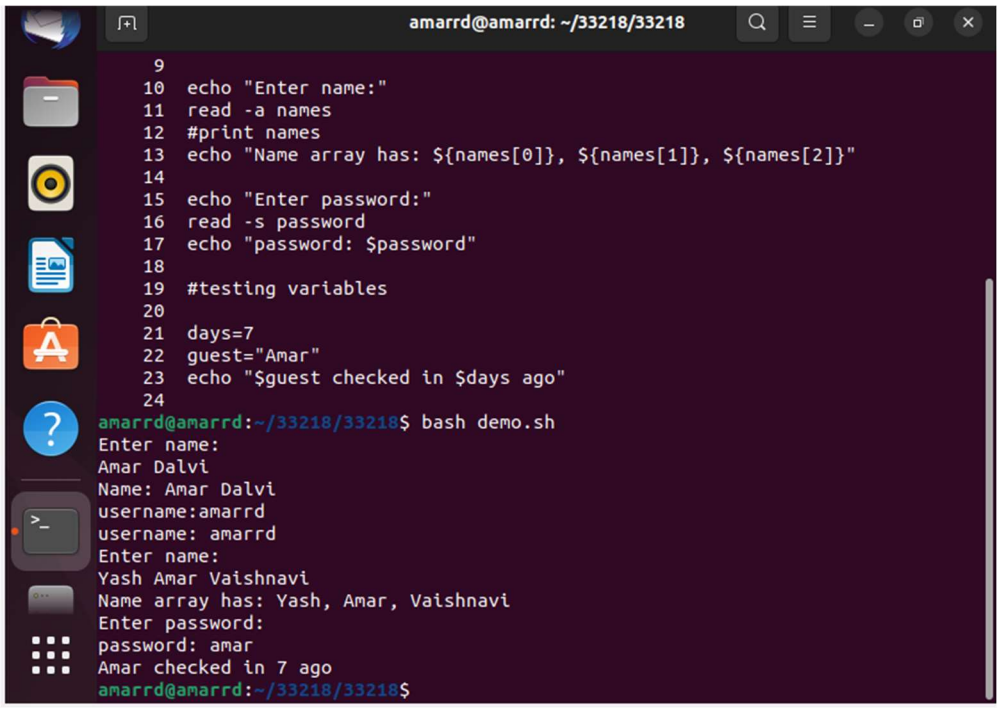


3. **cat**

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


amarrd@amarrd: ~/33218/33218
amarrd@amarrd:~/33218$ cat -n demo.sh
cat: demo.sh: No such file or directory
amarrd@amarrd:~/33218$ cd 33218
amarrd@amarrd:~/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 $days ago"
24
amarrd@amarrd:~/33218/33218$

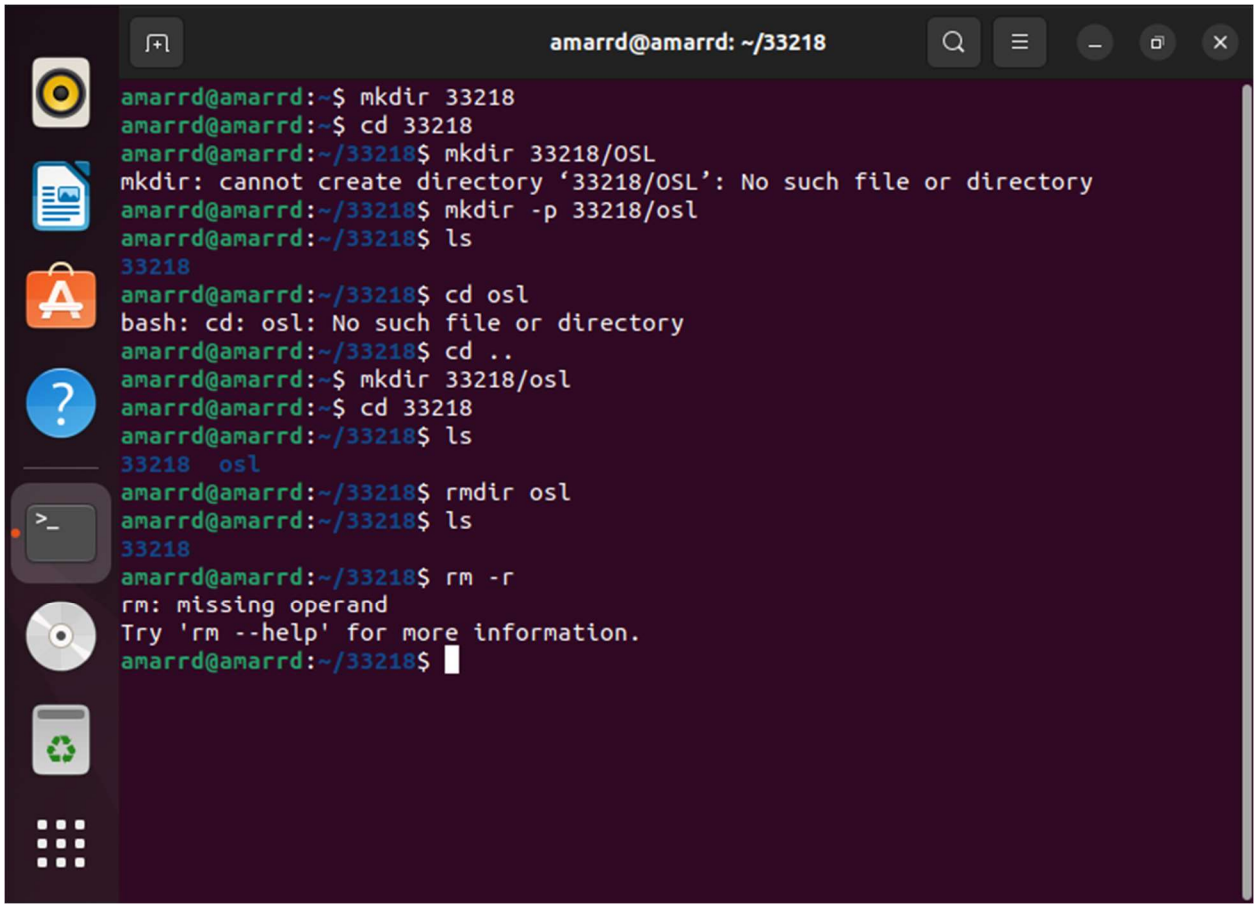

amarrd@amarrd: ~/33218/33218
9
10 echo "Enter name:"
11 read -a names
12 #print names
13 echo "Name array has: ${names[0]}, ${names[1]}, ${names[2]}"
14
15 echo "Enter password:"
16 read -s password
17 echo "password: $password"
18
19 #testing variables
20
21 days=7
22 guest="Amar"
23 echo "$guest checked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



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**

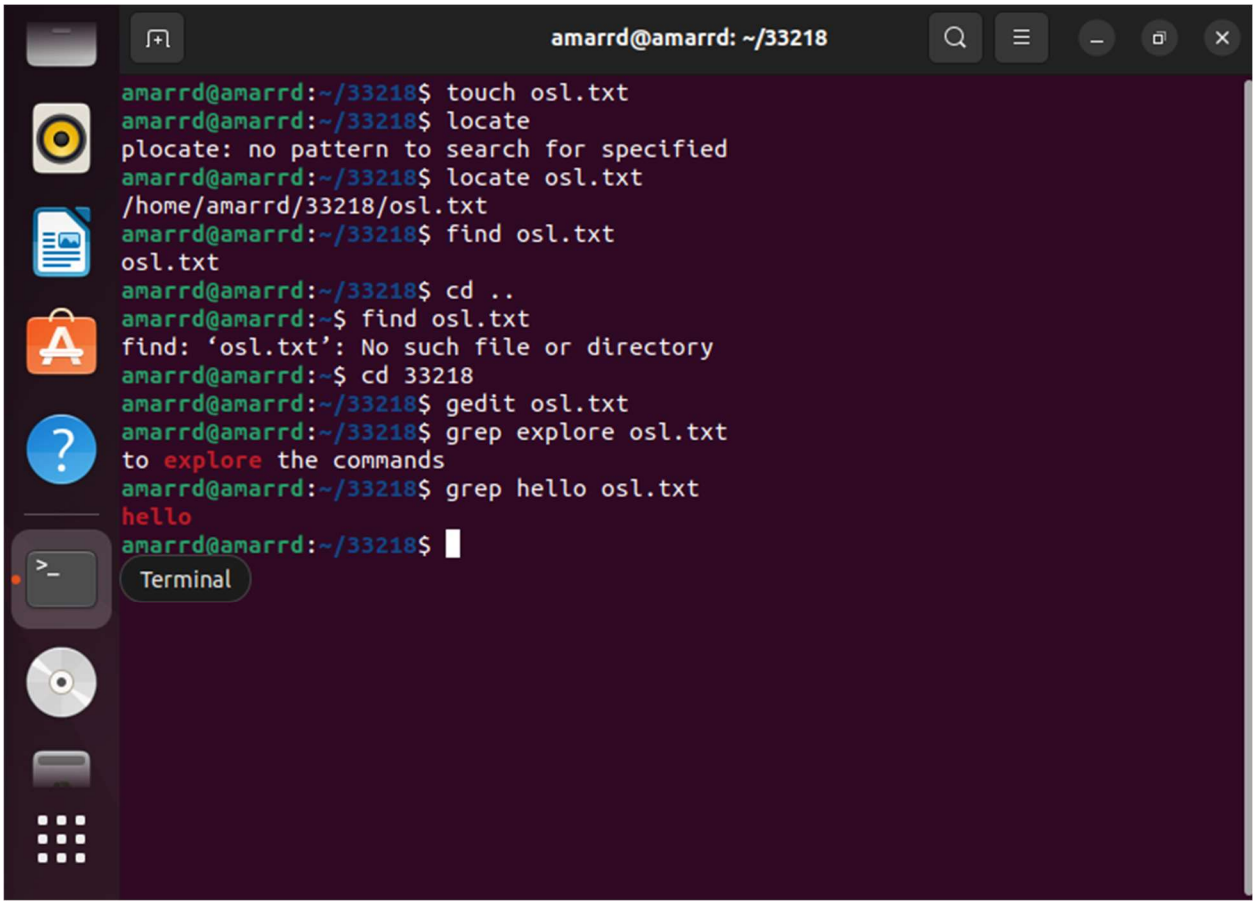
   You can use this command to locate a file, just like the search command in Windows. What's more, using the -i argument along with this command will make it caseinsensitive, so you can search for a file even if you don't remember its

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



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.

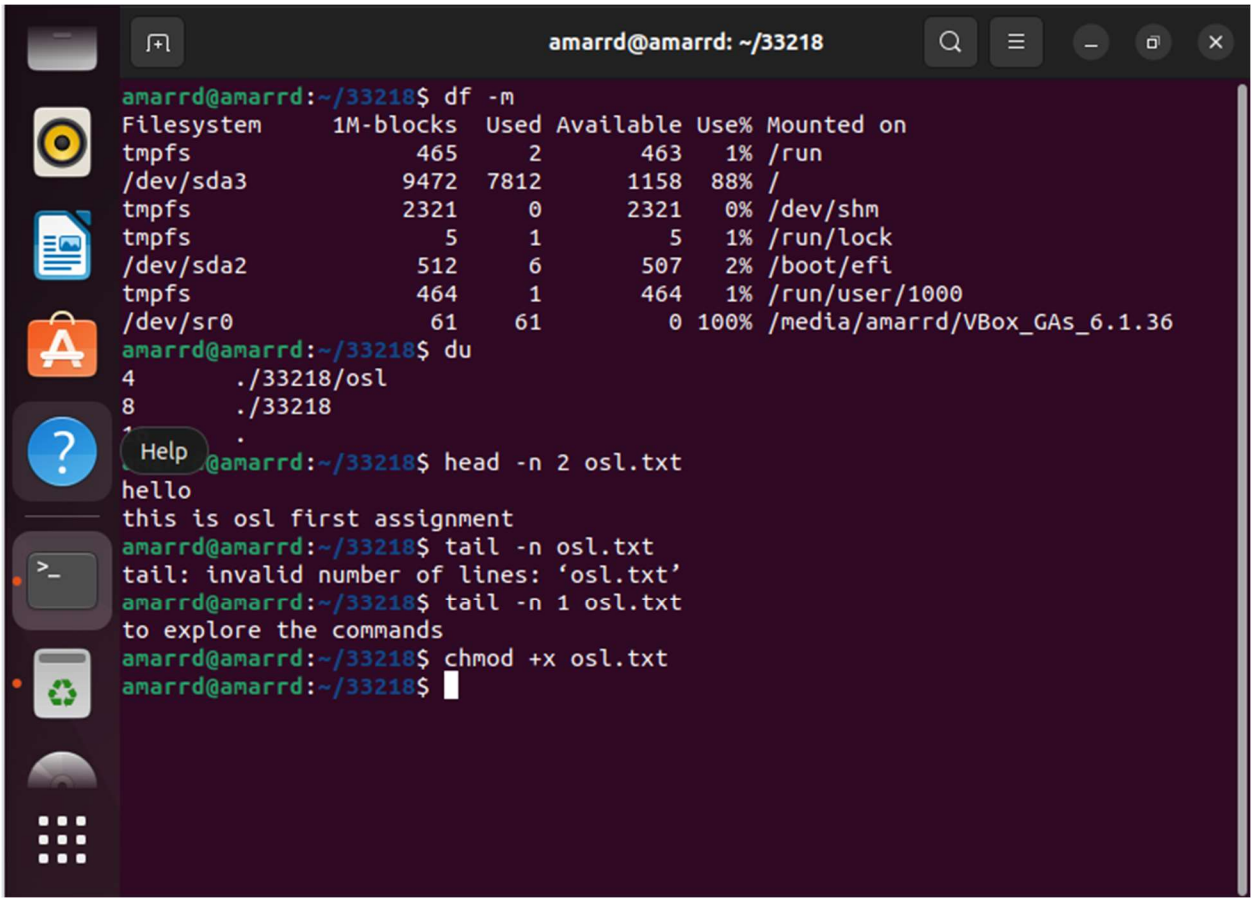
13. head command

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.



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.