



Linux Command Line

Mohan Kumar Pulibanti

Download and Install Oracle VM Virtual Box

Download link: <https://www.oracle.com/virtualization/technologies/vm/downloads/virtualbox-downloads.html>

Installation Video Reference: https://youtu.be/6_irBgXcK_Q

Setting up Linux Virtual Machine (ubuntu distribution):

ubuntu Desktop download link: <https://ubuntu.com/download/desktop>

Ubuntu setup Video reference: <https://youtu.be/GKpoU0qNOTQ>

Launching Linux for the First Time:

Video reference: <https://youtu.be/3Xm74sHZnVM>

Our First Commands:

Shortcuts:

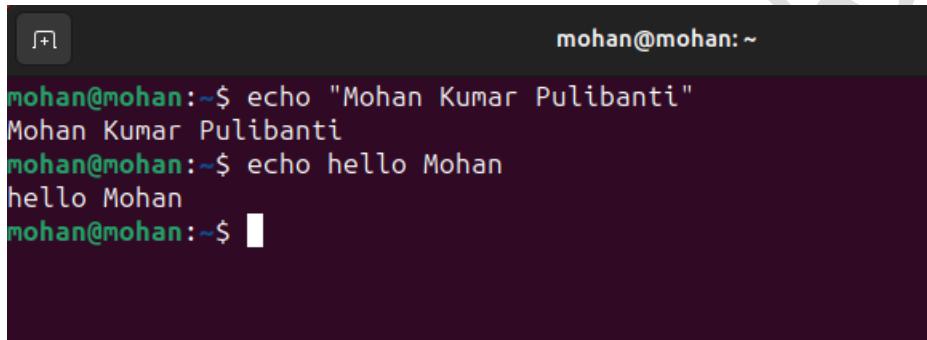
Open terminal: **ctrl+alt+t**

Clear Terminal: **ctrl+l**

Close Terminal: **ctrl+d**

Commands:

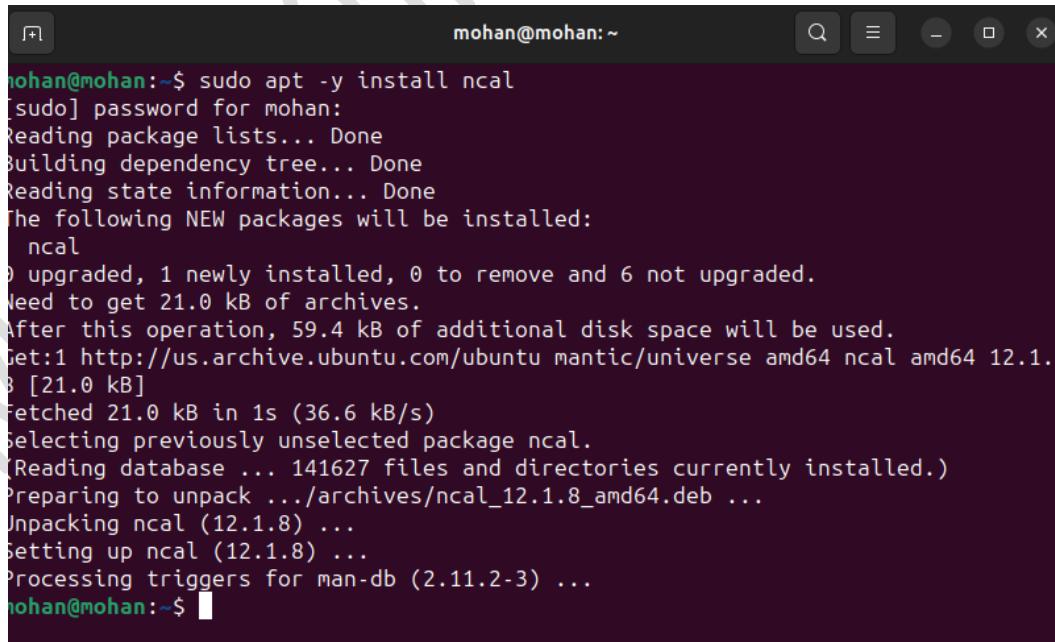
- 1) **echo** – prints what we give as input.



```
mohan@mohan:~$ echo "Mohan Kumar Pulibanti"
Mohan Kumar Pulibanti
mohan@mohan:~$ echo hello Mohan
hello Mohan
mohan@mohan:~$
```

- 2) **cal** – cal command is no longer included in the latest versions of Ubuntu

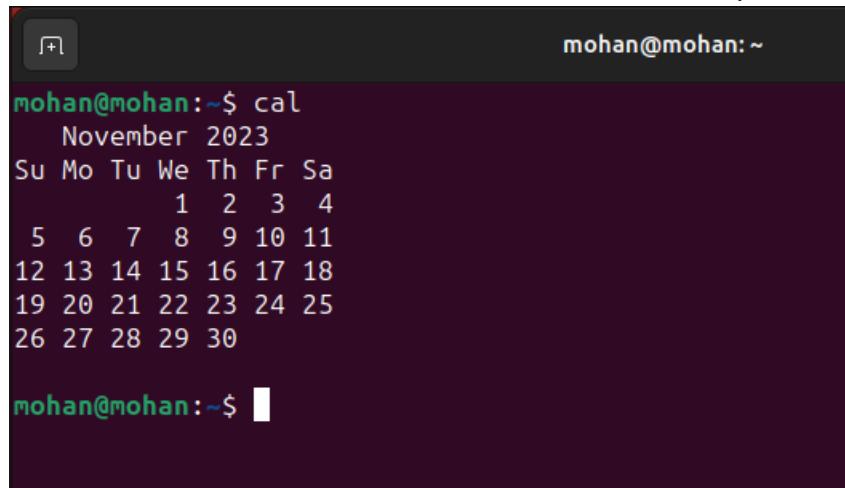
➤ To Install Cal command – “**sudo apt -y install ncal**”



```
mohan@mohan:~$ sudo apt -y install ncal
[sudo] password for mohan:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  ncal
0 upgraded, 1 newly installed, 0 to remove and 6 not upgraded.
Need to get 21.0 kB of archives.
After this operation, 59.4 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu mantic/universe amd64 ncal amd64 12.1.8 [21.0 kB]
Fetched 21.0 kB in 1s (36.6 kB/s)
Selecting previously unselected package ncal.
(Reading database ... 141627 files and directories currently installed.)
Preparing to unpack .../archives/ncal_12.1.8_amd64.deb ...
Unpacking ncal (12.1.8) ...
Setting up ncal (12.1.8) ...
Processing triggers for man-db (2.11.2-3) ...
mohan@mohan:~$
```

- Cal will show the calendar on the screen.

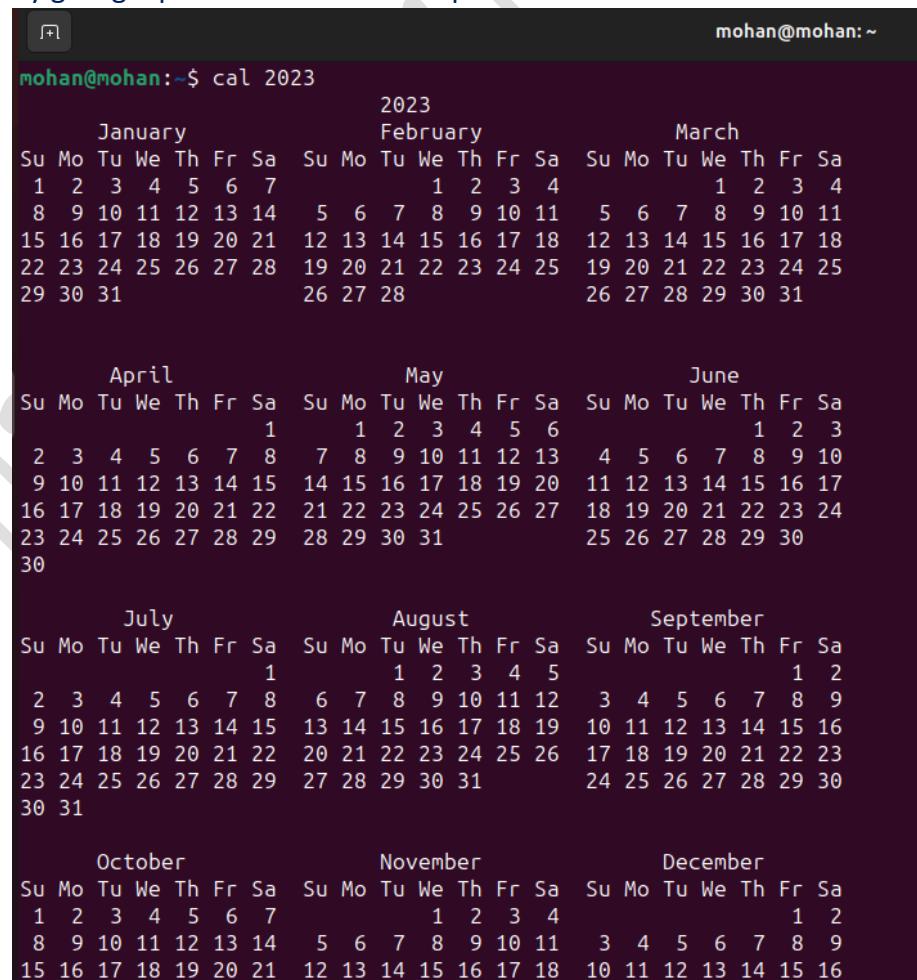
cal – will show the calendar for the current month and year.



```
mohan@mohan:~$ cal
November 2023
Su Mo Tu We Th Fr Sa
      1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30
```

- Cal command can be customized (from the default behavior) by giving input or by giving options

- By giving input – **cal 2023** – this inputs calendar for 2023



```
mohan@mohan:~$ cal 2023
2023
January          February          March
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7   1  2  3  4   1  2  3  4
 8  9 10 11 12 13 14   5  6  7  8  9 10 11   5  6  7  8  9 10 11
15 16 17 18 19 20 21   12 13 14 15 16 17 18   12 13 14 15 16 17 18
22 23 24 25 26 27 28   19 20 21 22 23 24 25   19 20 21 22 23 24 25
29 30 31                  26 27 28                  26 27 28 29 30 31

April            May              June
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
                    1   1  2  3  4  5  6   1   2  3  4  5  6   1  2  3
 2  3  4  5  6  7  8   7  8  9 10 11 12 13   4  5  6  7  8  9 10
 9 10 11 12 13 14 15   14 15 16 17 18 19 20   11 12 13 14 15 16 17
16 17 18 19 20 21 22   21 22 23 24 25 26 27   18 19 20 21 22 23 24
23 24 25 26 27 28 29   28 29 30 31                  25 26 27 28 29 30
30

July            August           September
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
                    1   1  2  3  4  5   1   2  3  4  5   1  2
 2  3  4  5  6  7  8   6  7  8  9 10 11 12   3  4  5  6  7  8  9
 9 10 11 12 13 14 15   13 14 15 16 17 18 19   10 11 12 13 14 15 16
16 17 18 19 20 21 22   20 21 22 23 24 25 26   17 18 19 20 21 22 23
23 24 25 26 27 28 29   27 28 29 30 31                  24 25 26 27 28 29 30
30 31

October         November        December
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
                    1   1  2  3  4  5   1   2  3  4   1  2
 8  9 10 11 12 13 14   5  6  7  8  9 10 11   3  4  5  6  7  8  9
15 16 17 18 19 20 21   12 13 14 15 16 17 18   10 11 12 13 14 15 16
```

- Passing options to cal command

cal -y : prints the calendar for the current year

```
mohan@mohan:~$ cal -y
          2023
January           February          March
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7     1  2  3  4       1  2  3  4
 8  9 10 11 12 13 14    5  6  7  8  9 10 11   5  6  7  8  9 10 11
15 16 17 18 19 20 21   12 13 14 15 16 17 18   12 13 14 15 16 17 18
22 23 24 25 26 27 28   19 20 21 22 23 24 25   19 20 21 22 23 24 25
29 30 31                  26 27 28               26 27 28 29 30 31

April            May              June
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
                      1     1  2  3  4  5  6       1  2  3
 2  3  4  5  6  7  8     7  8  9 10 11 12 13   4  5  6  7  8  9 10
 9 10 11 12 13 14 15   14 15 16 17 18 19 20   11 12 13 14 15 16 17
16 17 18 19 20 21 22   21 22 23 24 25 26 27   18 19 20 21 22 23 24
23 24 25 26 27 28 29   28 29 30 31             25 26 27 28 29 30
30

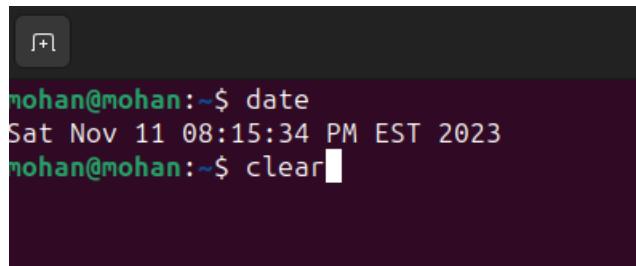
July            August          September
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
                      1     1  2  3  4  5       1  2
 2  3  4  5  6  7  8     6  7  8  9 10 11 12   3  4  5  6  7  8  9
 9 10 11 12 13 14 15   13 14 15 16 17 18 19   10 11 12 13 14 15 16
16 17 18 19 20 21 22   20 21 22 23 24 25 26   17 18 19 20 21 22 23
23 24 25 26 27 28 29   27 28 29 30 31             24 25 26 27 28 29 30
30 31

October         November        December
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7     1  2  3  4       1  2
 8  9 10 11 12 13 14    5  6  7  8  9 10 11   3  4  5  6  7  8  9
15 16 17 18 19 20 21   12 13 14 15 16 17 18   10 11 12 13 14 15 16
```

➤ **date** : prints todays date, time, time zone and year

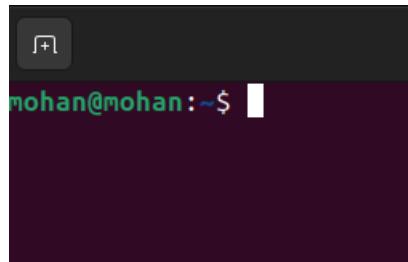
```
mohan@mohan:~$ date
Sat Nov 11 08:15:34 PM EST 2023
mohan@mohan:~$
```

- **clear** – to clear the command line.



```
mohan@mohan:~$ date  
Sat Nov 11 08:15:34 PM EST 2023  
mohan@mohan:~$ clear
```

Output of clear command:



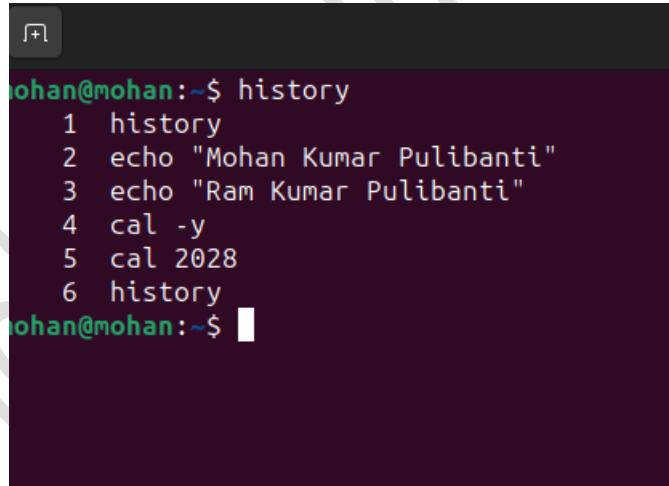
```
mohan@mohan:~$
```

- **Keyboard keys can be used to traverse through the commands that we used previously.**



Arrows_to_Traverse.mp4

- **history** – is to see all the commands we used in history along with line numbers.



```
mohan@mohan:~$ history  
1 history  
2 echo "Mohan Kumar Pulibanti"  
3 echo "Ram Kumar Pulibanti"  
4 cal -y  
5 cal 2028  
6 history  
mohan@mohan:~$
```

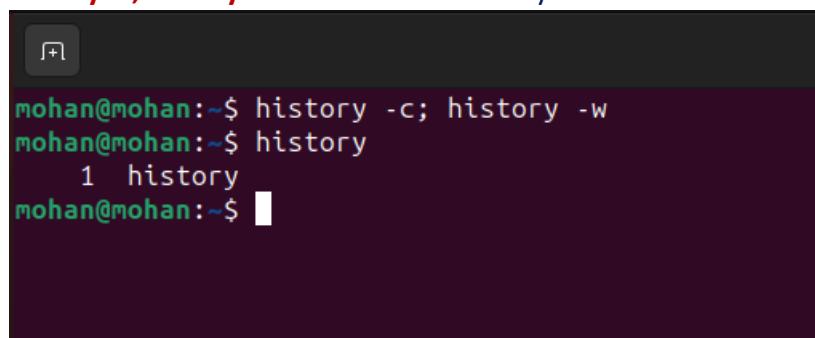
- To execute a particular command with the number from the history (when we have 100's of commands in the history we can't use keyboard arrow keys to traverse through them, in that case this will be useful)
Command → !2 (!<Command Number>)

```
mohan@mohan:~$ history
1 history
2 echo "Mohan Kumar Pulibanti"
3 echo "Ram Kumar Pulibanti"
4 cal -y
5 cal 2028
6 history
mohan@mohan:~$ !2
echo "Mohan Kumar Pulibanti"
Mohan Kumar Pulibanti
mohan@mohan:~$
```

- If you want to execute the most recent command
Command: !!

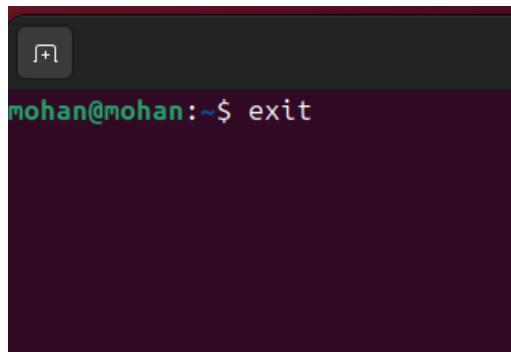
```
mohan@mohan:~$ history
1 history
2 echo "Mohan Kumar Pulibanti"
3 echo "Ram Kumar Pulibanti"
4 cal -y
5 cal 2028
6 history
7 echo "Mohan Kumar Pulibanti"
8 histort
9 history
mohan@mohan:~$ !2
echo "Mohan Kumar Pulibanti" } Most Recent Command
Mohan Kumar Pulibanti
mohan@mohan:~$ !!
echo "Mohan Kumar Pulibanti"
Mohan Kumar Pulibanti
mohan@mohan:~$
```

- **history -c; history -w** → clears the history and writes the history.



```
mohan@mohan:~$ history -c; history -w
mohan@mohan:~$ history
 1 history
mohan@mohan:~$
```

- **exit** – command to close the terminal.

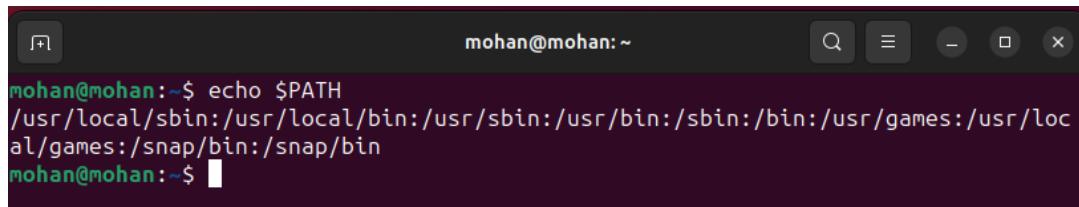


```
mohan@mohan:~$ exit
```

Understanding Command Structure

- All commands are computer programs that are installed somewhere on our computer.
Eg: date is a program, cal is a program and echo is another program.
- Command Structure
 - commandName -options inputs**
 - commndName -option1 option1input -option2 option2input input1 input2**
 - options – is to customize the behavior of the command.**
 - inputs – on which the command operates on**
- Based on the command name, Shell knows the name of the program that the user wants to run. Then, Shell searches for the program based on the name in the Shell path.
- Shell Path is the List of the folders that can contain these programs.
- To see the shell's path, below is the command.

echo \$PATH



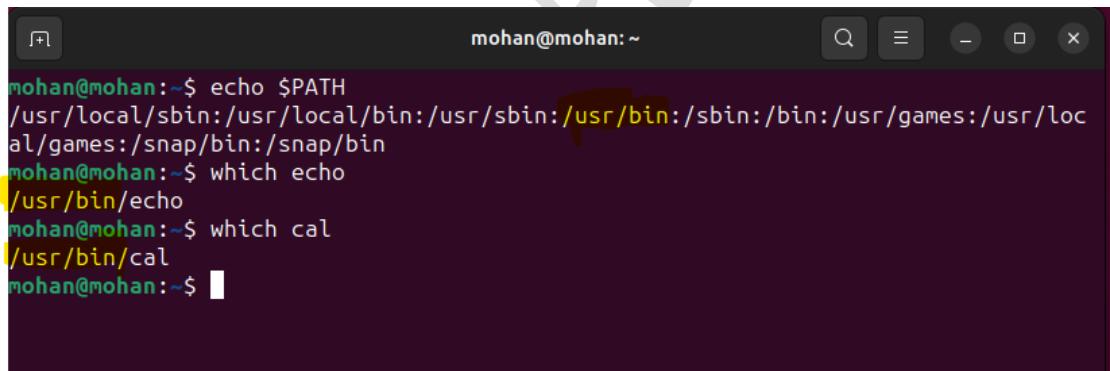
```
mohan@mohan:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/snap/bin
mohan@mohan:~$
```

- Shell will start searching for the program based on the `commandName` in the leftmost folder (`/usr/local/sbin`), if not found it goes to the next folder. If not found in any of the folders, it throws an error saying command not found.



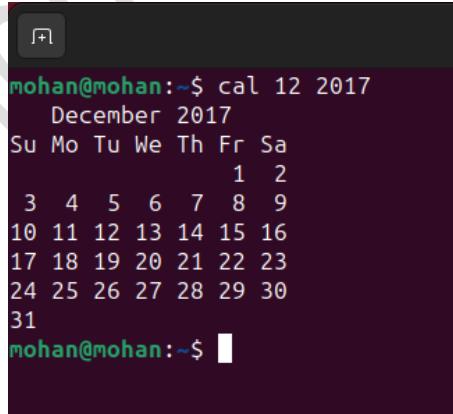
mohan@mohan:~\$ mohan kumar
mohan: command not found
mohan@mohan:~\$

- **which** – tells us in which folder the program is present for the `commandName` given.



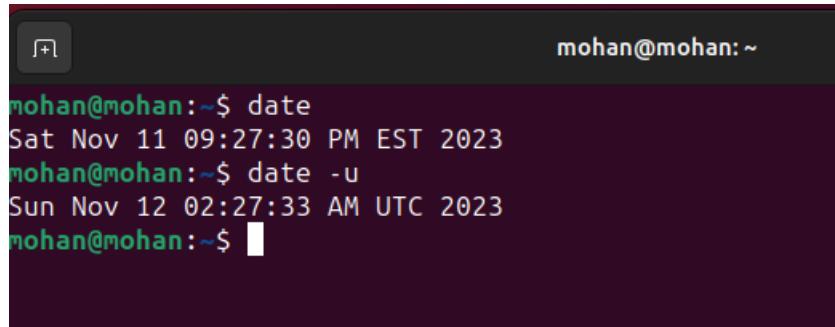
mohan@mohan:~\$ echo \$PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/snap/bin
mohan@mohan:~\$ which echo
/usr/bin/echo
mohan@mohan:~\$ which cal
/usr/bin/cal
mohan@mohan:~\$

- Not all commands require input, but some commands require inputs (The fancy name for input is operand). We can give multiple inputs to a command by space separation.
Eg: **cal 12 2017** (12 is a month, 2017 is a year → 2 inputs)



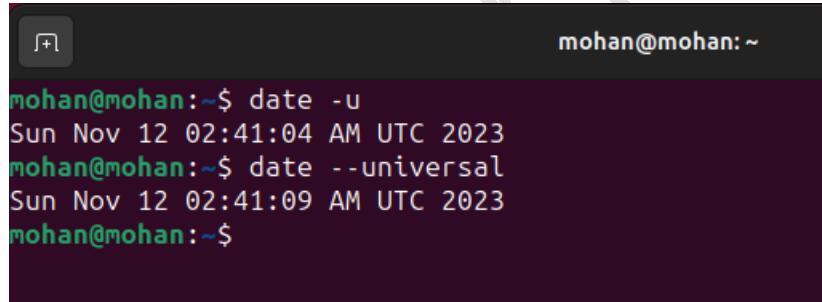
mohan@mohan:~\$ cal 12 2017
December 2017
Su Mo Tu We Th Fr Sa
1 2
3 4 5 6 7 8 9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31
mohan@mohan:~\$

- Options given to command (cal -y) → all options will be preceded by ‘-’. y is the option here and it is to say cal to print the calendar of the current year.
Other examples: date -u (to print date in UTC format whereas just date will print date in EST format)



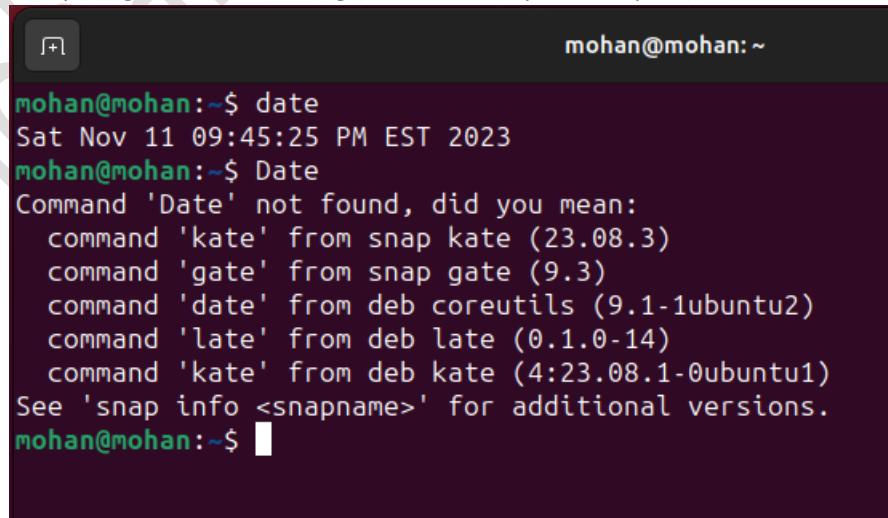
```
mohan@mohan:~$ date
Sat Nov 11 09:27:30 PM EST 2023
mohan@mohan:~$ date -u
Sun Nov 12 02:27:33 AM UTC 2023
mohan@mohan:~$
```

- Multiple options can be given to a command in two ways, below are the formats
 - **date -u -a -b -c**
 - **date -uabc**
- Shuffling the options to a command doesn't matter "date -uabc", "date -aabc", "date -abcu" will have the same effect.
- Options will also have long-form names. Long-form names will be preceded by "--", whereas short-form will be preceded by "-".

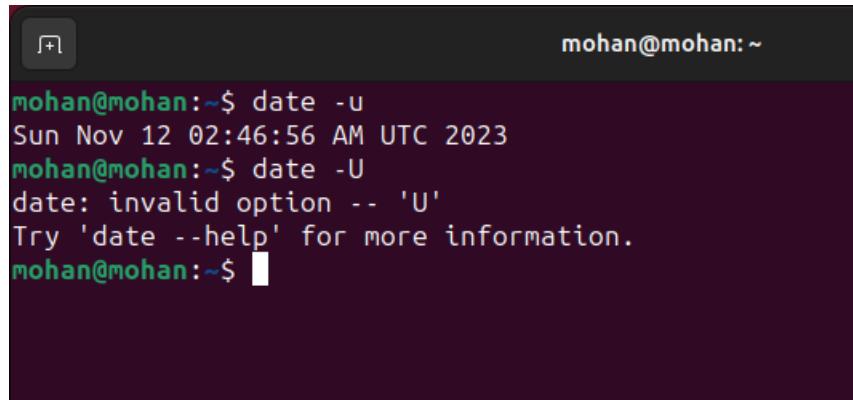


```
mohan@mohan:~$ date -u
Sun Nov 12 02:41:04 AM UTC 2023
mohan@mohan:~$ date --universal
Sun Nov 12 02:41:09 AM UTC 2023
mohan@mohan:~$
```

- Format to chain multiple long form options
date --universal --longform2 --longform3
- Everything in Linux including (commands, options, inputs) is case sensitive.



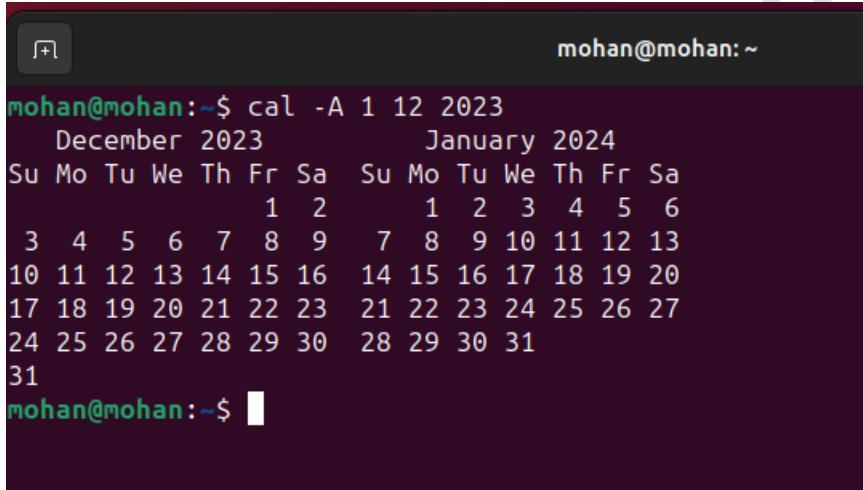
```
mohan@mohan:~$ date
Sat Nov 11 09:45:25 PM EST 2023
mohan@mohan:~$ Date
Command 'Date' not found, did you mean:
  command 'kate' from snap kate (23.08.3)
  command 'gate' from snap gate (9.3)
  command 'date' from deb coreutils (9.1-1ubuntu2)
  command 'late' from deb late (0.1.0-14)
  command 'kate' from deb kate (4:23.08.1-0ubuntu1)
See 'snap info <snapname>' for additional versions.
mohan@mohan:~$
```



```
mohan@mohan:~$ date -u
Sun Nov 12 02:46:56 AM UTC 2023
mohan@mohan:~$ date -U
date: invalid option -- 'U'
Try 'date --help' for more information.
mohan@mohan:~$
```

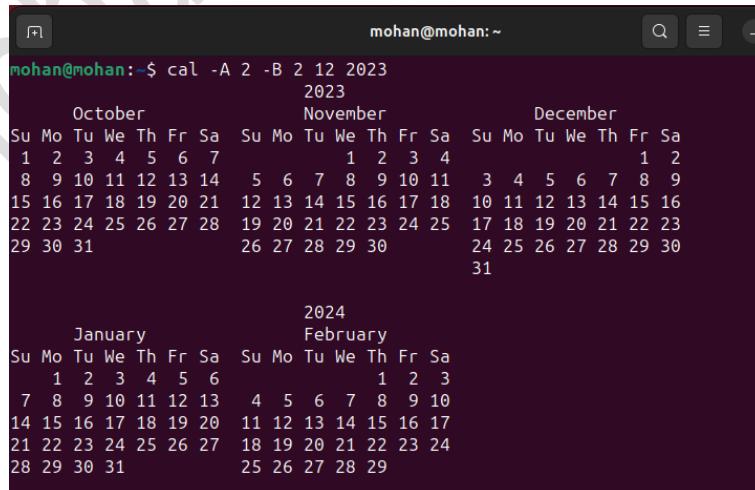
- Options can also have inputs.

- **cal -A 1 12 2017** (here 1 is input for option A) → this command says to print the calendar for 12,2017 and 1 month after. Here A stands for after.



```
mohan@mohan:~$ cal -A 1 12 2023
December 2023           January 2024
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
          1  2           1  2  3  4  5  6
 3  4  5  6  7  8  9   7  8  9 10 11 12 13
10 11 12 13 14 15 16  14 15 16 17 18 19 20
17 18 19 20 21 22 23  21 22 23 24 25 26 27
24 25 26 27 28 29 30  28 29 30 31
31
mohan@mohan:~$
```

- **cal -A 2 -B 2 12 2023 / cal -A2 -B2 12 2023**(A stands for after, B stands for before) – this command says print calendar for 12,2023 and 2 months before and after to 12,2023.



```
mohan@mohan:~$ cal -A 2 -B 2 12 2023
2023
October           November           December
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
  1  2  3  4  5  6  7   1  2  3  4           1  2
  8  9 10 11 12 13 14   5  6  7  8  9 10 11   3  4  5  6  7  8  9
15 16 17 18 19 20 21   12 13 14 15 16 17 18   10 11 12 13 14 15 16
22 23 24 25 26 27 28   19 20 21 22 23 24 25   17 18 19 20 21 22 23
29 30 31               26 27 28 29 30           24 25 26 27 28 29 30
                           31

                January           February
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
  1  2  3  4  5  6           1  2  3
  7  8  9 10 11 12 13   4  5  6  7  8  9 10
14 15 16 17 18 19 20   11 12 13 14 15 16 17
21 22 23 24 25 26 27   18 19 20 21 22 23 24
28 29 30 31             25 26 27 28 29
```

Using the Linux manual

Manual Structure Cheat Sheet

Section	Contains	Description
1	User Commands	Commands that can be run from the shell by a normal user (typically no administrative privileges are needed)
2	System Calls	Programming functions used to make calls to the Linux kernel
3	C Library Functions	Programming functions that provide interfaces to specific programming libraries.
4	Devices and Special Files	File system nodes that represent hardware devices or software devices.
5	File Formats and Conventions	The structure and format of file types or specific configuration files.
6	Games	Games available on the system
7	Miscellaneous	Overviews of miscellaneous topics such as protocols, filesystems and so on.
8	System administration tools and Daemons	Commands that require root or other administrative privileges to use

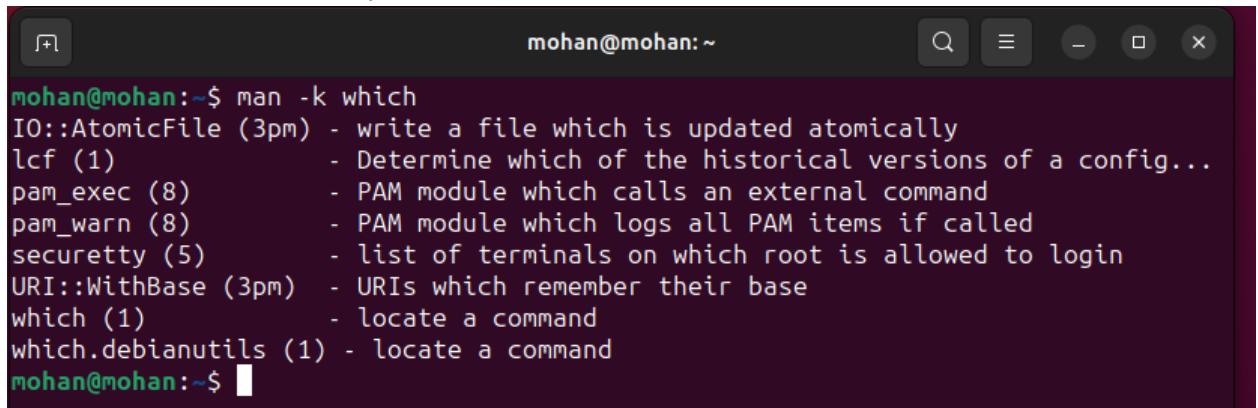
Note: Sections 1, 5, 8 are what you will likely use most often.

4. Some Important Definitions

Command	An instruction typed in the terminal and submitted to the shell for interpretation.
Shell	A program that interprets commands for meaning.
Terminal	A graphical window where commands can be typed and submitted to the shell.

- Command to search manual with a search term.

man -k <searchterm> → **man -k which** (search all the man pages and return results that match the seachterm which)

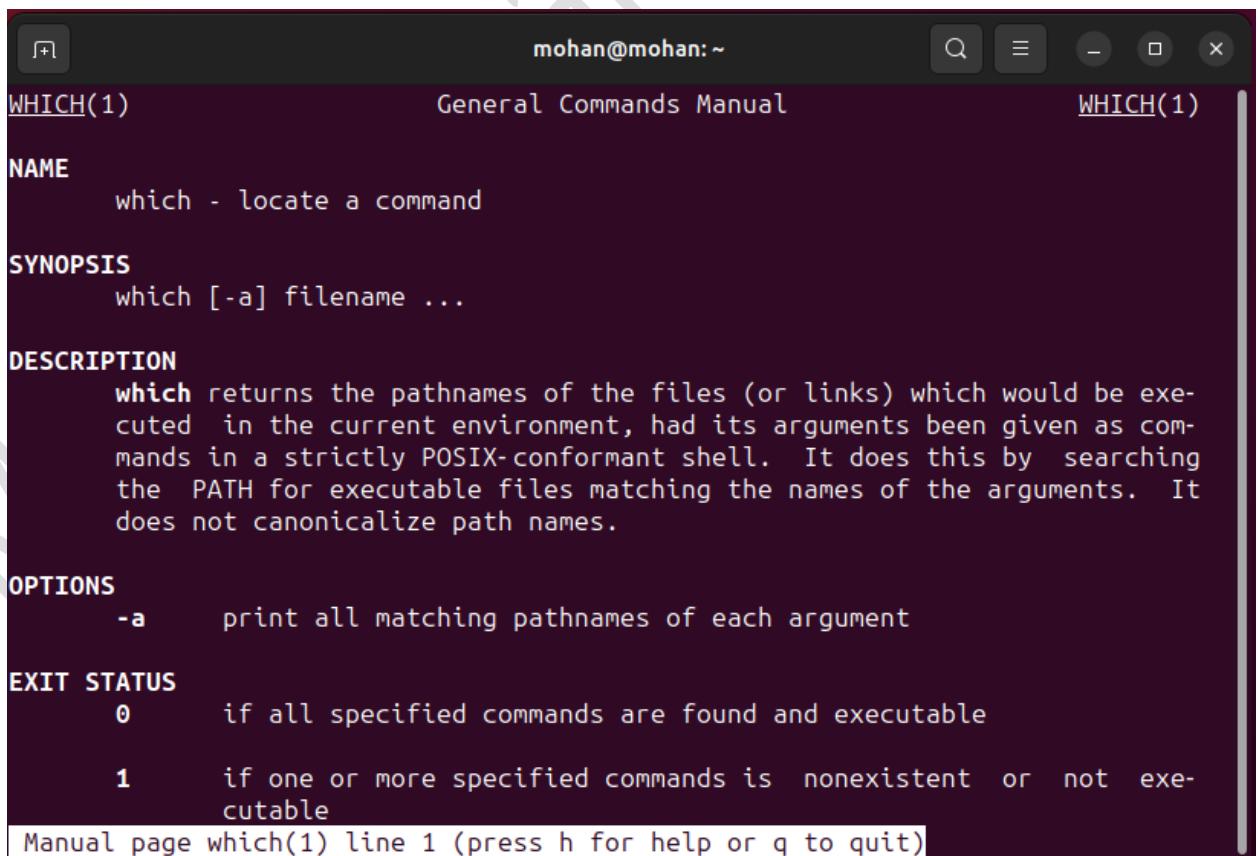


```
mohan@mohan:~$ man -k which
IO::AtomicFile (3pm) - write a file which is updated atomically
lcf (1)           - Determine which of the historical versions of a config...
pam_exec (8)      - PAM module which calls an external command
pam_warn (8)      - PAM module which logs all PAM items if called
securetty (5)     - list of terminals on which root is allowed to login
URI::WithBase (3pm) - URIs which remember their base
which (1)          - locate a command
which.debianutils (1) - locate a command
mohan@mohan:~$
```

The number in the brackets (**which (1) – locate a command**) specifies the section in which it found this result.

- To further get details of the search term in a specific section, the command is

man <section_number> <search_term> → **man 1 which** → **search which term in section 1**



```
WHICH(1)                         General Commands Manual                         WHICH(1)

NAME
    which - locate a command

SYNOPSIS
    which [-a] filename ...

DESCRIPTION
    which returns the pathnames of the files (or links) which would be executed in the current environment, had its arguments been given as commands in a strictly POSIX-conformant shell. It does this by searching the PATH for executable files matching the names of the arguments. It does not canonicalize path names.

OPTIONS
    -a      print all matching pathnames of each argument

EXIT STATUS
    0      if all specified commands are found and executable
    1      if one or more specified commands is nonexistent or not executable
Manual page which(1) line 1 (press h for help or q to quit)
```

- When searching 1st section of the manual page, we don't need to specifically give the section number i.e., “**man 1 which**” is the same as “**man which**”, in other words, if we don't specify the section_number it searches 1st section.
- On the manual page, if the command format seems like **commandName [THING] THING ...**, below is the meaning.

Command Synopsis Cheat Sheet

Section	Meaning
[THING]	THING is optional.
<THING>	THING is mandatory (required)
THING ...	THING can be repeated (limitlessly)
THING1 THING2	Use THING1 OR THING2. Not Both.
<i>THING</i>	[Notice the Italics] Replace <i>THING</i> with whatever is appropriate.

Useful Extra Resource:

<https://medium.com/@jaewei.j.w/how-to-read-man-page-synopsis-3408e7fd0e42>

- man -k “list directory contents” → searches for the term “list of directory contents”
man -k list directory contents → searches for terms “list”, “directory”, “contents”

```
mohan@mohan:~$ man -k "list directory contents"
dir (1)          - list directory contents
ls (1)          - list directory contents
ntfsfs (8)       - list directory contents on an NTFS filesystem
vdir (1)         - list directory contents
mohan@mohan:~$ man -k list directory contents
acl (5)          - Access Control Lists
add-apt-repository (1) - Adds a repository into the /etc/apt/sources.list or ...
add-shell (8)      - add shells to the list of valid login shells
american-english (5) - a list of English words
appres (1)        - list X application resource database
apt-add-repository (1) - Adds a repository into the /etc/apt/sources.list or ...
axfer-list (1)     - dump lists of available sound devices and nodes to tra...
basename (1)       - strip directory and suffix from filenames
bindtextdomain (3) - set directory containing message catalogs
bpf-helpers (7)    - list of eBPF helper functions
british-english (5) - a list of English words
chacl (1)          - change the access control list of a file or directory
check-language-support (1) - returns the list of missing packages in order to...
chroot (8)         - run command or interactive shell with special root dir...
column (1)         - columnate lists
corelist (1)       - a commandline frontend to Module::CoreList
```

- Not all Commands will have man pages, some commands with are present in the shell (called shell commands) can be found with the help for help command.

```
mohan@mohan:~$ man cd
No manual entry for cd
mohan@mohan:~$
```

```
mohan@mohan:~$ help cd
cd: cd [-L|[-P [-e]] [-@]] [dir]
      Change the shell working directory.

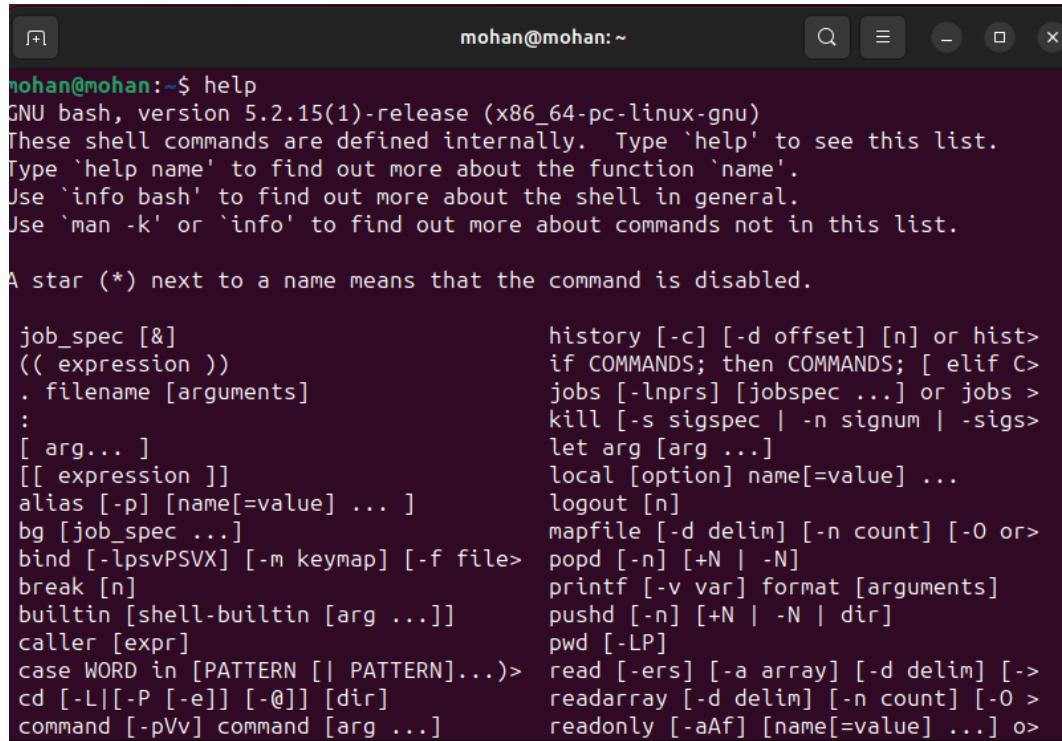
      Change the current directory to DIR.  The default DIR is the value of the
      HOME shell variable. If DIR is "-", it is converted to $OLDPWD.

      The variable CDPATH defines the search path for the directory containing
      DIR. Alternative directory names in CDPATH are separated by a colon (:).
      A null directory name is the same as the current directory. If DIR begins
      with a slash (/), then CDPATH is not used.

      If the directory is not found, and the shell option `cdable_vars' is set,
      the word is assumed to be a variable name. If that variable has a value,
      its value is used for DIR.

Options:
  -L      force symbolic links to be followed: resolve symbolic
          links in DIR after processing instances of `..'
  -P      use the physical directory structure without following
          symbolic links: resolve symbolic links in DIR before
          processing instances of `..'
  -e      if the -P option is supplied, and the current working
          directory is a symbolic link, follow it
```

- help command will return all the shell commands, to explore these commands we need to use **help <commandname>**



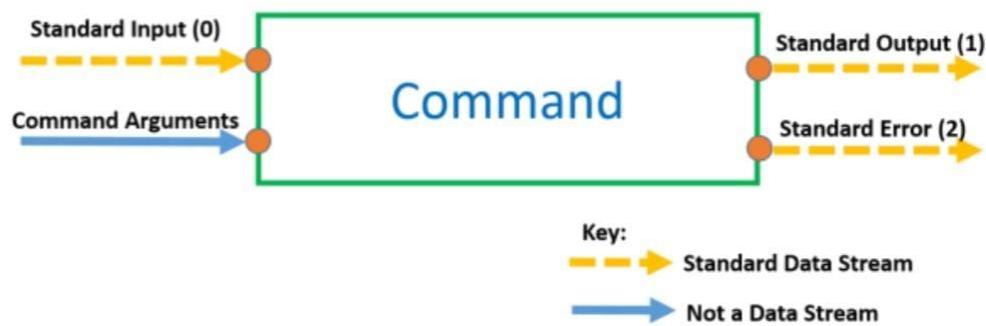
```
mohan@mohan:~$ help
GNU bash, version 5.2.15(1)-release (x86_64-pc-linux-gnu)
These shell commands are defined internally. Type `help' to see this list.
Type `help name' to find out more about the function `name'.
Use `info bash' to find out more about the shell in general.
Use `man -k' or `info' to find out more about commands not in this list.

A star (*) next to a name means that the command is disabled.

job_spec [&]
(( expression ))
. filename [arguments]
:
[ arg... ]
[[ expression ]]
alias [-p] [name[=value] ... ]
bg [job_spec ...]
bind [-lpsvPSVX] [-m keymap] [-f file]
break [n]
builtin [shell-builtin [arg ...]]
caller [expr]
case WORD in [PATTERN [| PATTERN]...)...
cd [-L|[-P [-e]] [-@]] [dir]
command [-pVv] command [arg ...]
history [-c] [-d offset] [n] or hist>
if COMMANDS; then COMMANDS; [ elif C>
jobs [-lnprs] [jobspec ...] or jobs >
kill [-s sigspec | -n signum | -sigs>
let arg [arg ...]
local [option] name[=value] ...
logout [n]
mapfile [-d delim] [-n count] [-o or>
popd [-n] [+N | -N]
printf [-v var] format [arguments]
pushd [-n] [+N | -N | dir]
pwd [-LP]
read [-ers] [-a array] [-d delim] [->
readarray [-d delim] [-n count] [-o >
readonly [-aAf] [name[=value] ...] o>
```

Command Input and Output

7. Command Input and Output



Standard Data Streams can be **redirected** and are identified using their stream number.

Redirection of the standard output of one command to the standard input of another command is known as **piping**.

- Standard output and standard error will by default be connected to the terminal screen.
But we can redirect these to wherever we wish.
- Standard input by default is connected to the keyboard.
Example: Input is given by typing using a computer keyboard

```
mohan@mohan:~$ cat
MOhan Kumar Pulibanti
MOhan Kumar Pulibanti
```

- `cal 12 2017 → 12 2017` are called command line inputs.
- Data Streams can flow and piped together, whereas command line arguments are more Static and still.
- Not all commands accept Standard inputs (eg: echo command), but almost all the commands accept command line arguments.

Redirection

- `< → Input`
- `> → Output, 1> Standard output stream, 2> Standard Error stream`
- `cat 0< input.txt 1> output.txt 2> error.txt` → this gets the input from `input.txt`, outputs that to `output.txt`, and if any errors stream that to `error.txt`
- `>` → Truncates and writes to the new file, `>>` → appends to the existing file.
- `1>` is same as `>`, `0<` is same as `<`

Piping

- Piping is all about taking output from one command and sending that as input to another command.
- Piping the output of date tp cut command:
`date | cut --delimiter " " --fields 1`

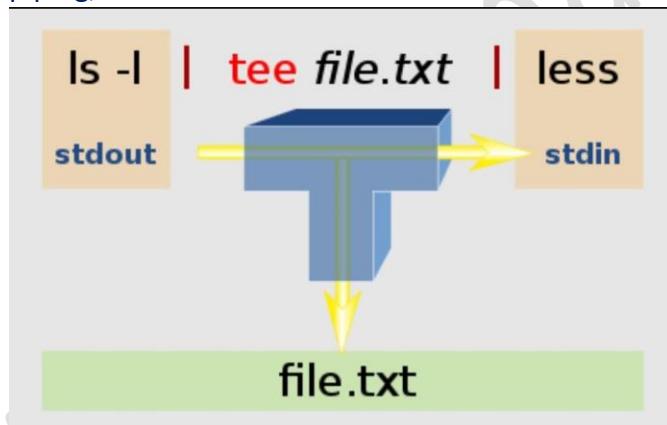
```
mohan@mohan:~/Desktop$ date
Sun Nov 12 01:38:55 AM EST 2023
mohan@mohan:~/Desktop$ date | cut --delimiter " " --fields 1
Sun
```

➤ `date | cut > day.txt --delimiter " " --fields1`
`date | cut --delimiter " " -fields 1 > day.txt`
`date | cut --delimiter " " >day.txt --fields 1`

Any of the above 3 combinations would yield the same results, we can shuffle the places of different parameters.

```
mohan@mohan:~/Desktop$ date | cut >day.txt --delimit " " --fields 1
mohan@mohan:~/Desktop$ cat day.txt
Sun
mohan@mohan:~/Desktop$ date | cut --delimit " " >day.txt --fields 1
mohan@mohan:~/Desktop$ cat day.txt
Sun
mohan@mohan:~/Desktop$ date | cut --delimit " " --fields 1 >day.txt
mohan@mohan:~/Desktop$ cat day.txt
Sun
mohan@mohan:~/Desktop$
```

➤ `date | cut > day.txt --delimiter " " -fields 1` → If we want to save the date to the file and also pass the output of the date to the cut command, it's not possible with this kind of piping, we need to use tee to achieve this.



```
mohan@mohan:~/Desktop$ date | tee date.txt |cut --delimiter " " --fields 1 >day.txt
mohan@mohan:~/Desktop$ cat date.txt
Sun Nov 12 02:07:40 AM EST 2023
mohan@mohan:~/Desktop$ cat day.txt
Sun
mohan@mohan:~/Desktop$
```

- **xargs** – helps to convert standard output to command line arguments as not all the commands accept standard inputs. For instance, the echo command doesn't accept standard input, it only allows command line arguments.

```
mohan@mohan:~/Desktop$ echo "hello"
hello
mohan@mohan:~/Desktop$ date | echo
date output is not considered by echo
mohan@mohan:~/Desktop$ date | echo "hello" whereas echo just printed its argument - hello
hello
mohan@mohan:~/Desktop$ date | xargs echo with xargs output of date is converted as
Sun Nov 12 02:14:03 AM EST 2023 command line arg and sent to echo
mohan@mohan:~/Desktop$ date | xargs echo "Mohan" when echo has its args and command line
Mohan Sun Nov 12 02:14:16 AM EST 2023 arguments via piping, it prints its args first and
mohan@mohan:~/Desktop$ [ ] later it prints the command line argument.

So, Mohan and then date got printed to terminal
```

- piping further with xargs

```
mohan@mohan:~/Desktop$ date | xargs echo "Mohan" | cat >xrguy.txt
mohan@mohan:~/Desktop$ cat xrguy.txt
Mohan Sun Nov 12 02:21:25 AM EST 2023
mohan@mohan:~/Desktop$ [ ]
```

- **rm** command is used to delete a file and it accepts only command line arguments. So, when piping the output of some command to rm , we need to use xargs.

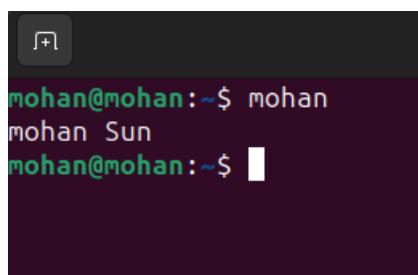
Aliases (nickname for a command or a pipeline)

- Step1: create .bash_aliases file and save it in the home directory
- Step2: edit the .bash_aliases file and add aliases to a particular command. For Example:
alias mohan='date | tee savedate.txt | cut --delimiter " " --fields 1 | xargs echo'

```
Open .bash_aliases
alias mohan='date | tee savedate.txt | cut --delimiter " " --fields
1 |xargs echo mohan'
```

- Save and close the editor of .bash_aliases file.

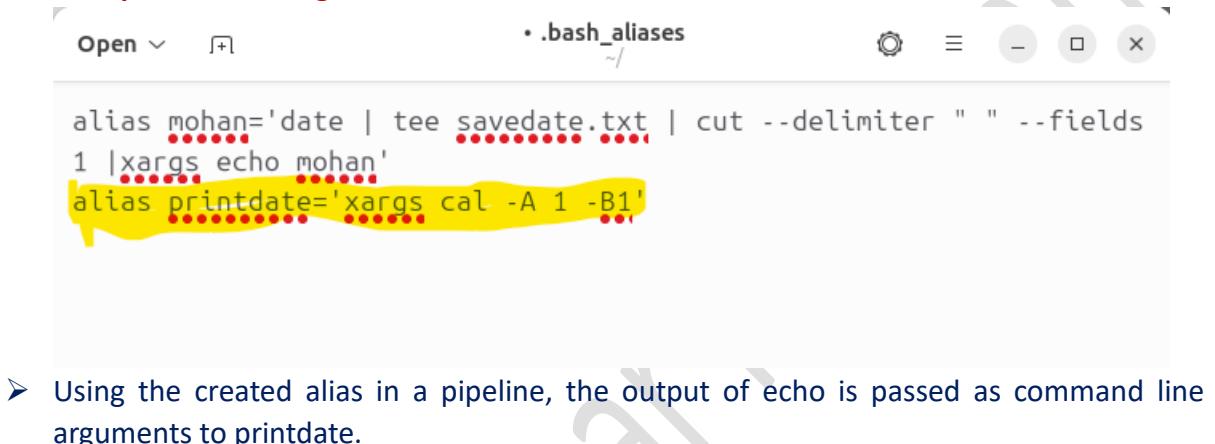
- Execute mohan from terminal.



```
mohan@mohan:~$ mohan
mohan Sun
mohan@mohan:~$
```

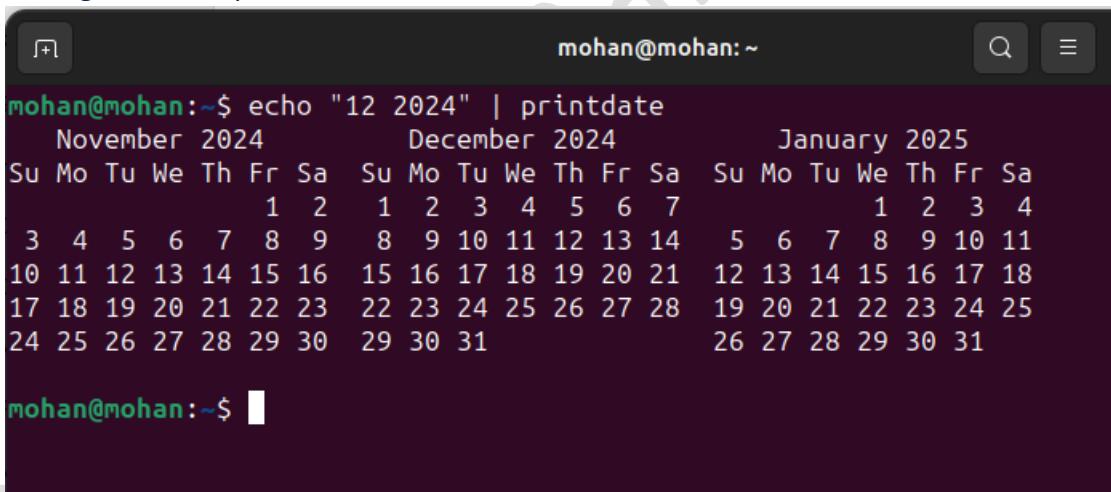
- Using Aliases in a pipeline. Create an alias as below

```
alias printdate='xargs cal -A 1 -B1'
```



```
Open ⌂ .bash_aliases ⌂
alias mohan='date | tee savedate.txt | cut --delimiter " " --fields
1 |xargs echo mohan'
alias printdate='xargs cal -A 1 -B1'
```

- Using the created alias in a pipeline, the output of echo is passed as command line arguments to printdate.



```
mohan@mohan:~$ echo "12 2024" | printdate
November 2024          December 2024          January 2025
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
                   1  2   1  2   3  4   5  6   7   1  2   3  4
                   3  4   5  6   7  8   9  10  11  12  13  14   5  6   7  8   9  10  11
                   10 11  12 13  14 15  16   15 16  17 18  19 20  21   12 13  14 15  16 17  18
                   17 18  19 20  21 22  23   22 23  24 25  26 27  28   19 20  21 22  23 24  25
                   24 25  26 27  28 29  30   29 30  31   26 27  28 29  30 31
```

- Aliases are accessible when you restart your terminal.

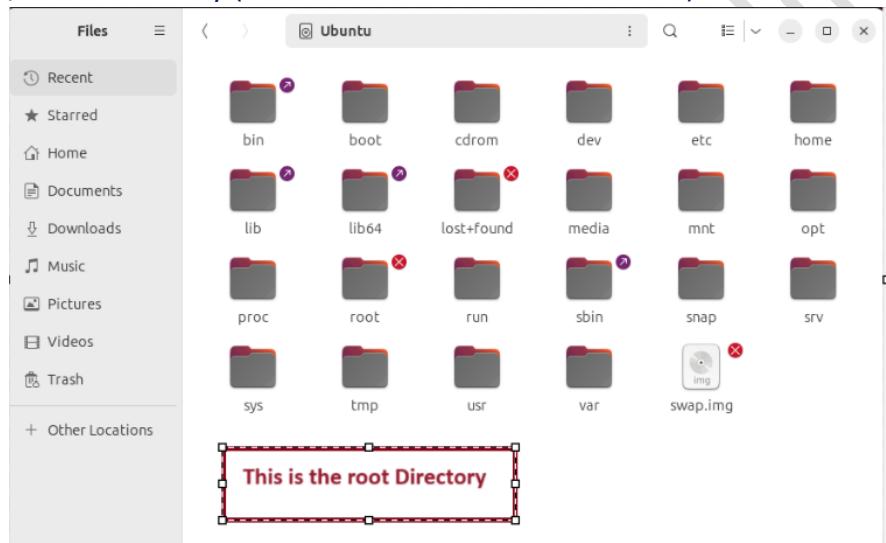
Navigating File System

mohan@mohan:~\$

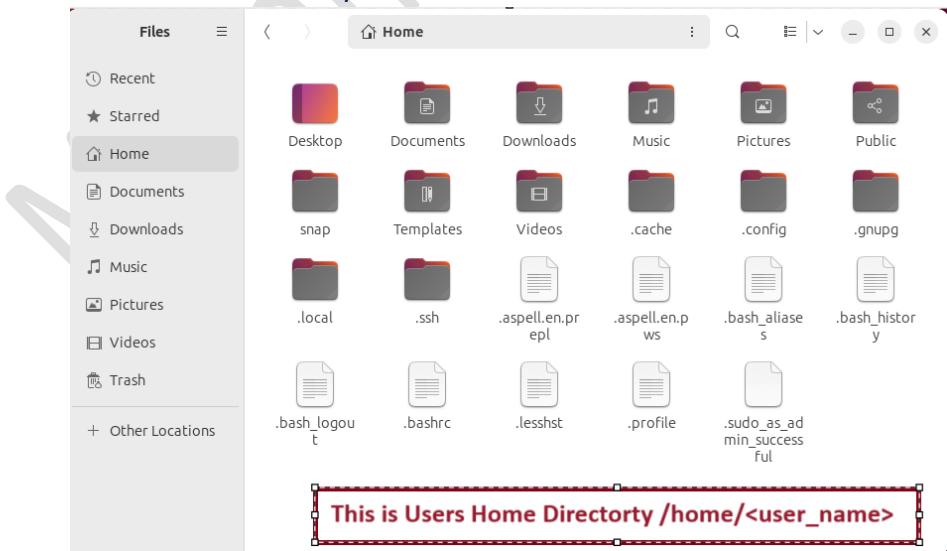
Annotations:

- username: points to the text "mohan@mohan:"
- virtualMachineName: points to the text "mohan"
- current user home directory: points to the tilde (~) symbol.

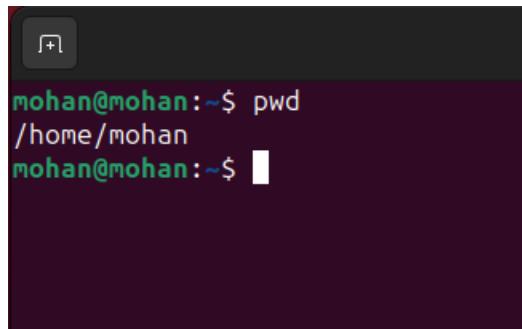
- / - root directory (Click on other locations → ubuntu)



- Users Home Directory

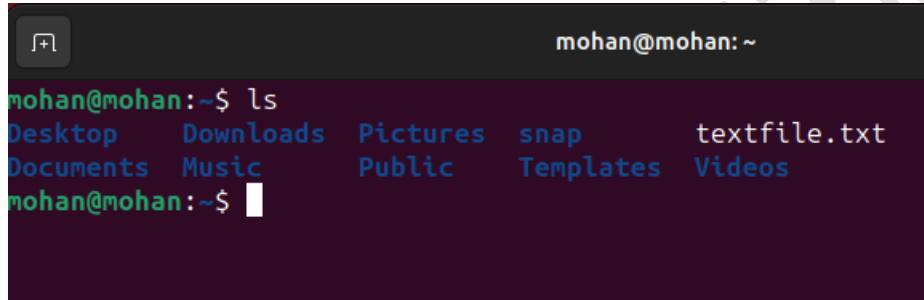


- **pwd** → print working directory. (tells us the path in which shell is currently operating in)



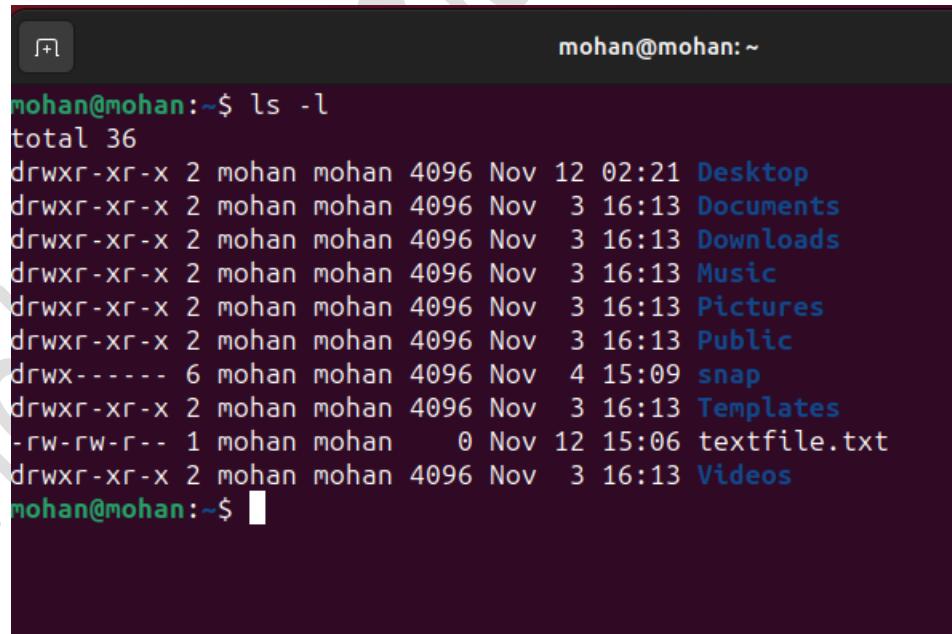
```
mohan@mohan:~$ pwd
/home/mohan
mohan@mohan:~$
```

- **absolute path** → if the path starts all the way from the home directory.
Eg: /home/mohan
- **ls <path of directory>** → list the files in a directory, output in blue color indicates folders and others as files, green color indicates programs



```
mohan@mohan:~$ ls
Desktop  Downloads  Pictures  snap      textfile.txt
Documents  Music      Public    Templates  Videos
mohan@mohan:~$
```

- **ls -l** → long list the files in a directory



```
mohan@mohan:~$ ls -l
total 36
drwxr-xr-x 2 mohan mohan 4096 Nov 12 02:21 Desktop
drwxr-xr-x 2 mohan mohan 4096 Nov  3 16:13 Documents
drwxr-xr-x 2 mohan mohan 4096 Nov  3 16:13 Downloads
drwxr-xr-x 2 mohan mohan 4096 Nov  3 16:13 Music
drwxr-xr-x 2 mohan mohan 4096 Nov  3 16:13 Pictures
drwxr-xr-x 2 mohan mohan 4096 Nov  3 16:13 Public
drwx----- 6 mohan mohan 4096 Nov  4 15:09 snap
drwxr-xr-x 2 mohan mohan 4096 Nov  3 16:13 Templates
-rw-rw-r-- 1 mohan mohan     0 Nov 12 15:06 textfile.txt
drwxr-xr-x 2 mohan mohan 4096 Nov  3 16:13 Videos
mohan@mohan:~$
```

- **ls -a** → long list the files in the directory including hidden files. Names starting with . are hidden files. (.. are two important hidden folders that help to navigate between directories, ‘cd .’ with point to the same directory, cd .. takes to the parent directory)

```
mohan@mohan:~$ ls -a
.
..
.aspell.en.prepl .bashrc .lesshst .ssh
.aspell.en.pws .cache .local .sudo_as_admin_successful
.bash_aliases Desktop Pictures Templates
.bash_history Documents .profile textfile.txt
.bash_logout Downloads Public Videos
.gnupg snap

mohan@mohan:~$ ls
Desktop Downloads Pictures snap      textfile.txt
Documents Music     Public   Templates Videos
mohan@mohan:~$
```

- **ls -lh** → long list files in the directory in human readable format

```
mohan@mohan:~$ ls -lh
total 36K
drwxr-xr-x 2 mohan mohan 4.0K Nov 12 02:21 Desktop
drwxr-xr-x 2 mohan mohan 4.0K Nov  3 16:13 Documents
drwxr-xr-x 2 mohan mohan 4.0K Nov  3 16:13 Downloads
drwxr-xr-x 2 mohan mohan 4.0K Nov  3 16:13 Music
drwxr-xr-x 2 mohan mohan 4.0K Nov  3 16:13 Pictures
drwxr-xr-x 2 mohan mohan 4.0K Nov  3 16:13 Public
drwx----- 6 mohan mohan 4.0K Nov  4 15:09 snap
drwxr-xr-x 2 mohan mohan 4.0K Nov  3 16:13 Templates
-rw-rw-r-- 1 mohan mohan    0 Nov 12 15:06 myfile.txt
drwxr-xr-x 2 mohan mohan 4.0K Nov  3 16:13 Videos
mohan@mohan:~$
```

- It accepts everything as command line arguments.

- **ls -F** → classify the files, directories will be ended with /, whereas files will not have / at end

```
mohan@mohan:~$ ls -F
Desktop/ Downloads/ Pictures/ snap/      textfile.txt
Documents/ Music/ Public/ Templates/ Videos/
mohan@mohan:~$
```

- **cd <directory path>** → change directory.

```
mohan@mohan:~/Downloads
mohan@mohan:~$ cd /home/mohan/Downloads
mohan@mohan:~/Downloads$ pwd
/home/mohan/Downloads
mohan@mohan:~/Downloads$ cd ..
mohan@mohan:~$ pwd
/home/mohan
mohan@mohan:~$ cd Downloads
mohan@mohan:~/Downloads$ pwd
/home/mohan/Downloads
mohan@mohan:~/Downloads$
```

Absolute Path
As its started from root directory '/'

Relative Path
as we just gave Downloads

- cd without any directory path will take you to the user home directory wherever you are.

```
mohan@mohan:~/Downloads$ pwd
/home/mohan/Downloads
mohan@mohan:~/Downloads$ cd
mohan@mohan:~$ pwd
/home/mohan
mohan@mohan:~$
```

- using cd we can't enter all the directories as some directories will have restricted permission. (e.g.: root folder will have restricted permissions)

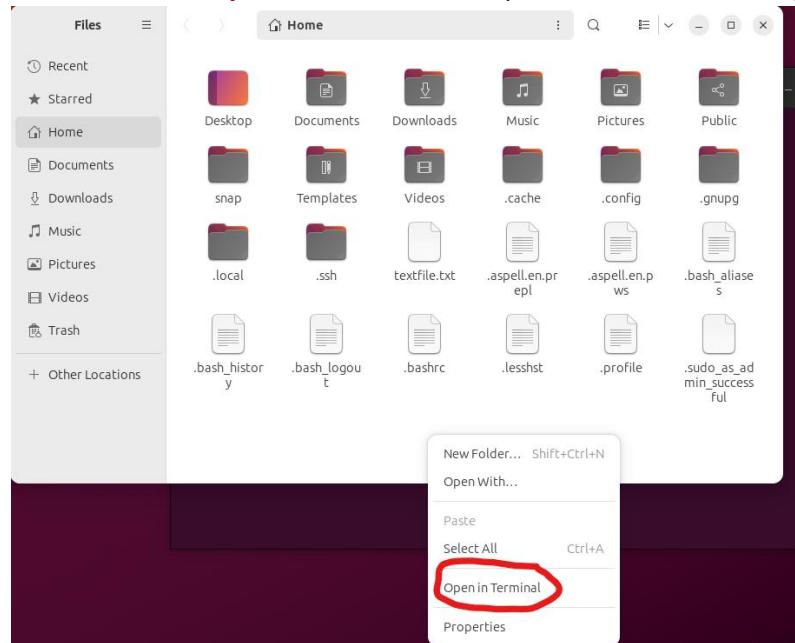
```
mohan@mohan:/$ pwd
/
mohan@mohan:/$ ls
bin  cdrom  etc  lib  lost+found  mnt  proc  run  snap  swap.img  tmp  var
boot  dev  home  lib64  media  opt  root  sbin  srv  sys  usr
mohan@mohan:/$ cd root
bash: cd: root: Permission denied
mohan@mohan:/$
```

- **Tab Completion** → Auto suggests the paths when we press the tab by entering a few letters, when no suggestions are given even when we press tab, double click tab to show the conflicts.



Tab_Completion.mp4

- **Right-click and select “open in terminal”** to open a terminal where you are graphically.

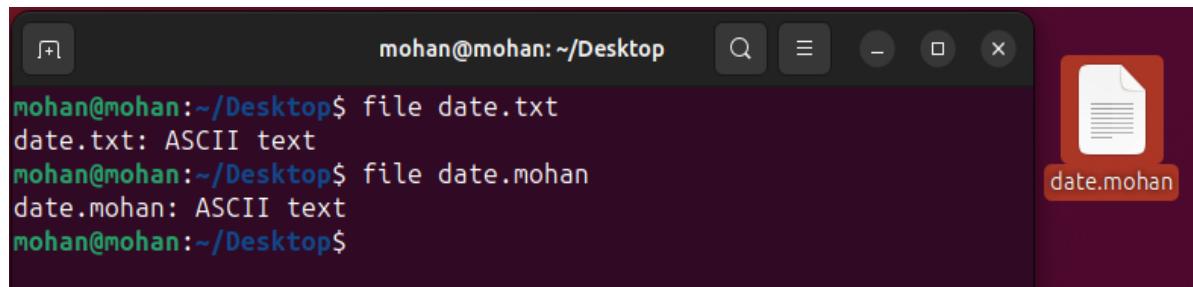


File Extensions in Linux

- In Linux, File extensions don't matter. It not only looks at the file extensions but also looks at the content of the file to determine what file it is. But sometimes, file extensions do matter, let's dive deep into that.
- **file** → this command tells what type of file we are generally dealing with.

```
mohan@mohan: ~/Desktop
mohan@mohan:~/Desktop$ file date.txt
date.txt: ASCII text
mohan@mohan:~/Desktop$
```

- if we rename date.txt to date.mohan and execute the file command again on date.mohan, it still tells us that it's an ASCII text file

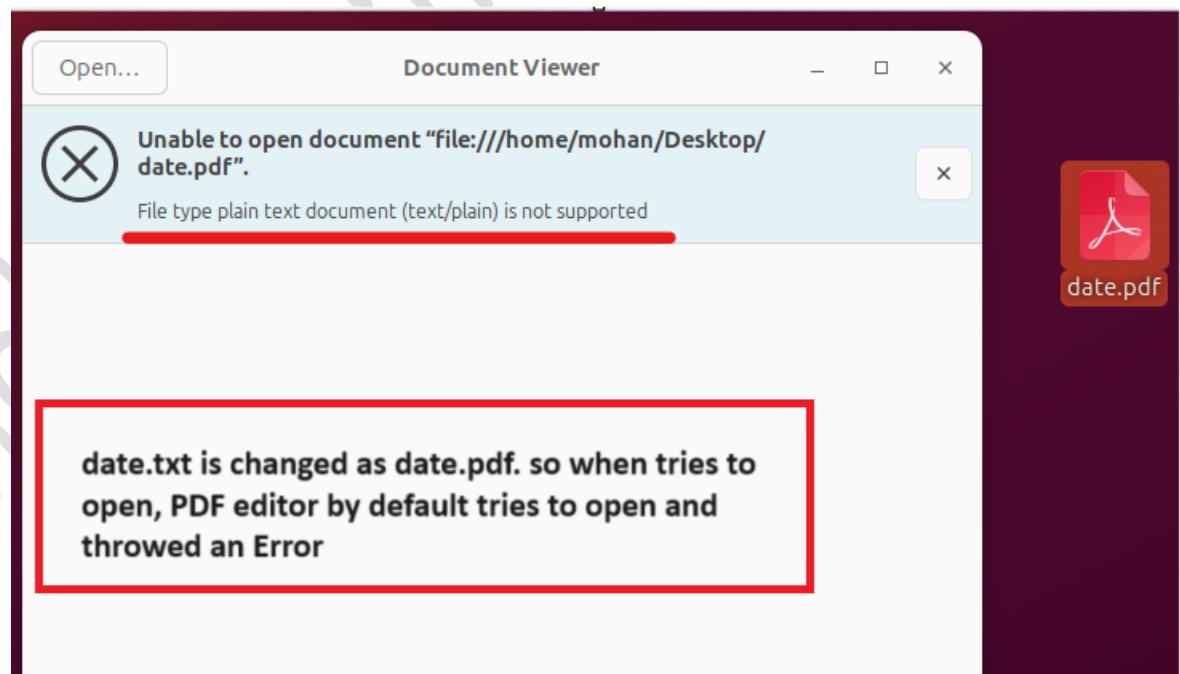


mohan@mohan:~/Desktop\$ file date.txt
date.txt: ASCII text
mohan@mohan:~/Desktop\$ file date.mohan
date.mohan: ASCII text
mohan@mohan:~/Desktop\$

A screenshot of a terminal window titled "mohan@mohan: ~/Desktop". The window shows the command "file date.txt" followed by its output "date.txt: ASCII text". Then, the command "file date.mohan" is run, and its output "date.mohan: ASCII text" is shown. A file icon for "date.mohan" is visible in the top right corner of the terminal window.

- In Linux file extensions don't matter, it determine the file type based on the piece of code present inside the top of every file called header, it serves as a kind of label for Linux to read.
- The programs/Sofwares that are installed on top of the operating systems such as PDF viewers etc, they might need the file extension in order to open them, while the operating system doesn't care.

For example, if data.txt file is changed to data.pdf then by default PDF editor software installed tries to open that file as the extension is .pdf but throws an error.

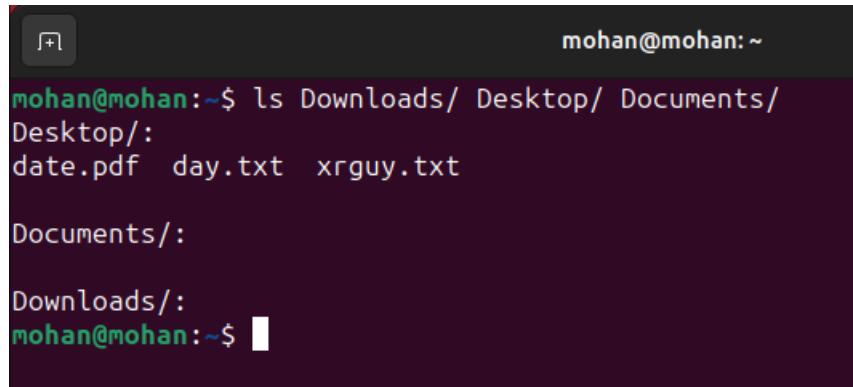


- You can name files whatever you want in Linux (even .mohan) but try not to confuse third party softwares like pdf editors etc.,

Wildcards

- Scenario: Is all the file in Desktop, Downloads and Documents.

Without Wildcards:

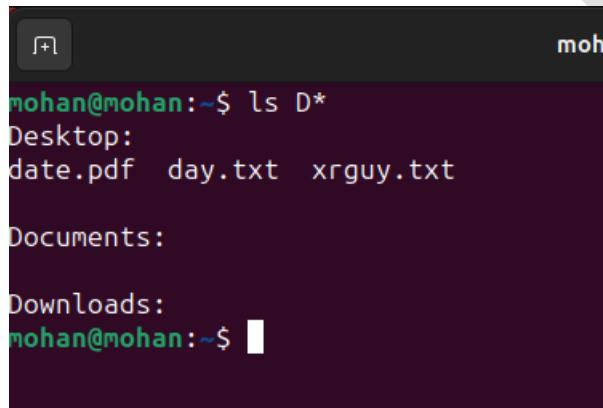


```
mohan@mohan:~$ ls Downloads/ Desktop/ Documents/
Desktop:
date.pdf day.txt xrguy.txt

Documents:

Downloads:
mohan@mohan:~$
```

With Wildcards:



```
mohan@mohan:~$ ls D*
Desktop:
date.pdf day.txt xrguy.txt

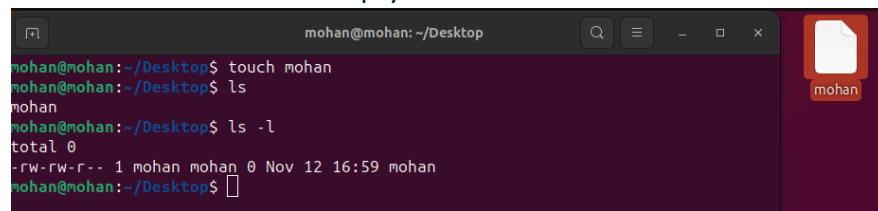
Documents:

Downloads:
mohan@mohan:~$
```

- * → Anything
- ? → will match one place (?txt matches A.txt, B.txt, C.txt...)
- [a-b],[abc],[A-Z],[012345],[ABC] → matches contents mentioned in the square braces for one place.

Creating Files and Folders:

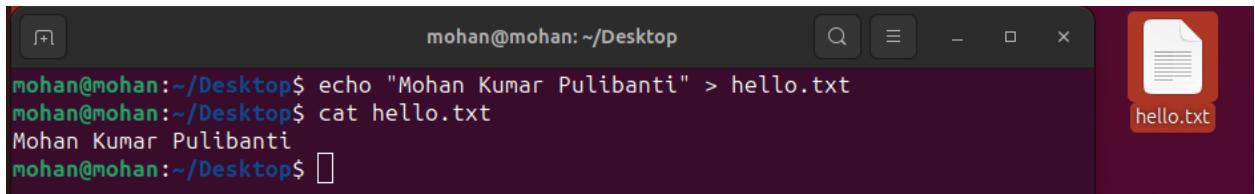
- **touch** → is used to create new empty files.



```
mohan@mohan:~/Desktop$ touch mohan
mohan@mohan:~/Desktop$ ls
mohan
mohan@mohan:~/Desktop$ ls -l
total 0
-rw-rw-r-- 1 mohan mohan 0 Nov 12 16:59 mohan
mohan@mohan:~/Desktop$
```

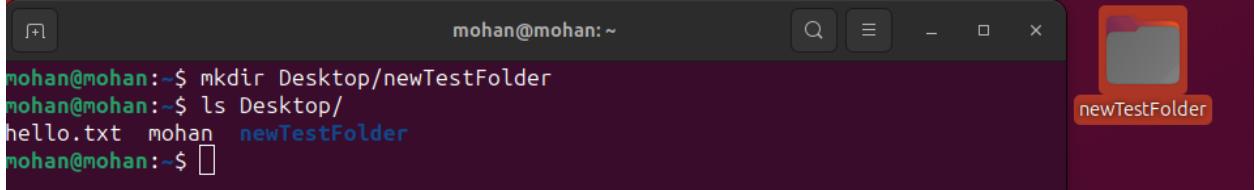
- Create file and add content to it.

```
echo "Mohan Kumar Pulibanti" > hello.txt
```



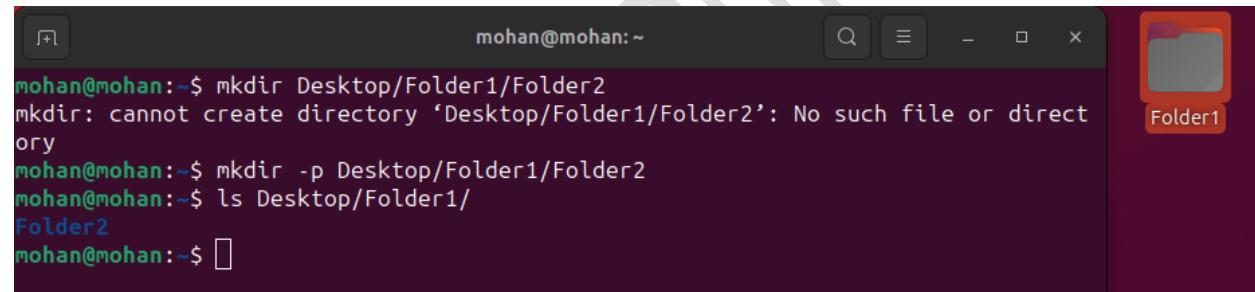
```
mohan@mohan:~/Desktop$ echo "Mohan Kumar Pulibanti" > hello.txt
mohan@mohan:~/Desktop$ cat hello.txt
Mohan Kumar Pulibanti
mohan@mohan:~/Desktop$
```

- **mkdir** → used to create folders



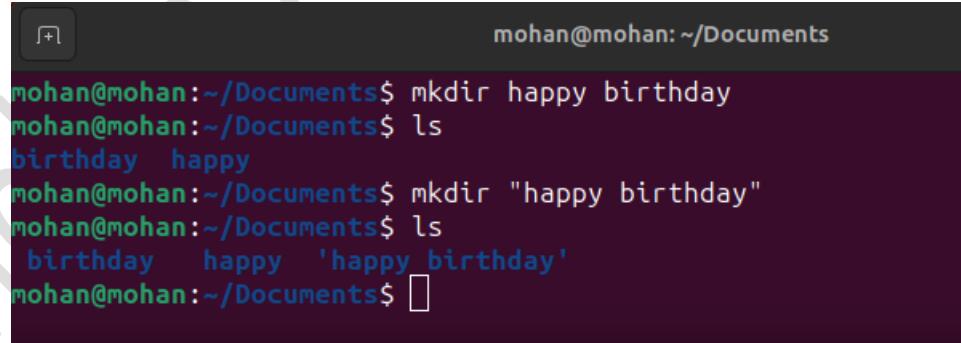
```
mohan@mohan:~$ mkdir Desktop/newTestFolder
mohan@mohan:~$ ls Desktop/
hello.txt mohan newTestFolder
mohan@mohan:~$
```

- if we want to create nested folders, mkdir (i.e., mkdir Desktop/Folder1/Folder2 – to create two folders Folder1 and Folder2 in desktop) throws an error. We must use “**mkdir -p <directory>**”

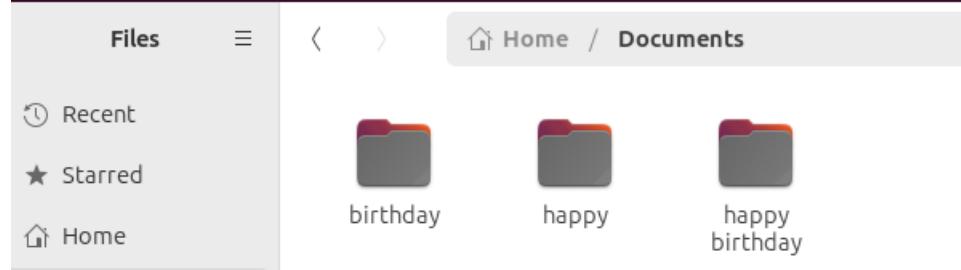


```
mohan@mohan:~$ mkdir Desktop/Folder1/Folder2
mkdir: cannot create directory 'Desktop/Folder1/Folder2': No such file or directory
mohan@mohan:~$ mkdir -p Desktop/Folder1/Folder2
mohan@mohan:~$ ls Desktop/Folder1/
Folder2
mohan@mohan:~$
```

- **mkdir happy birthday** → this will create two folders happy and birthday
- **mkdir "happy birthday"** → this will create one folder with name 'happy birthday'



```
mohan@mohan:~/Documents$ mkdir happy birthday
mohan@mohan:~/Documents$ ls
birthday happy
mohan@mohan:~/Documents$ mkdir "happy birthday"
mohan@mohan:~/Documents$ ls
birthday happy 'happy birthday'
mohan@mohan:~/Documents$
```



Brace Expansion

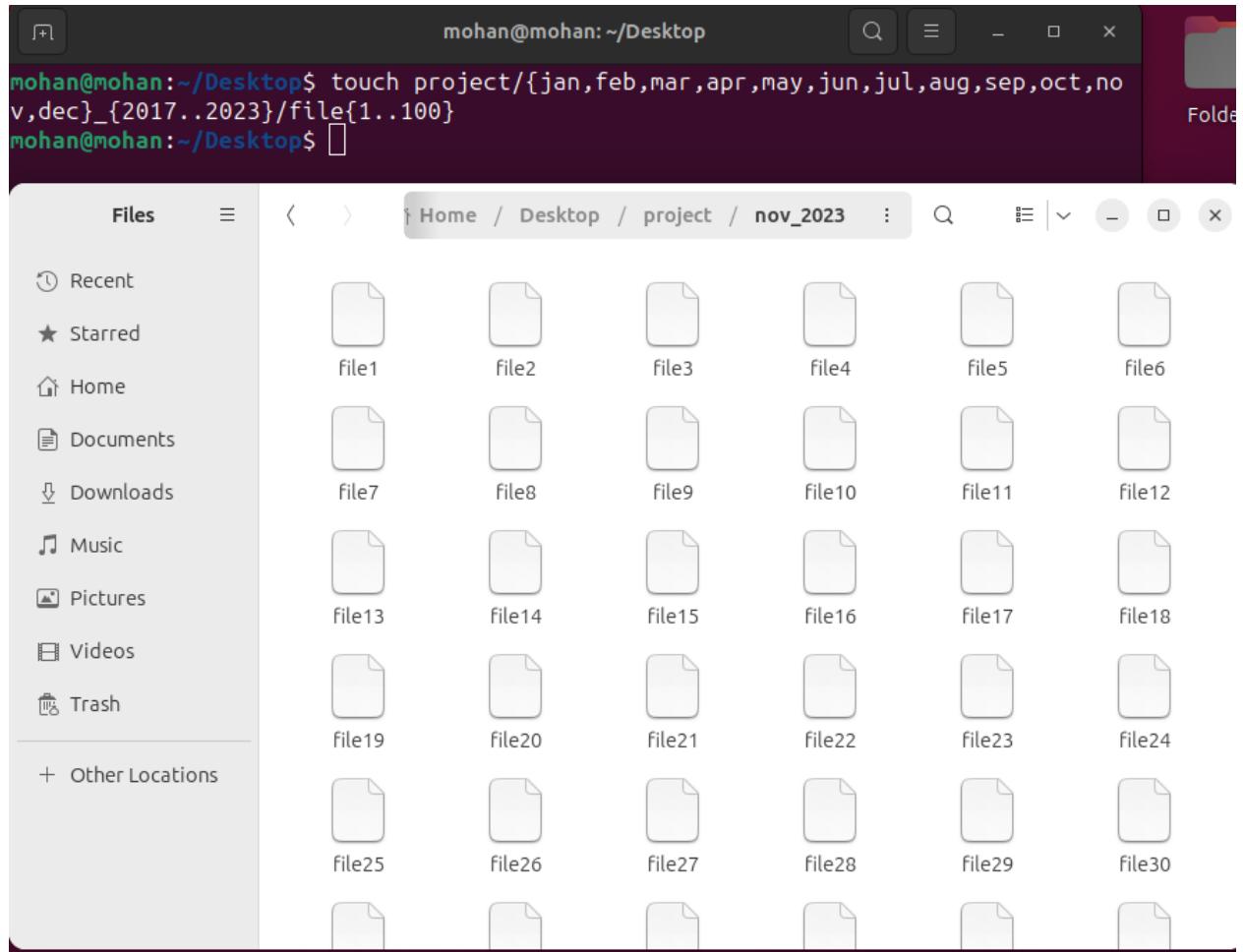
- brace expansion can be used to do complex operations. Scenarios below
- If we want to create a huge number of folders for every month and year from 2017 to 2023, we can do that with single command using Brace Expansion as below.

The terminal window shows the command:

```
mohan@mohan:~/Documents$ cd ..  
mohan@mohan:~$ cd Desktop/  
mohan@mohan:~/Desktop$ mkdir -p project/{jan,feb,mar,apr,mai,jun,jul,aug,sep,oct,nov,dec}_{2017..2023}  
mohan@mohan:~/Desktop$
```

The file manager window shows the directory structure created by the command. It contains a 'project' folder with subfolders for each month from January to December for each year from 2017 to 2023. The subfolders are named like 'jan_2017', 'feb_2017', etc.

- We can shorten the above command in the below way
mkdir -p project/{jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov, dec}_{2017..2023}
- if we want to create 100 files in each of the above folders with names as file1,file2....file100 below is the command
touch project/{jan,fe,mar,apr,mai,jun,jul,aug,sep,oct,nov,dec}_{2017..2023}/file{1..100}



Deleting Files and Folders

- **rm <file_path>** → to remove files. (Wild cards can also be used along with rm)

```
mohan@mohan:~/Desktop$ ls
older1 hello.txt mohan newTestFolder project
mohan@mohan:~/Desktop$ rm hello.txt
mohan@mohan:~/Desktop$ ls
older1 mohan newTestFolder project
mohan@mohan:~/Desktop$
```

- **rm** can't be used to delete directories without providing -r option

```
mohan@mohan:~/Desktop$ ls
Folder1 mohan newTestFolder project
mohan@mohan:~/Desktop$ rm Folder1/
rm: cannot remove 'Folder1/': Is a directory
mohan@mohan:~/Desktop$
```

- **rmdir** can be used to delete empty directories, it can't delete non-empty directories.

```
mohan@mohan:~/Desktop$ rmdir Folder1
rmdir: failed to remove 'Folder1': Directory not empty
mohan@mohan:~/Desktop$ ls
Folder1 mohan newTestFolder project
mohan@mohan:~/Desktop$ rmdir newTestFolder/
mohan@mohan:~/Desktop$ ls
Folder1 mohan project
mohan@mohan:~/Desktop$
```

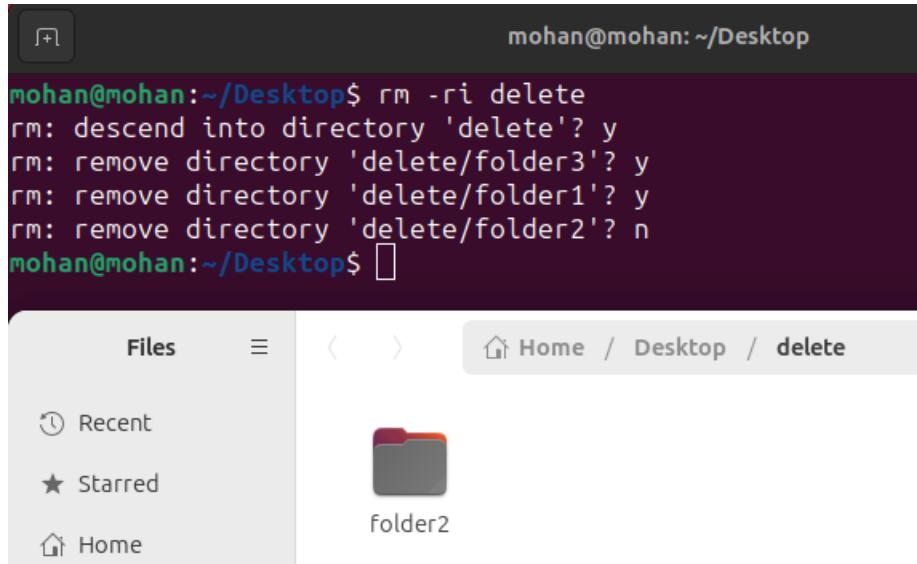
- **rm -r <directory-path>** → to delete non-empty directories recursively.

```
mohan@mohan:~/Desktop$ rmdir Folder1/
rmdir: failed to remove 'Folder1/': Directory not empty
mohan@mohan:~/Desktop$ ls
Folder1 project
mohan@mohan:~/Desktop$ rm -r Folder1/
mohan@mohan:~/Desktop$ ls
project
mohan@mohan:~/Desktop$
```

- recursively deleting is a very dangerous operation. We can control deleting the files using the interactive option.

rm -ri <directorypath>

Every time we delete a file, it asks us yes/No



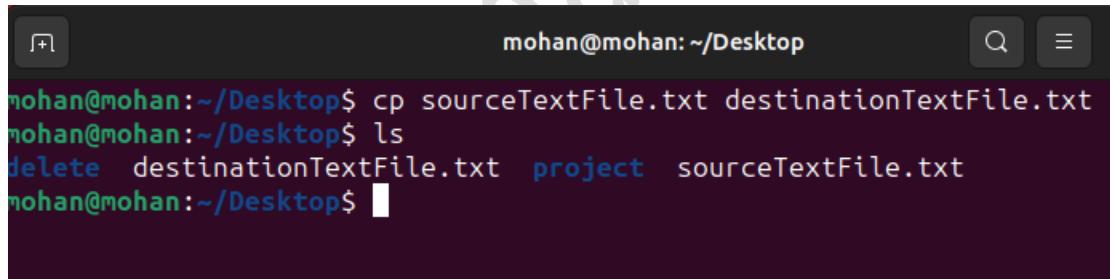
```
mohan@mohan:~/Desktop$ rm -ri delete
rm: descend into directory 'delete'? y
rm: remove directory 'delete/folder3'? y
rm: remove directory 'delete/folder1'? y
rm: remove directory 'delete/folder2'? n
mohan@mohan:~/Desktop$
```

Files ⌂ Recent ⌂ Starred ⌂ Home Home / Desktop / delete

Folder2

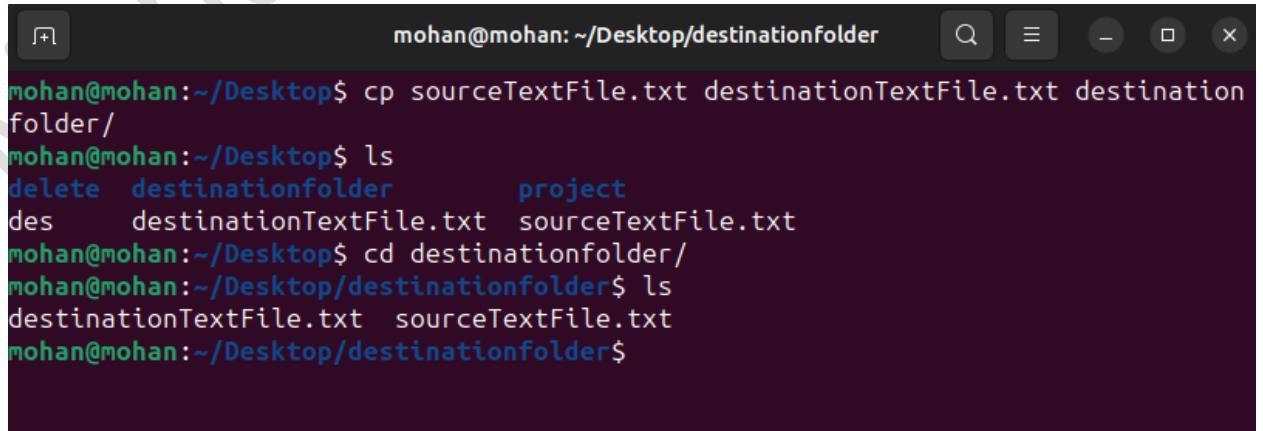
Copying Files and Folders

- cp <Source filepath to copy> <destination filepath to paste>



```
mohan@mohan:~/Desktop$ cp sourceTextFile.txt destinationTextFile.txt
mohan@mohan:~/Desktop$ ls
delete destinationTextFile.txt project sourceTextFile.txt
mohan@mohan:~/Desktop$
```

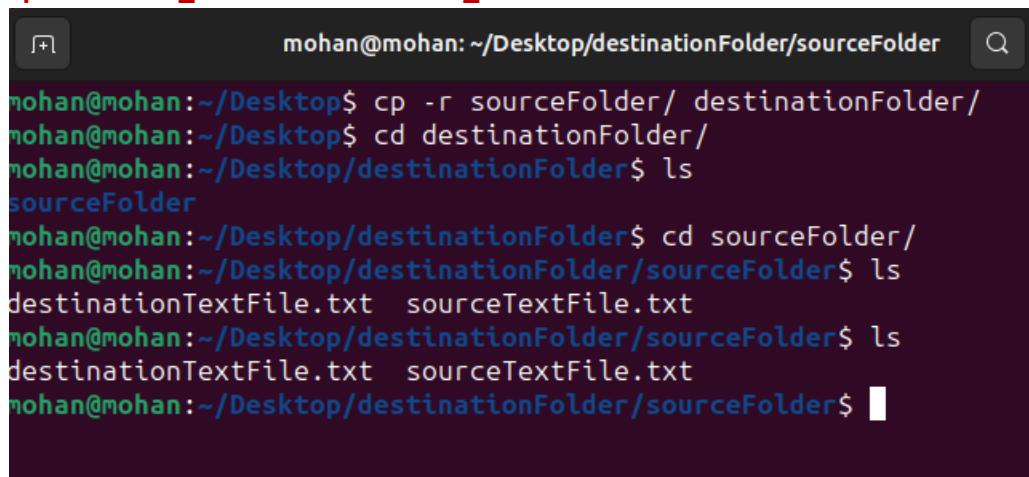
- Copying Multiple text files to destination folder



```
mohan@mohan:~/Desktop$ cp sourceTextFile.txt destinationTextFile.txt destination
folder/
mohan@mohan:~/Desktop$ ls
delete destinationfolder project
des destinationTextFile.txt sourceTextFile.txt
mohan@mohan:~/Desktop$ cd destinationfolder/
mohan@mohan:~/Desktop/destinationfolder$ ls
destinationTextFile.txt sourceTextFile.txt
mohan@mohan:~/Desktop/destinationfolder$
```

- Copy the entire folder to a destination folder command.

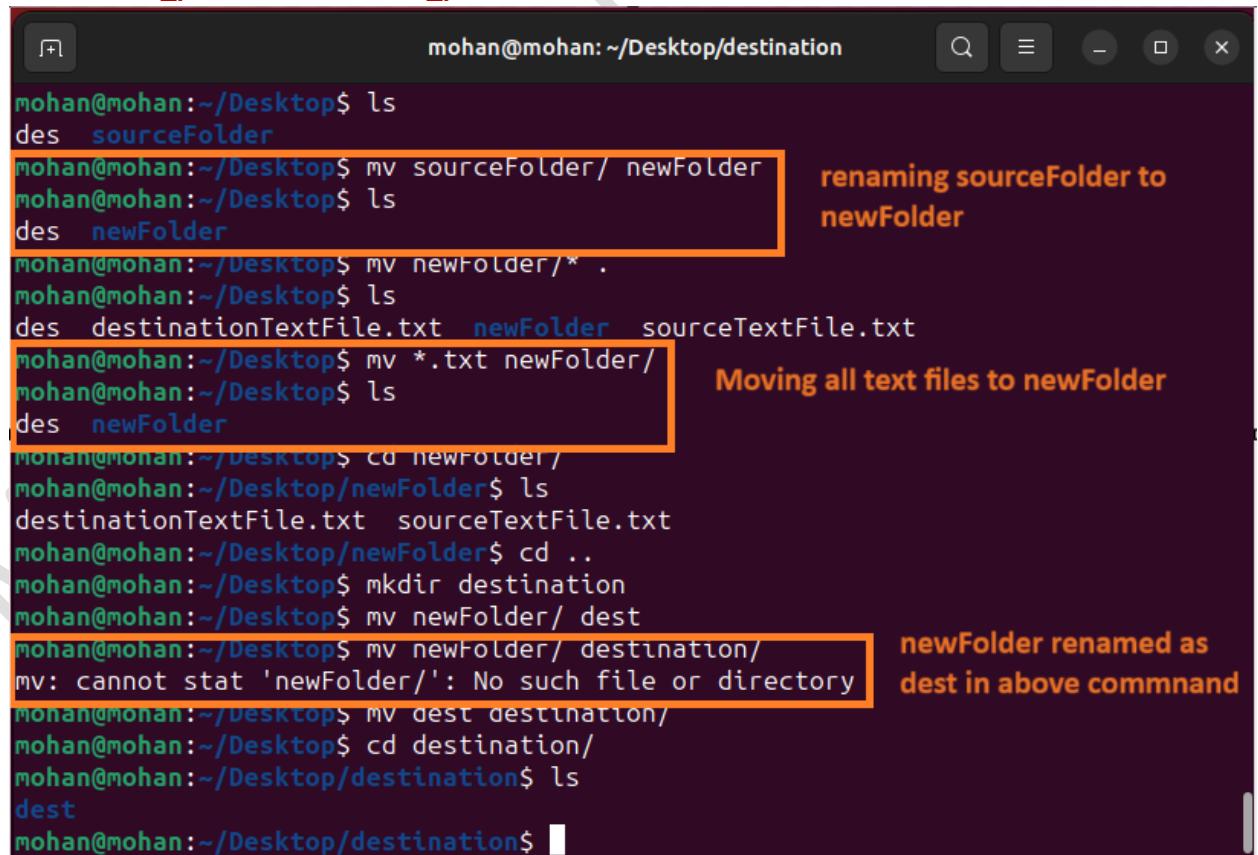
`cp -r <source_folder> <destination_folder>`



```
mohan@mohan:~/Desktop/destinationFolder/sourceFolder$ cp -r sourceFolder/ destinationFolder/
mohan@mohan:~/Desktop$ cd destinationFolder/
mohan@mohan:~/Desktop/destinationFolder$ ls
sourceFolder
mohan@mohan:~/Desktop/destinationFolder$ cd sourceFolder/
mohan@mohan:~/Desktop/destinationFolder/sourceFolder$ ls
destinationTextFile.txt sourceTextFile.txt
mohan@mohan:~/Desktop/destinationFolder/sourceFolder$ ls
destinationTextFile.txt sourceTextFile.txt
mohan@mohan:~/Desktop/destinationFolder/sourceFolder$
```

Moving + renaming Files and Directories

- `mv <source_path> <destination_path>` → move files from source to destination.



```
mohan@mohan:~/Desktop$ ls
des sourceFolder
mohan@mohan:~/Desktop$ mv sourceFolder/ newFolder
mohan@mohan:~/Desktop$ ls
des newFolder
mohan@mohan:~/Desktop$ mv newFolder/* .
mohan@mohan:~/Desktop$ ls
des destinationTextFile.txt newFolder sourceTextFile.txt
mohan@mohan:~/Desktop$ mv *.txt newFolder/
mohan@mohan:~/Desktop$ ls
des newFolder
mohan@mohan:~/Desktop$ cd newFolder/
mohan@mohan:~/Desktop/newFolder$ ls
destinationTextFile.txt sourceTextFile.txt
mohan@mohan:~/Desktop/newFolder$ cd ..
mohan@mohan:~/Desktop$ mkdir destination
mohan@mohan:~/Desktop$ mv newFolder/ dest
mohan@mohan:~/Desktop$ mv newFolder/ destination/
mv: cannot stat 'newFolder/': No such file or directory
mohan@mohan:~/Desktop$ mv dest destination/
mohan@mohan:~/Desktop$ cd destination/
mohan@mohan:~/Desktop/destination$ ls
dest
mohan@mohan:~/Desktop/destination$
```

renaming sourceFolder to newFolder

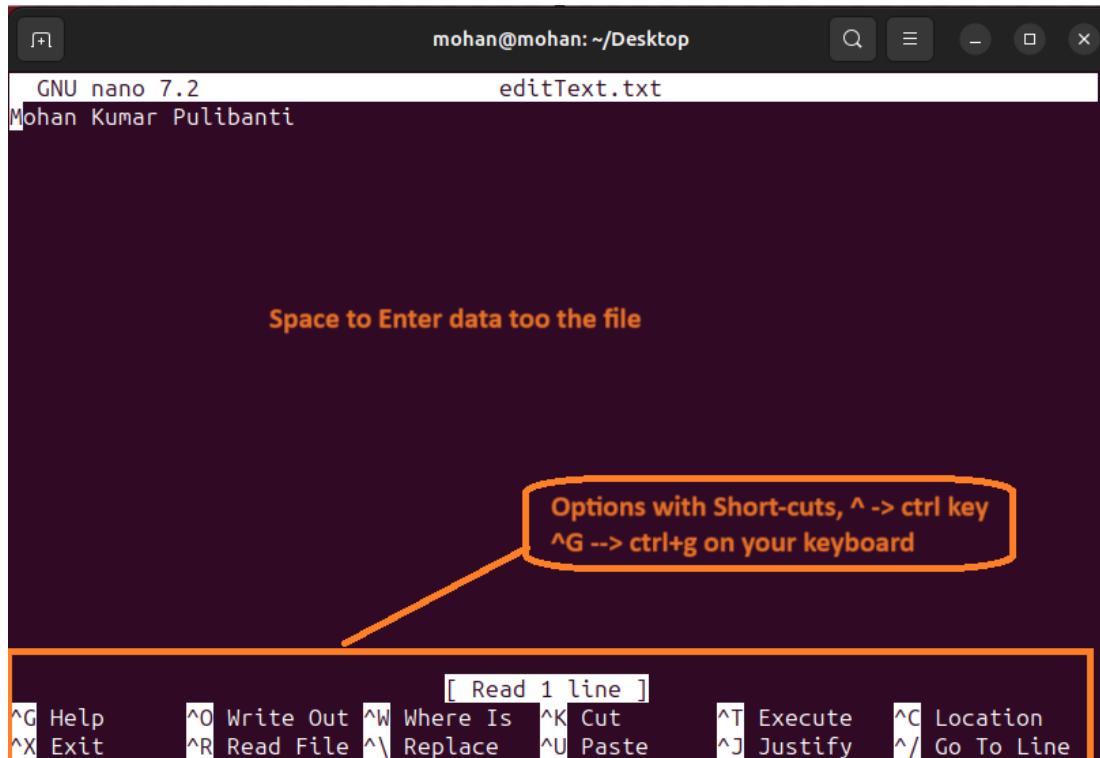
Moving all text files to newFolder

newFolder renamed as dest in above command

Editing Files using Nano

- **nano <file_path>** - used to edit files in the terminal.

Type “nano editText.txt” in your terminal and the below window will open along with options to do different operations on the editing window



- **^G Help** → opens help window
- **^X Exit** → to Exit editing mode
- **^O Write Out** → to write something to the file. Add the content in the space to enter data and then click Ctrl+o to save that data to the file.
- **^R Read File** → to read data from a different file and append to this file.
- **^ W Where Is** → searches for a particular word in the file and highlights the first found match (Searches in a Case insensitive manner)

The screenshot shows a terminal window titled "mohan@mohan: ~/Desktop" running the "GNU nano 7.2" editor. The file being edited is "editTex.txt". The content of the file includes several lines of text: "Mohan Kumar Pulibanti", "Mohan Kumar Pulibanti", "Ram Kumar Pulibanti", "Linux Tutorial", "testing echo read", "Ram Kumar Pulibanti", "Linux Tutorial", "Linux Tutorial", "Linux Tutorial", and "Linux Tutorial". At the bottom of the screen, there is a search bar with the placeholder text "Search [read]:" and a series of keyboard shortcut keys. An orange callout box highlights the key "M-B" in the list, which corresponds to the search history. The keyboard shortcut keys listed are: ^G Help, M-C Case Sens, M-B Backwards, ^P Older, ^T Go To Line, ^C Cancel, M-R Reg.exp., ^R Replace, and ^N Newer.

M can be Alt, Cmd, Esc depending on your keyboard layout

Search [read]:							
^G Help	M-C Case Sens	M-B Backwards	^P Older	^T Go To Line			
^C Cancel	M-R Reg.exp.	^R Replace	^N Newer				

- M-B → press Alt+B, this option will search the file in the backward direction from the cursor position and when it finds the string given to be searched, it highlights that.
- M-C → press Alt+C, to search the file as case sensitive, default would be case insensitive.
- ^P , ^N → to traverse through the strings we searched from search history.

```
mohan@mohan: ~/Desktop
GNU nano 7.2          editTex.txt *
Mohan Kumar Viyyampeta
Mohan Kumar Viyyampeta
Ram Kumar Viyyampeta
Linux Tutorial
testing echo read
Ram Kumar Viyyampeta
Linux Tutorial
Linux Tutorial
Linux Tutorial
Linux Tutorial
```

[Cancelled]

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line

-
- **^\ Replace** → Searches and replaces a particular string in the file. It can replace one instance at a time in interactive mode / all at one go.
- **^K Cut** → cuts a particular line where the pointer is
- **^U Paste** → pastes the text that was cut at the position where pointer is placed. If nothing is cut, throws error msg as Cut Buffer is Empty.
- **^T Execute** → to execute different operations like Spell Check, Cut Till End etc.,

```
mohan@mohan: ~/Desktop
GNU nano 7.2          editTex.txt *
Mohan Kumar Viyyampeta Mohan Kumar Viyyampeta Ram Kumar Viyyampeta Linux
Tutorial testing echo read Ram Kumar Viyyampeta Linux Tutorial Linux
Tutorial Linux Tutorial Linux Tutorial
```

Command to execute:

^G Help M-F New Buffer ^S Spell Check ^J Full Justify ^V Cut Till End
^C Cancel M-\ Pipe Text ^Y Linter ^O Formatter ^Z Suspend

- Steps to enable spell Check:
 - 1) sudo nano /etc/nanorc
 - 2) Search “aspell” and uncomment line set speller “aspell -x -c”
 - 3) Save and Exit
- ^J Justify → to bring multiple line to some smaller number of lines (Justifies)
- ^C Location → gives the location (row number and column number) of the current pointer position.
- ^/ Go To Line → when we give row and column numbers it takes us to that position
- M-U → Alt+u is to undo
- M-E → Alt+e is to redo

GNU nano 7.2

mohan@mohan: ~/Desktop

editTex.txt

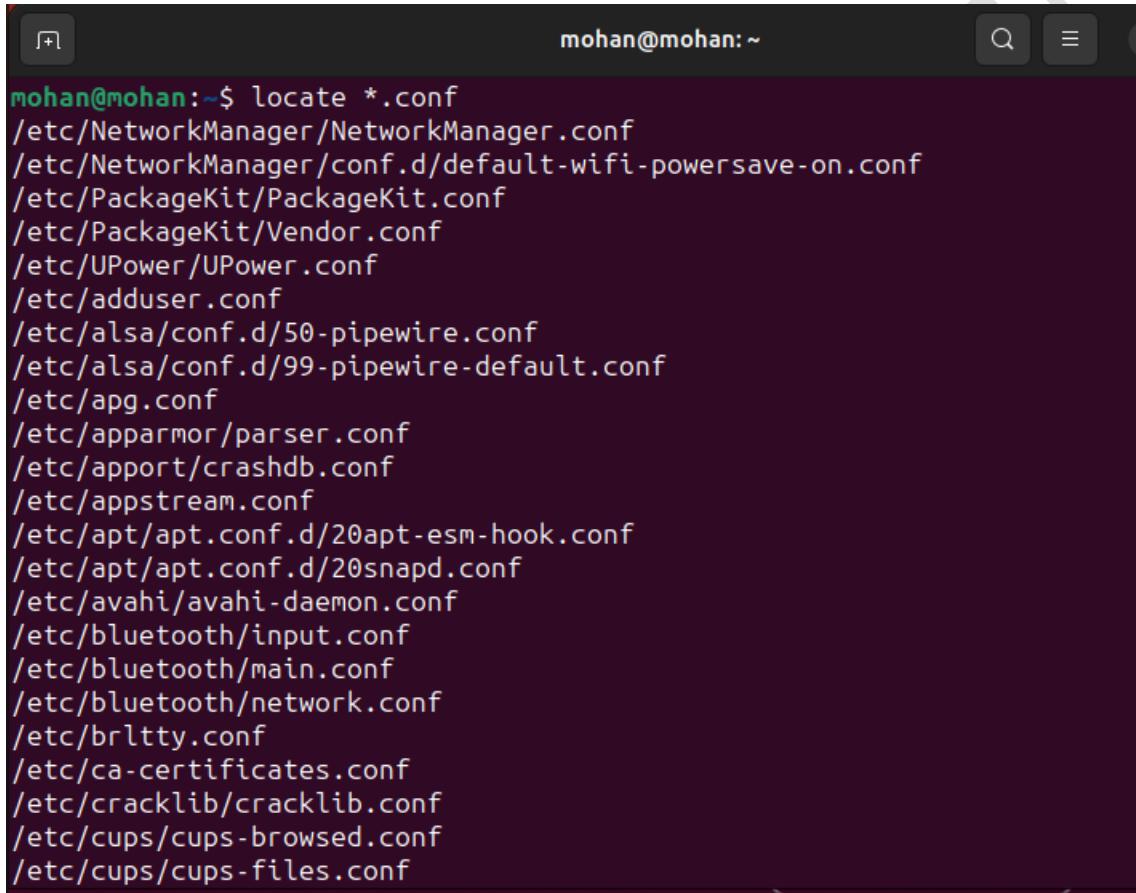
lohan Kumar Viyyampeta
lohan Kumar Viyyampeta
am Kumar Viyyampeta
linux Tutorial
esting echo read
am Kumar Viyyampeta
inux Tutorial
inux Tutorial
inux Tutorial
inux Tutorial

[Undid justification]

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line M-E Redo

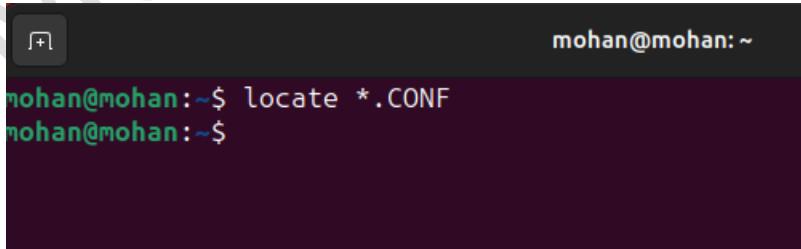
Locate Command

- Installing locate and mlocate commands
 - sudo apt install locate**
 - sudo apt install mlocate**
- locate command will give all matching paths for your search on the standard output. It works by searching a database file that's on your system. This database file holds all location information about every file that's on your system.



```
mohan@mohan:~$ locate *.conf
/etc/NetworkManager/NetworkManager.conf
/etc/NetworkManager/conf.d/default-wifi-powersave-on.conf
/etc/PackageKit/PackageKit.conf
/etc/PackageKit/Vendor.conf
/etc/UPower/UPower.conf
/etc/adduser.conf
/etc/alsa/conf.d/50-pipewire.conf
/etc/alsa/conf.d/99-pipewire-default.conf
/etc/apg.conf
/etc/apparmor/parser.conf
/etc/apport/crashdb.conf
/etc/appstream.conf
/etc/apt/apt.conf.d/20apt-esm-hook.conf
/etc/apt/apt.conf.d/20snapd.conf
/etc/avahi/avahi-daemon.conf
/etc/bluetooth/input.conf
/etc/bluetooth/main.conf
/etc/bluetooth/network.conf
/etc/brltty.conf
/etc/ca-certificates.conf
/etc/cracklib/cracklib.conf
/etc/cups/cups-browsed.conf
/etc/cups/cups-files.conf
```

- locate search is case sensitive, if we want to search case-insensitive way we have to use option **-i**



```
mohan@mohan:~$ locate *.CONF
mohan@mohan:~$
```

```
mohan@mohan:~$ locate *.CONF
mohan@mohan:~$ locate -i *.CONF
/etc/NetworkManager/NetworkManager.conf
/etc/NetworkManager/conf.d/default-wifi-powersave-on.conf
/etc/PackageKit/PackageKit.conf
/etc/PackageKit/Vendor.conf
/etc/UPower/UPower.conf
/etc/adduser.conf
/etc/alsa/conf.d/50-pipewire.conf
/etc/alsa/conf.d/99-pipewire-default.conf
/etc/apg.conf
/etc/apparmor/parser.conf
/etc/apport/crashdb.conf
/etc/appstream.conf
/etc/apt/apt.conf.d/20apt-esm-hook.conf
/etc/apt/apt.conf.d/20snapd.conf
/etc/avahi/avahi-daemon.conf
/etc/bluetooth/input.conf
/etc/bluetooth/main.conf
/etc/bluetooth/network.conf
/etc/brltty.conf
/etc/ca-certificates.conf
/etc/cracklib/cracklib.conf
/etc/cups/cups_browsed.conf
```

- limit the locate results to a specific number

locate -i --limit 3 *.conf

```
mohan@mohan:~$ locate -i --limit 3 *.conf
/etc/NetworkManager/NetworkManager.conf
/etc/NetworkManager/conf.d/default-wifi-powersave-on.conf
/etc/PackageKit/PackageKit.conf
mohan@mohan:~$
```

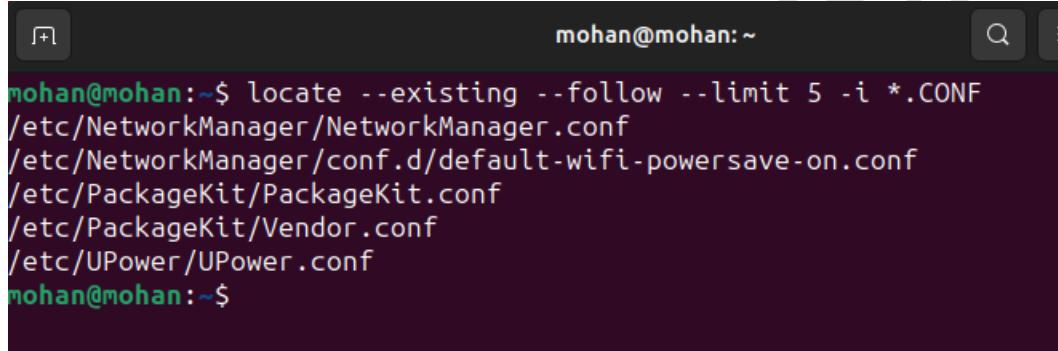
- to get the database details, in which locate command search

locate -S

```
mohan@mohan:~$ locate -S
Database /var/cache/locate/locatedb is in the GNU LOCATE02 format.
Database was last modified at 2023:11:14 00:18:43.243171667 -0500
Locate database size: 3532166 bytes
All Filenames: 272620
File names have a cumulative length of 17771571 bytes.
Of those file names,
        20 contain whitespace,
        0 contain newline characters,
        and 4 contain characters with the high bit set.
Compression ratio 80.12% (higher is better)
mohan@mohan:~$
```

- The database that is used by locate is updated just once a day, in the meantime if we add any files to our system, the database will not have those details and the locate command will not be able to locate those files until the database is updated.
- Sometimes we may move/delete the files but if these details are not updated in the database, if we use the locate command, we get the wrong results. To minimize these wrong results, we can use **locate -e *.conf / locate --existing *.conf** → The option -e not only fetches the details from the database but also checks if the file exists in the system or not before giving it to the standard output.
- Sometimes there are shortcuts and links on your system called symbolic links, that can be broken if files get moved around. To check if these shortcuts and links are still valid before the locate command reports them back, we can use

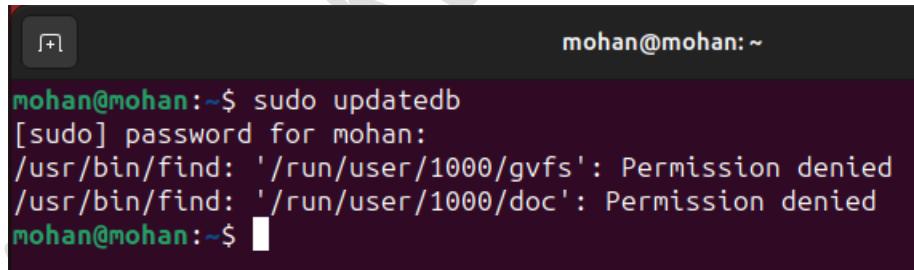
locate --follow *.conf



```
mohan@mohan:~$ locate --existing --follow --limit 5 -i *.CONF
/etc/NetworkManager/NetworkManager.conf
/etc/NetworkManager/conf.d/default-wifi-powersave-on.conf
/etc/PackageKit/PackageKit.conf
/etc/PackageKit/Vendor.conf
/etc/UPower/UPower.conf
mohan@mohan:~$
```

- The above commands are all safety measures, but when we want to update the database whenever we want to get the accurate results when we use locate

sudo updatedb



```
mohan@mohan:~$ sudo updatedb
[sudo] password for mohan:
/usr/bin/find: '/run/user/1000/gvfs': Permission denied
/usr/bin/find: '/run/user/1000/doc': Permission denied
mohan@mohan:~$
```

- Using a database is very fast but requires updating. Database automatically gets updated only once a day.

Find Command

- find command will return all the files and folders (to infinite depths) under the current directory in which it's executed.

find

```
mohan@mohan:~$ cd Desktop/  
mohan@mohan:~/Desktop$ find  
. ./data  
./data/editText.txt  
./editTex.txt  
mohan@mohan:~/Desktop$
```

- find command will returns files and folders, whereas locate command will return only folders.
- find command will not use database, so it's up to date and accurate but it's slower compared to locate.
- We can also give path parameter to find command, to fetch the files and folders in that directory.

```
mohan@mohan:~$ find Documents/  
Documents/  
Documents/birthday  
Documents/happy birthday  
Documents/happy  
mohan@mohan:~$
```

- Command to control the depth of the search command
find . -maxdepth 2 → find all the files and folders in the current directory(.) to a depth of 2

```
mohan@mohan:~$ find  
. ./data  
./data/Level1  
./data/Level1/Level2  
./data/Level1/Level2/Level3  
./data/Level1/Level2/Level3/Level4  
./data/Level1/Level2/Level3/Level4/Level5  
./data/Level1/Level2/Level3/Level4/Level5/editText.txt  
./data/Level1/editText.txt  
./data/editText.txt  
./editTex.txt  
mohan@mohan:~/Desktop$ find . -maxdepth 2  
. ./data  
./data/Level1  
./data/editText.txt  
./editTex.txt  
mohan@mohan:~/Desktop$
```

- By default, find command will search for both files and folders, if we want to search for only a specific type, below is the command.

find . -type f → to find only files

find . -type d → to find only directories

```
mohan@mohan:~/Desktop$ find . -type f
./data/Level1/Level2/Level3/Level4/Level5/editText.txt
./data/Level1/editText.txt
./data/editText.txt
./editTex.txt
mohan@mohan:~/Desktop$ find . -type d
.
./data
./data/Level1
./data/Level1/Level2
./data/Level1/Level2/Level3
./data/Level1/Level2/Level3/Level4
./data/Level1/Level2/Level3/Level4/Level5
mohan@mohan:~/Desktop$
```

- maxdepth option should always come before type option.

find . -maxdepth 1 -type f

- if we want to find a specific file using name

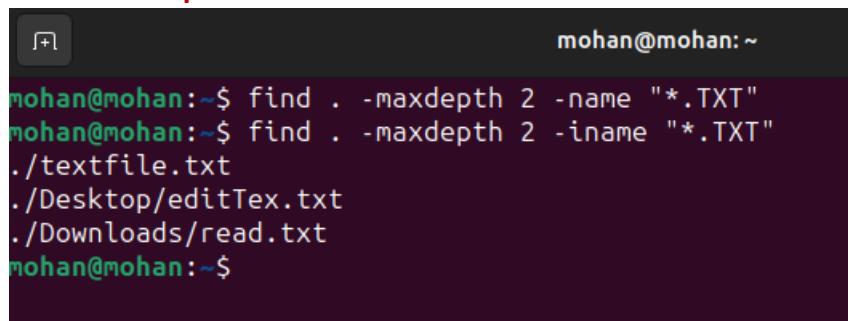
find . -name "edittext.txt"

```
mohan@mohan:~
mohan@mohan:~$ find . -name "editTex.txt"
./Desktop/editTex.txt
mohan@mohan:~$
```

```
mohan@mohan:~
mohan@mohan:~$ find . -maxdepth 2 -name "*.txt"
./textfile.txt
./Desktop/editTex.txt
./Downloads/read.txt
mohan@mohan:~$
```

- to find the files using case-insensitive file names, below is the command.

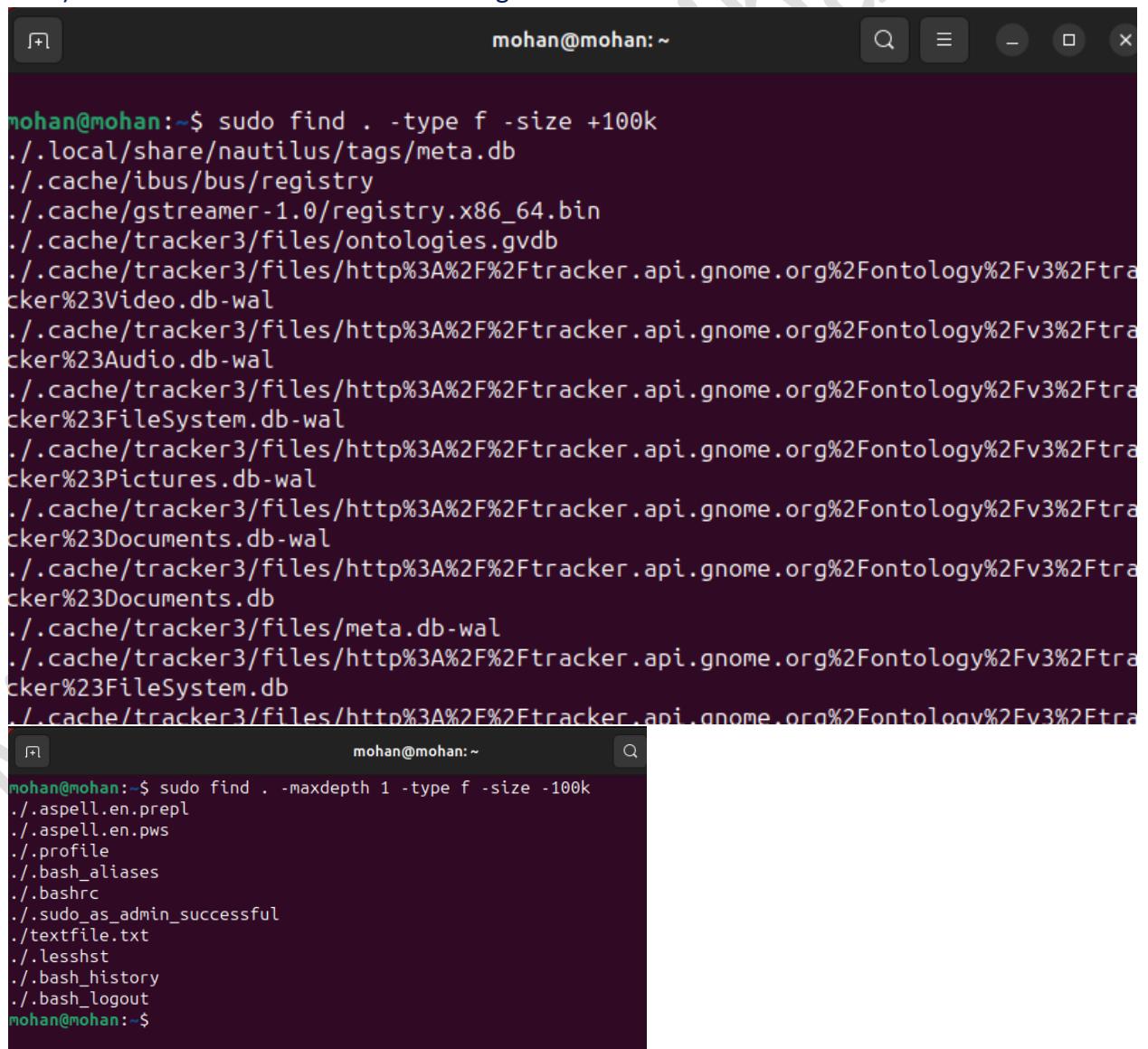
find . -maxdepth 1 -iname “?.TXT”



```
mohan@mohan:~$ find . -maxdepth 2 -name "*.TXT"
mohan@mohan:~$ find . -maxdepth 2 -iname "*.TXT"
./textfile.txt
./Desktop/editTex.txt
./Downloads/read.txt
mohan@mohan:~$
```

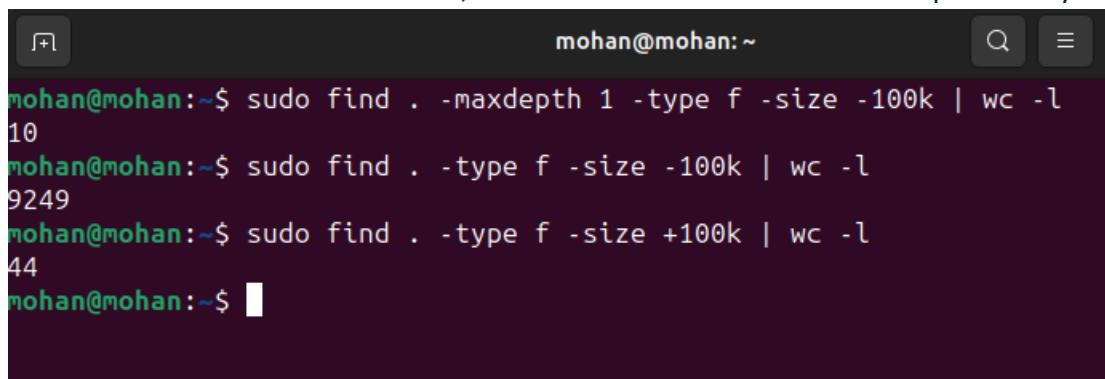
- to find the files based on its size

find / -maxdepth 1 -iname “?.txt” -size +100k (+ indicates more than/- indicates less than) → returns all the files whose size is greater than 100k



```
mohan@mohan:~$ sudo find . -type f -size +100k
./.local/share/nautilus/tags/meta.db
./.cache/ibus/bus/registry
./.cache/gstremer-1.0/registry.x86_64.bin
./.cache/tracker3/files/ontologies.gvdb
./.cache/tracker3/files/http%3A%2F%2Ftracker.api.gnome.org%2Fontology%2Fv3%2Ftrac
ker%23Video.db-wal
./.cache/tracker3/files/http%3A%2F%2Ftracker.api.gnome.org%2Fontology%2Fv3%2Ftrac
ker%23Audio.db-wal
./.cache/tracker3/files/http%3A%2F%2Ftracker.api.gnome.org%2Fontology%2Fv3%2Ftrac
ker%23FileSystem.db-wal
./.cache/tracker3/files/http%3A%2F%2Ftracker.api.gnome.org%2Fontology%2Fv3%2Ftrac
ker%23Pictures.db-wal
./.cache/tracker3/files/http%3A%2F%2Ftracker.api.gnome.org%2Fontology%2Fv3%2Ftrac
ker%23Documents.db-wal
./.cache/tracker3/files/http%3A%2F%2Ftracker.api.gnome.org%2Fontology%2Fv3%2Ftrac
ker%23Documents.db
./.cache/tracker3/files/meta.db-wal
./.cache/tracker3/files/http%3A%2F%2Ftracker.api.gnome.org%2Fontology%2Fv3%2Ftrac
ker%23FileSystem.db
./.cache/tracker3/files/http%3A%2F%2Ftracker.api.gnome.org%2Fontology%2Fv3%2Ftrac
ker%23Filesystem.db
mohan@mohan:~$ sudo find . -maxdepth 1 -type f -size -100k
./.aspell.en.prepl
./.aspell.en.pws
./.profile
./.bash_aliases
./.bashrc
./.sudo_as_admin_successful
./textfile.txt
./.lessht
./.bash_history
./.bash_logout
mohan@mohan:~$
```

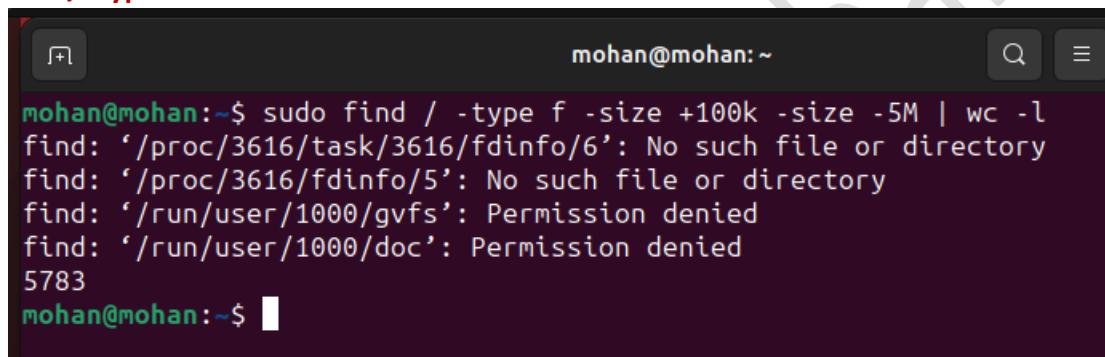
- **wc -l** → this returns the lines count, wc stands for word count and the option -l says lines.



```
mohan@mohan:~$ sudo find . -maxdepth 1 -type f -size -100k | wc -l
10
mohan@mohan:~$ sudo find . -type f -size -100k | wc -l
9249
mohan@mohan:~$ sudo find . -type f -size +100k | wc -l
44
mohan@mohan:~$
```

- to find file size bigger than 100KB and smaller 5MB

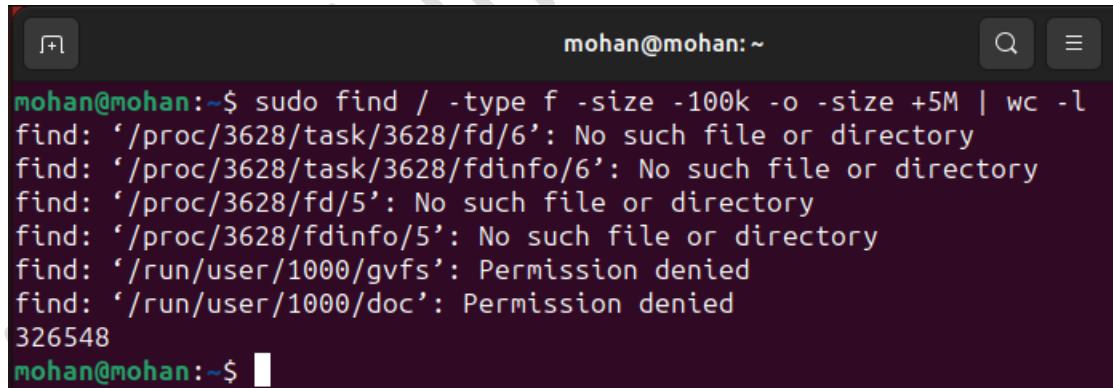
find / -type f -size +100k -size -5M



```
mohan@mohan:~$ sudo find / -type f -size +100k -size -5M | wc -l
find: '/proc/3616/task/3616/fdinfo/6': No such file or directory
find: '/proc/3616/fdinfo/5': No such file or directory
find: '/run/user/1000/gvfs': Permission denied
find: '/run/user/1000/doc': Permission denied
5783
mohan@mohan:~$
```

- to find files smaller than 100KB or bigger then 5MB below is the command.

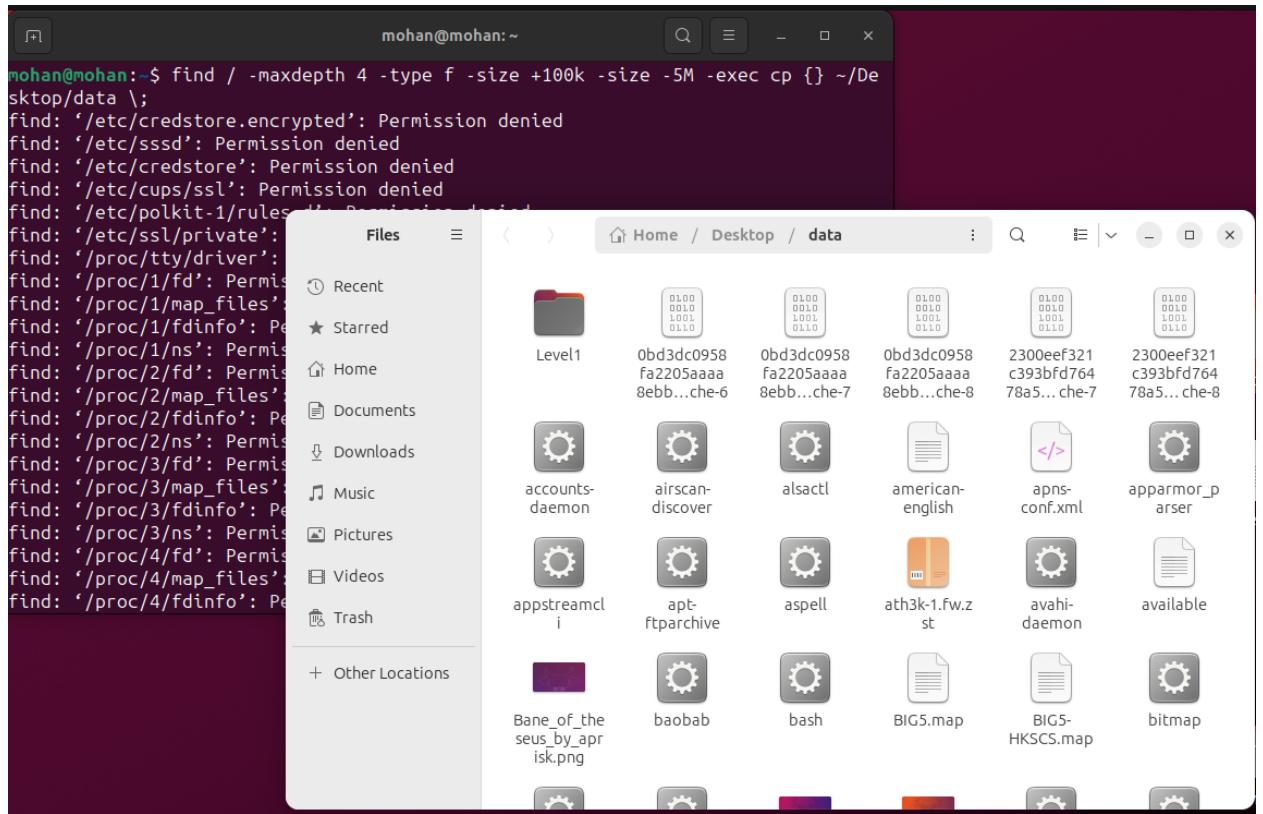
find / -type f -size -100K -o -size +5M → here -o stands for or



```
mohan@mohan:~$ sudo find / -type f -size -100k -o -size +5M | wc -l
find: '/proc/3628/task/3628/fd/6': No such file or directory
find: '/proc/3628/task/3628/fdinfo/6': No such file or directory
find: '/proc/3628/fd/5': No such file or directory
find: '/proc/3628/fdinfo/5': No such file or directory
find: '/run/user/1000/gvfs': Permission denied
find: '/run/user/1000/doc': Permission denied
326548
mohan@mohan:~$
```

- on the files found if we want to execute any operation, below is the command.

find / -maxdepth 4 -type f -size +100k -size -5M -exec cp {} ~/Desktop/Copyme \; → this finds all the files in root directory with size greater than 100MB and less than 5MB and copy those files to copyme folder on Desktop. (-exec is the command to execute a specific command, {} → everyfile that the file command found, \; to end the execute command)

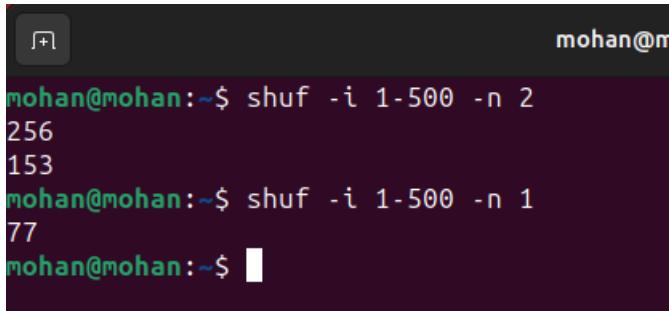


- exec command is not the safest of all things, instead of exec we can use -ok which will exactly do the same thing as exec, but instead of doing the action directly, it will ask us whether to do it or not and it proceeds with the action only based on our response.(if we type y it will continue, if we type n it will discard that)

find / -maxdepth 4 -type f -size +100k -size -5M -ok cp {} ~/Desktop/data \;



- command to pick a random folder out of 500 folders and create a text file in it.
touch data/Folder\$(shuf -i 1-500 -n 1)/needle.txt → shuf will randomly pick one number from 1 to 500, option n specifies how many numbers to pick and option -i specifies the range.

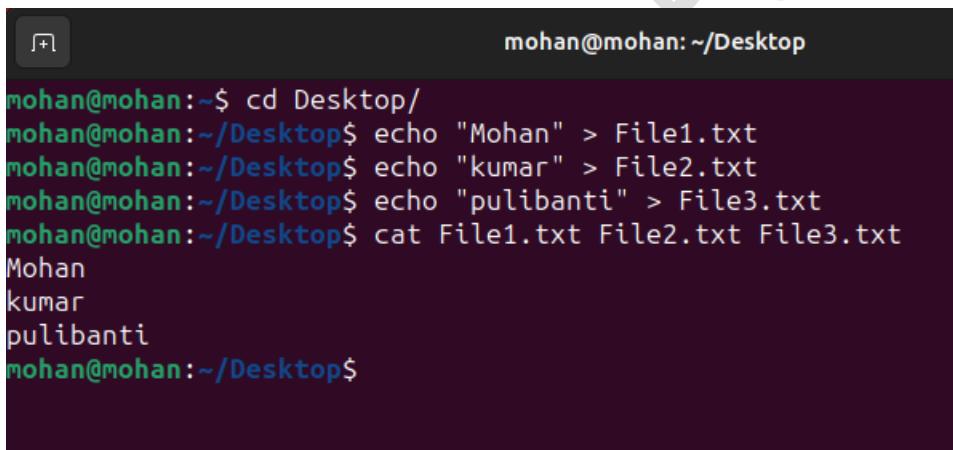


```
mohan@mohan:~$ shuf -i 1-500 -n 2
256
153
mohan@mohan:~$ shuf -i 1-500 -n 1
77
mohan@mohan:~$
```

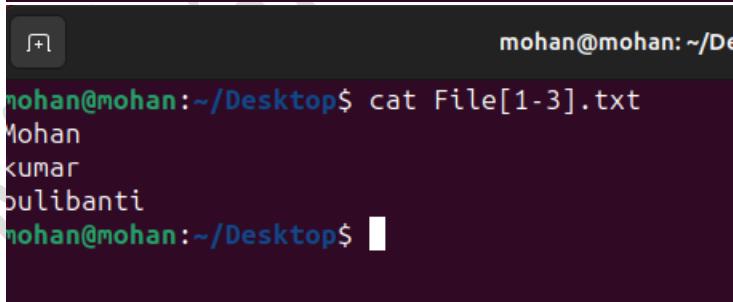
A screenshot of a terminal window titled "mohan@mohan". It shows two commands being run: "shuf -i 1-500 -n 2" which outputs the numbers 256 and 153, and "shuf -i 1-500 -n 1" which outputs the number 77.

Viewing Files

- cat <filepaths> → is to concatenate contents of all the files and outputs to the terminal.
cat file1.txt file2.txt file3.txt



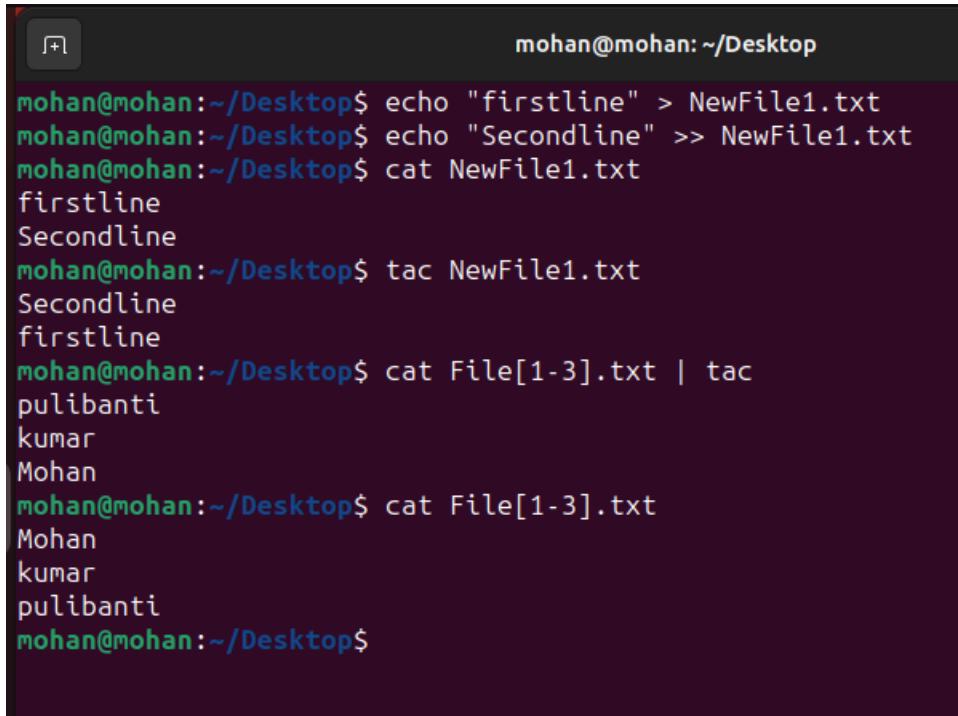
```
mohan@mohan:~/Desktop/
mohan@mohan:~/Desktop$ echo "Mohan" > File1.txt
mohan@mohan:~/Desktop$ echo "kumar" > File2.txt
mohan@mohan:~/Desktop$ echo "pulibanti" > File3.txt
mohan@mohan:~/Desktop$ cat File1.txt File2.txt File3.txt
Mohan
kumar
pulibanti
mohan@mohan:~/Desktop$
```

A screenshot of a terminal window titled "mohan@mohan: ~/Desktop". It shows the creation of three files ("File1.txt", "File2.txt", "File3.txt") using the "echo" command and then concatenating them with the "cat" command, resulting in the output "Mohan", "kumar", and "pulibanti".

```
mohan@mohan:~/Desktop$ cat File[1-3].txt
Mohan
kumar
pulibanti
mohan@mohan:~/Desktop$
```

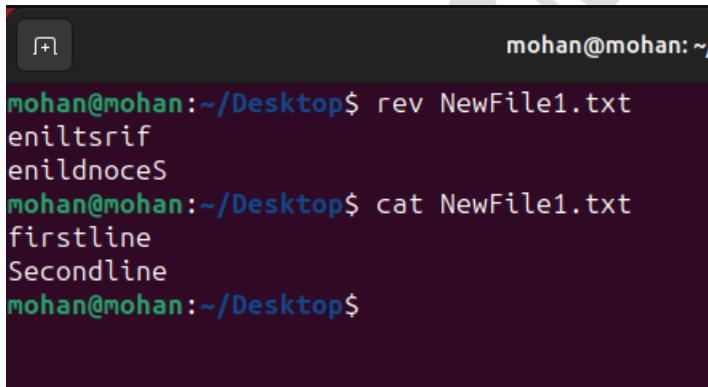
A screenshot of a terminal window titled "mohan@mohan: ~/Desktop". It shows the use of a wildcard ("File[1-3].txt") with the "cat" command, which concatenates the files "File1.txt", "File2.txt", and "File3.txt", resulting in the output "Mohan", "kumar", and "pulibanti".

- **tac <filepath>** → is to print the file contents in the reverse order [last line will be printed first, and first line will be printed last] – reversing a file vertically



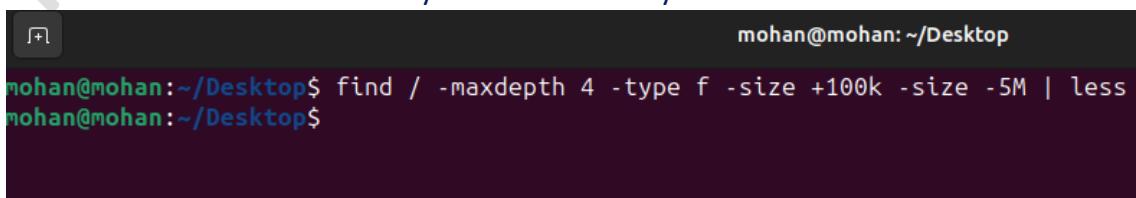
```
mohan@mohan:~/Desktop$ echo "firstline" > NewFile1.txt
mohan@mohan:~/Desktop$ echo "Secondline" >> NewFile1.txt
mohan@mohan:~/Desktop$ cat NewFile1.txt
firstline
Secondline
mohan@mohan:~/Desktop$ tac NewFile1.txt
Secondline
firstline
mohan@mohan:~/Desktop$ cat File[1-3].txt | tac
pulibanti
kumar
Mohan
mohan@mohan:~/Desktop$ cat File[1-3].txt
Mohan
kumar
pulibanti
mohan@mohan:~/Desktop$
```

- **rev <filepath>** - order of each line in the file content will be same but it reverses horizontally (for instance on ‘mohan’, if tac is applied output would be ‘nahom’)



```
mohan@mohan:~/Desktop$ rev NewFile1.txt
eniltsrif
enildnoceS
mohan@mohan:~/Desktop$ cat NewFile1.txt
firstline
Secondline
mohan@mohan:~/Desktop$
```

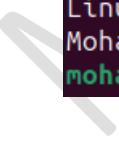
- When we have large content that got displayed on the terminal window it would be tough to scroll and read, to solve this we have **less** command which is one of the pager command. We can use arrow keys to traverse easily once below command is executed.



```
mohan@mohan:~/Desktop$ find / -maxdepth 4 -type f -size +100k -size -5M | less
mohan@mohan:~/Desktop$
```

- **Head** command is used to restrict the number of lines that get displayed on to the terminal. By default, if we don't give -n option it shows 10 lines.

```
cat editTex.txt | head  
cat editTex.txt | head -n 2  
head -n 2 editTex.txt
```



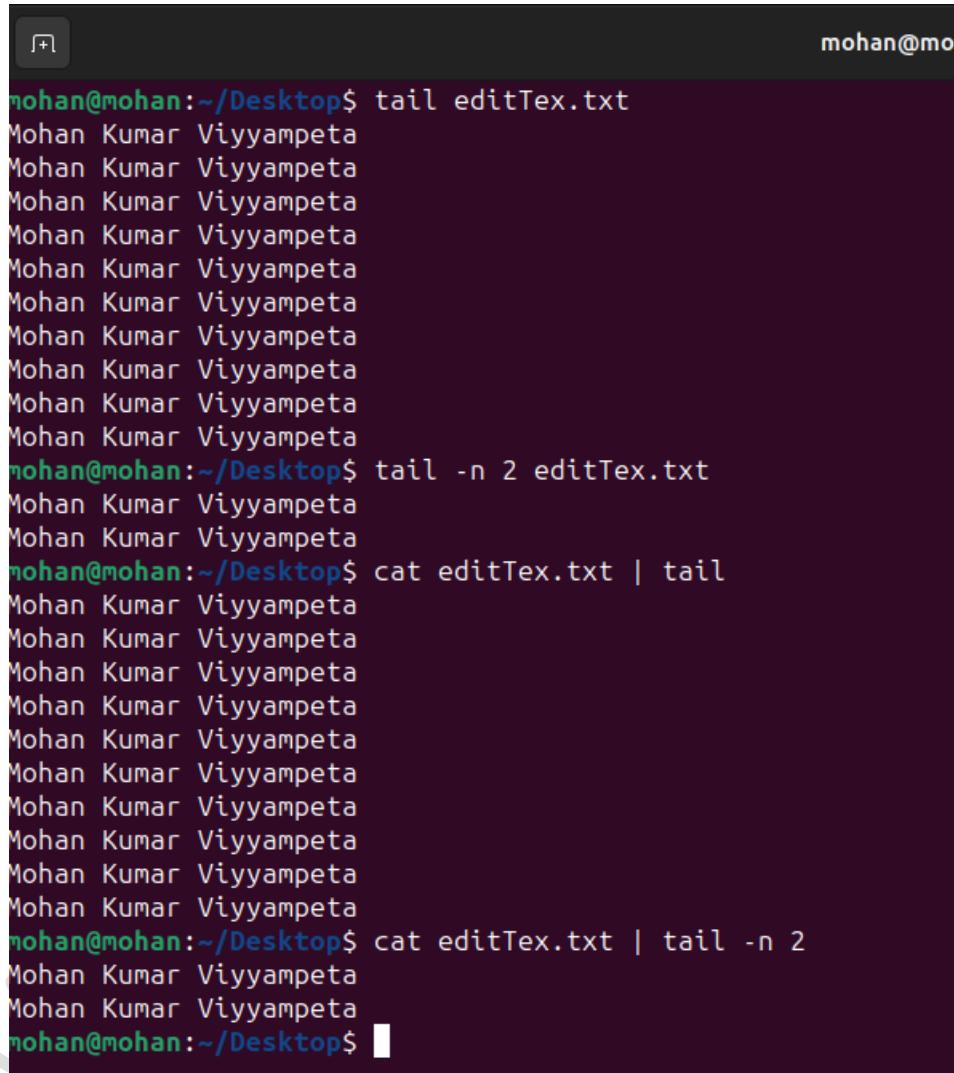
```
mohan@mohan:~/Desktop$ cat editTex.txt  
Mohan Kumar Viyyampeta  
Ram Kumar Viyyampeta  
Linux Tutorial  
testing echo read  
Ram Kumar Viyyampeta  
Linux Tutorial  
Linux Tutorial  
Linux Tutorial  
Linux Tutorial  
Mohan Kumar Viyyampeta  
mohan@mohan:~/Desktop$ cat editTex.txt | head  
Mohan Kumar Viyyampeta  
Ram Kumar Viyyampeta  
Linux Tutorial  
testing echo read  
Ram Kumar Viyyampeta  
Linux Tutorial  
Linux Tutorial  
Linux Tutorial  
Mohan Kumar Viyyampeta  
mohan@mohan:~/Desktop$
```

```
mohan@mohan:~/Desktop$ cat editTex.txt
Mohan Kumar Viyyampeta
Ram Kumar Viyyampeta
Linux Tutorial
testing echo read
Ram Kumar Viyyampeta
Linux Tutorial
Linux Tutorial
Linux Tutorial
Linux Tutorial
Mohan Kumar Viyyampeta
mohan@mohan:~/Desktop$ cat editTex.txt | head -n 2
Mohan Kumar Viyyampeta
Ram Kumar Viyyampeta
mohan@mohan:~/Desktop$
```

```
mohan@mohan:~/Desktop$ head editTex.txt
Mohan Kumar Viyyampeta
Ram Kumar Viyyampeta
Linux Tutorial
testing echo read
Ram Kumar Viyyampeta
Linux Tutorial
Linux Tutorial
Linux Tutorial
Linux Tutorial
Mohan Kumar Viyyampeta
mohan@mohan:~/Desktop$ head -n 2 editTex.txt
Mohan Kumar Viyyampeta
Ram Kumar Viyyampeta
mohan@mohan:~/Desktop$
```

- Tail command is quite opposite to Head command , this prints the number of lines from the bottom of the file

```
cat editText.txt | tail  
cat editText.txt | tail -n 2  
tail -n 2 editText.txt  
Tail editText.txt
```



mohan@mohan:~/Desktop\$ tail editText.txt

Mohan Kumar Viyyampeta

mohan@mohan:~/Desktop\$ tail -n 2 editText.txt

Mohan Kumar Viyyampeta

Mohan Kumar Viyyampeta

mohan@mohan:~/Desktop\$ cat editText.txt | tail

Mohan Kumar Viyyampeta

mohan@mohan:~/Desktop\$ cat editText.txt | tail -n 2

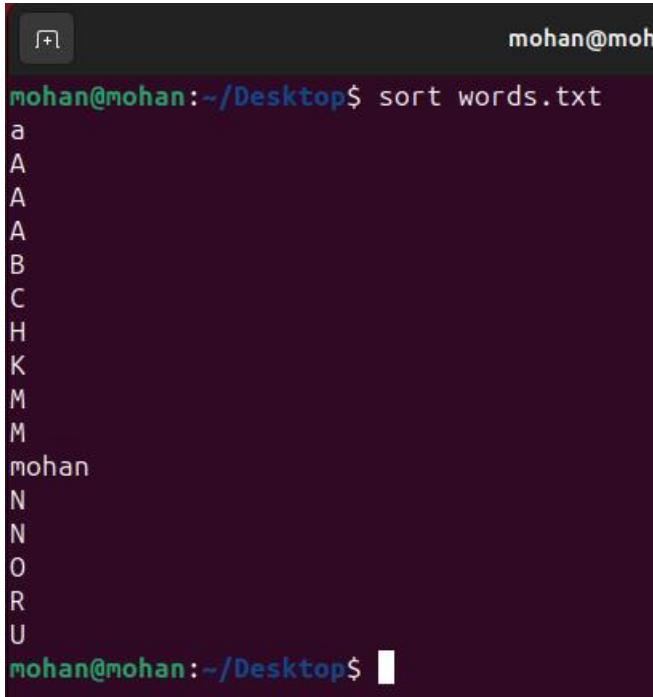
Mohan Kumar Viyyampeta

Mohan Kumar Viyyampeta

mohan@mohan:~/Desktop\$

Sorting Data

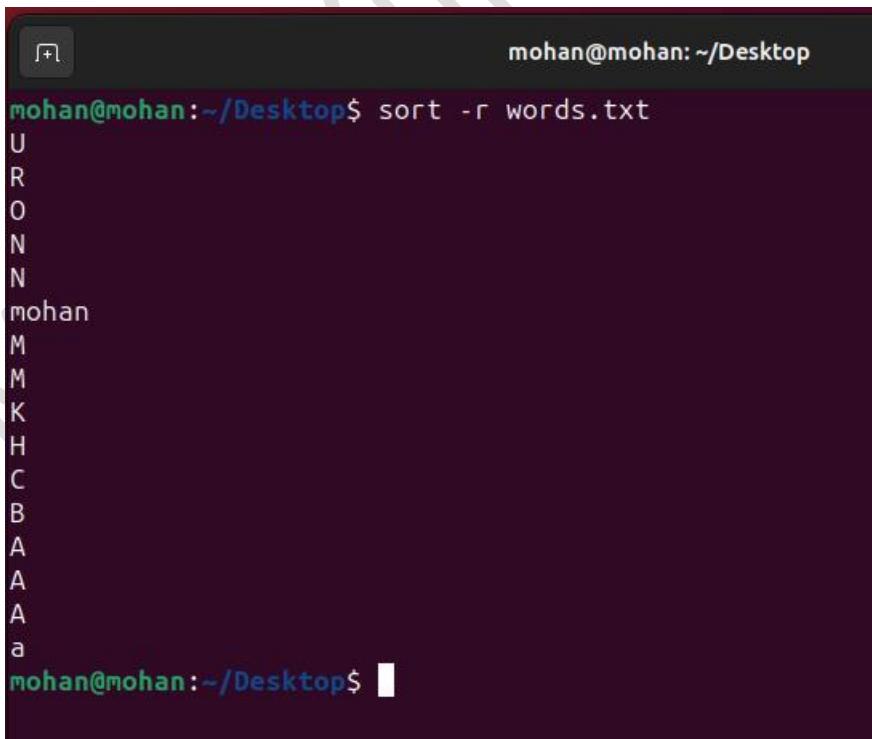
- To sort the contents of a file with words: sort <filepath>
sort word.txt



```
mohan@mohan:~/Desktop$ sort words.txt
a
A
A
A
B
C
H
K
M
M
mohan
N
N
O
R
U
mohan@mohan:~/Desktop$
```

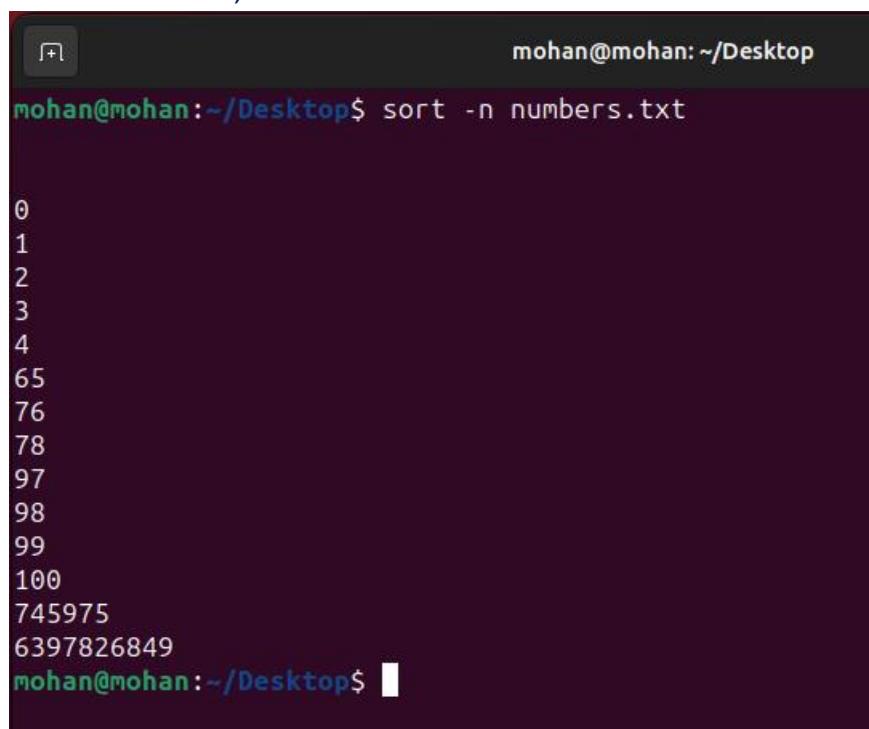
- The sort command tends to sort the smallest first i.e., (a-z, 0-9 etc). to sort the data in descending order we need to use -r option.

sort -r word.txt



```
mohan@mohan:~/Desktop$ sort -r words.txt
U
R
O
N
N
mohan
M
M
K
H
C
B
A
A
A
a
mohan@mohan:~/Desktop$
```

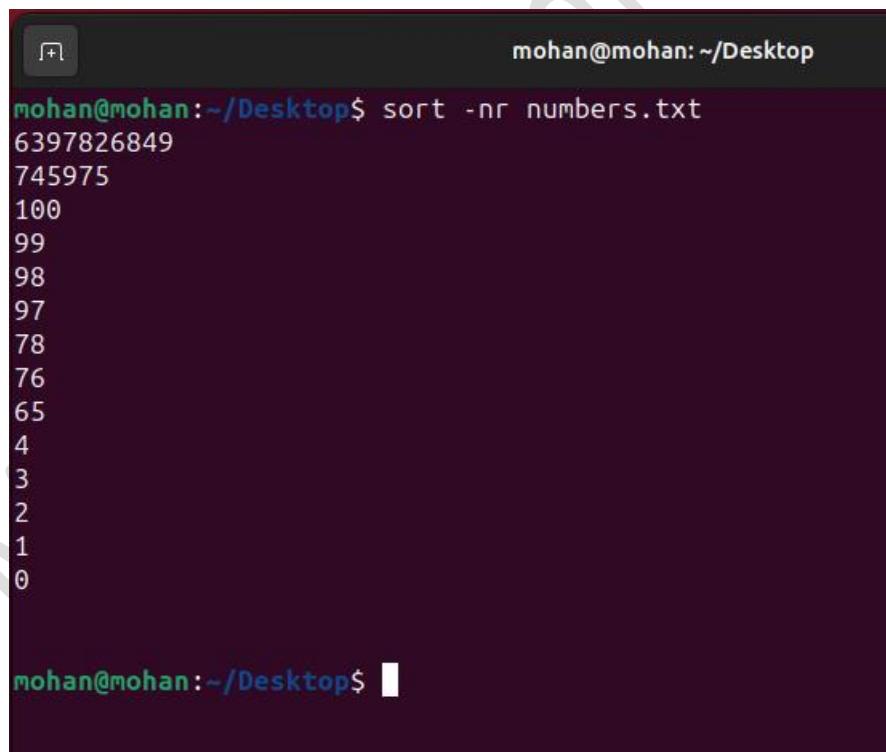
- To sort numerical, **sort -n numbers.txt**



```
mohan@mohan:~/Desktop$ sort -n numbers.txt

0
1
2
3
4
65
76
78
97
98
99
100
745975
6397826849
mohan@mohan:~/Desktop$
```

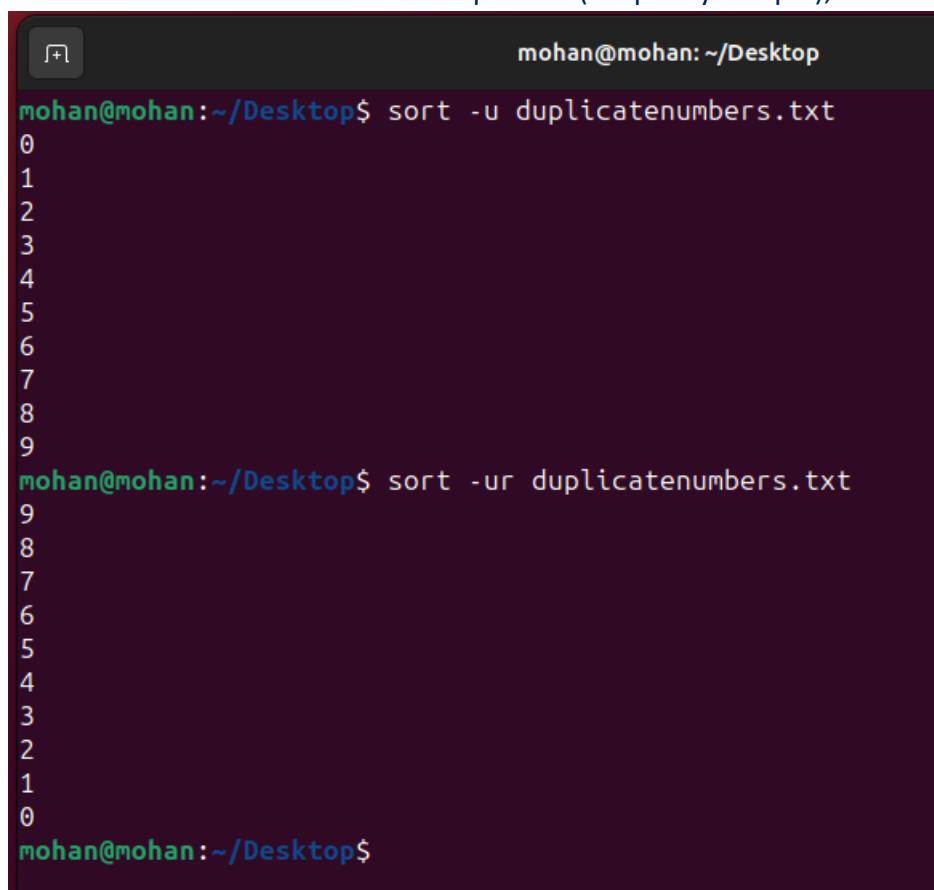
- To sort the numerical in reverse order. **sort -rn numbers.txt**



```
mohan@mohan:~/Desktop$ sort -nr numbers.txt
6397826849
745975
100
99
98
97
78
76
65
4
3
2
1
0

mohan@mohan:~/Desktop$
```

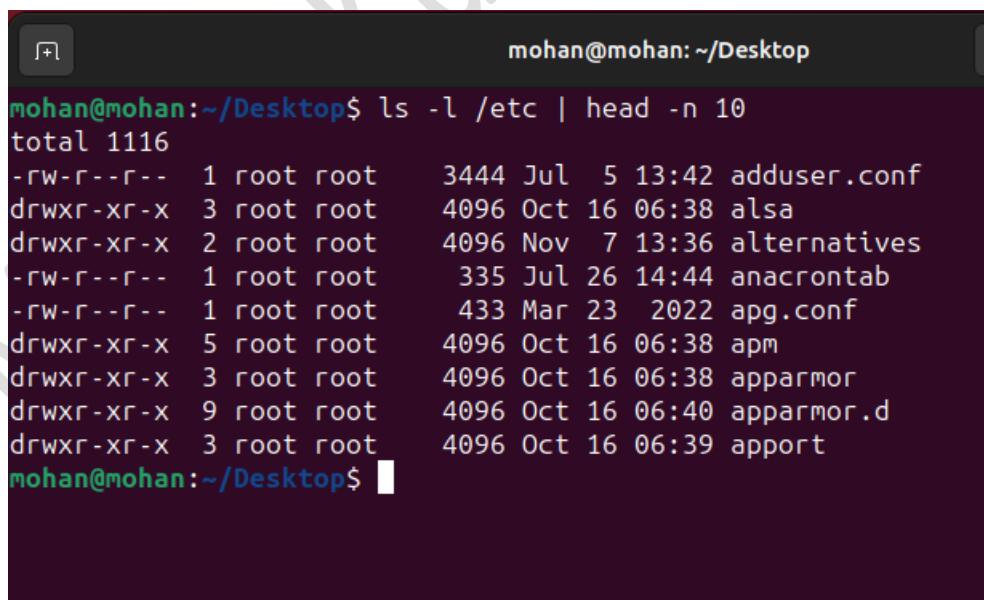
- To sort the numbers and remove duplicates (keep only Unique), **sort -u numbers.txt**



```
mohan@mohan:~/Desktop$ sort -u duplicatenumbers.txt
0
1
2
3
4
5
6
7
8
9
mohan@mohan:~/Desktop$ sort -ur duplicatenumbers.txt
9
8
7
6
5
4
3
2
1
0
mohan@mohan:~/Desktop$
```

A terminal window titled "mohan@mohan: ~/Desktop". It shows two commands being run. The first command, "sort -u duplicatenumbers.txt", outputs the numbers 0 through 9 each on a new line. The second command, "sort -ur duplicatenumbers.txt", outputs the same numbers in a random order. A watermark "Pulibanti" is diagonally across the terminal window.

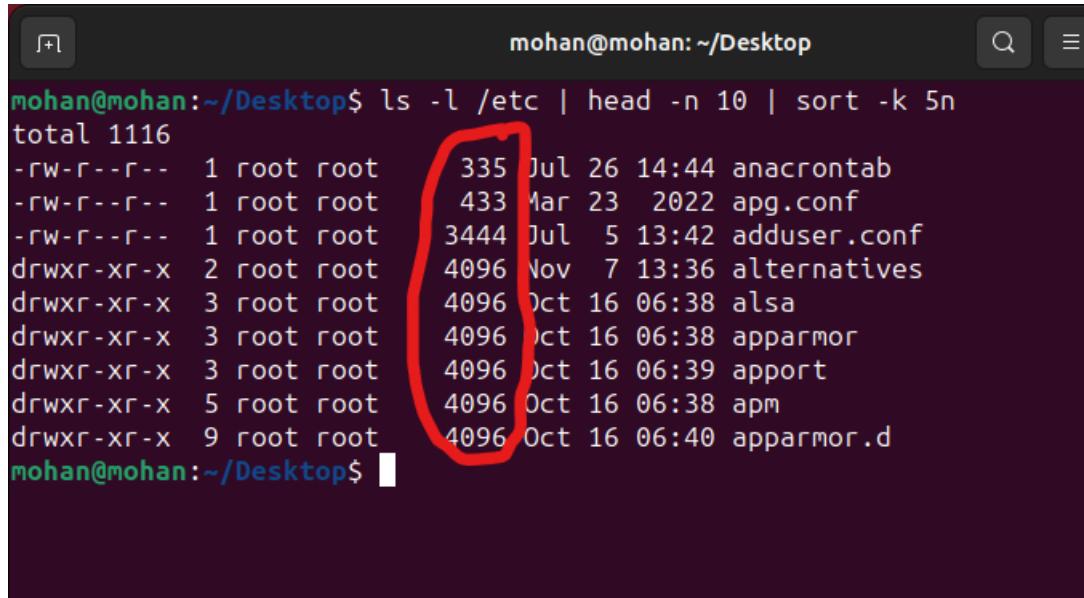
- Sort the below data based on columns.



```
mohan@mohan:~/Desktop$ ls -l /etc | head -n 10
total 1116
-rw-r--r-- 1 root root 3444 Jul  5 13:42 adduser.conf
drwxr-xr-x 3 root root 4096 Oct 16 06:38 alsa
drwxr-xr-x 2 root root 4096 Nov  7 13:36 alternatives
-rw-r--r-- 1 root root 335 Jul 26 14:44 anacrontab
-rw-r--r-- 1 root root 433 Mar 23 2022 apg.conf
drwxr-xr-x 5 root root 4096 Oct 16 06:38 apt
drwxr-xr-x 3 root root 4096 Oct 16 06:38 apparmor
drwxr-xr-x 9 root root 4096 Oct 16 06:40 apparmor.d
drwxr-xr-x 3 root root 4096 Oct 16 06:39 apport
mohan@mohan:~/Desktop$
```

A terminal window titled "mohan@mohan: ~/Desktop". It shows the output of the command "ls -l /etc | head -n 10", which lists files and directories in the /etc directory with their permissions, ownership, and timestamps. A watermark "Pulibanti" is diagonally across the terminal window.

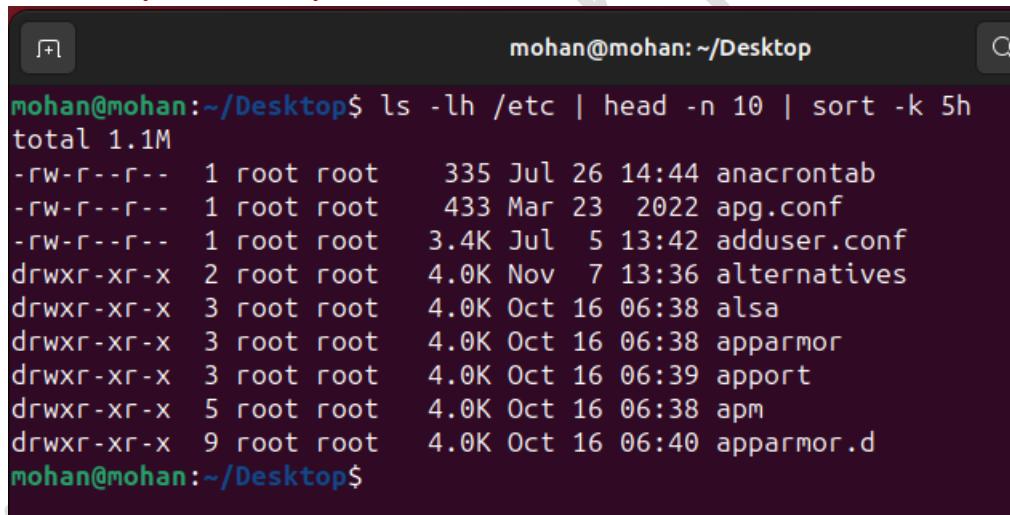
ls -l /etc | head -n 10 | sort -k 5n → -k indicates column option for which we need to pass KEYDEF, 5 indicates column number and n indicates numerical that we want to sort



```
mohan@mohan:~/Desktop$ ls -l /etc | head -n 10 | sort -k 5n
total 1116
-rw-r--r-- 1 root root 335 Jul 26 14:44 anacrontab
-rw-r--r-- 1 root root 433 Mar 23 2022 apg.conf
-rw-r--r-- 1 root root 3444 Jul 5 13:42 adduser.conf
drwxr-xr-x 2 root root 4096 Nov 7 13:36 alternatives
drwxr-xr-x 3 root root 4096 Oct 16 06:38 alsa
drwxr-xr-x 3 root root 4096 Oct 16 06:38 apparmor
drwxr-xr-x 3 root root 4096 Oct 16 06:39 apport
drwxr-xr-x 5 root root 4096 Oct 16 06:38 apm
drwxr-xr-x 9 root root 4096 Oct 16 06:40 apparmor.d
mohan@mohan:~/Desktop$
```

- To sort the data in human readable format

ls -lh /etc | head -n 10 | sort -k 5h



```
mohan@mohan:~/Desktop$ ls -lh /etc | head -n 10 | sort -k 5h
total 1.1M
-rw-r--r-- 1 root root 335 Jul 26 14:44 anacrontab
-rw-r--r-- 1 root root 433 Mar 23 2022 apg.conf
-rw-r--r-- 1 root root 3.4K Jul 5 13:42 adduser.conf
drwxr-xr-x 2 root root 4.0K Nov 7 13:36 alternatives
drwxr-xr-x 3 root root 4.0K Oct 16 06:38 alsa
drwxr-xr-x 3 root root 4.0K Oct 16 06:38 apparmor
drwxr-xr-x 3 root root 4.0K Oct 16 06:39 apport
drwxr-xr-x 5 root root 4.0K Oct 16 06:38 apm
drwxr-xr-x 9 root root 4.0K Oct 16 06:40 apparmor.d
mohan@mohan:~/Desktop$
```

- To sort the data based on month in 6th column

ls -lh /etc | head -n 10 | sort -k 6M

```
mohan@mohan:~/Desktop$ ls -lh /etc | head -n 10 | sort -k 6M
total 1.1M
-rw-r--r-- 1 root root    433 Mar 23 2022 apg.conf
-rw-r--r-- 1 root root   335 Jul 26 14:44 anacrontab
-rw-r--r-- 1 root root   3.4K Jul  5 13:42 adduser.conf
drwxr-xr-x 3 root root  4.0K Oct 16 06:38 alsa
drwxr-xr-x 3 root root  4.0K Oct 16 06:38 apparmor
drwxr-xr-x 3 root root  4.0K Oct 16 06:39 apport
drwxr-xr-x 5 root root  4.0K Oct 16 06:38 apt
drwxr-xr-x 9 root root  4.0K Oct 16 06:40 apparmor.d
drwxr-xr-x 2 root root  4.0K Nov  7 13:36 alternatives
mohan@mohan:~/Desktop$
```

File Archiving and Compression

- Archiving files is basically a two-step process
 - 1) Make tarball – tarball is the place where we place all the files that we want to compress (tarball is like a bag where we can place all files, like groceries in a bag)
 - 2) Compress tarball
- Make a tarball → **tar -cvf <tarfile_Name> <list_of_files_to_into_tarball>**
tar -cvf ourAchieve.tar files[1-3].txt → c for create, v- verbose that prints status of the operation with some message, f stands for we want to archive files, tarfile_name is the name of the tar file and the extension of this file would be .tar, and then the list of files we want to archive.

```
mohan@mohan:~/Desktop$ tar -cvf ourarchive.tar *.txt
duplicatenumbers.txt
numbers.txt
words.txt
mohan@mohan:~/Desktop$
```

- Tarball file will be in red color when we do ls

```
mohan@mohan:~/Desktop$ ls
duplicateNumbers.txt numbers.txt ourarchieve.tar words.txt
mohan@mohan:~/Desktop$
```

- If we check the size of tarball created, it would be more than the sum of sizes of all the files. As tarballs themselves will have some size.

```
mohan@mohan:~/Desktop$ ls -lh
total 24K
-rw-rw-r-- 1 mohan mohan 40 Nov 14 15:30 duplicateNumbers.txt
-rw-rw-r-- 1 mohan mohan 52 Nov 14 15:29 numbers.txt
-rw-rw-r-- 1 mohan mohan 10K Nov 14 16:50 ourarchieve.tar
-rw-rw-r-- 1 mohan mohan 36 Nov 14 15:29 words.txt
mohan@mohan:~/Desktop$ 10k > (40+52+36)K
```

- If we want to check what files are there in tarball

tar -tf <tarfilename> → option t stands for test and f for files
tar -tf ourarchieve.tar

```
mohan@mohan:~/Desktop$ tar -tf ourarchieve.tar
duplicateNumbers.txt
numbers.txt
words.txt
mohan@mohan:~/Desktop$
```

- To extract the files outside from tarball below is the command

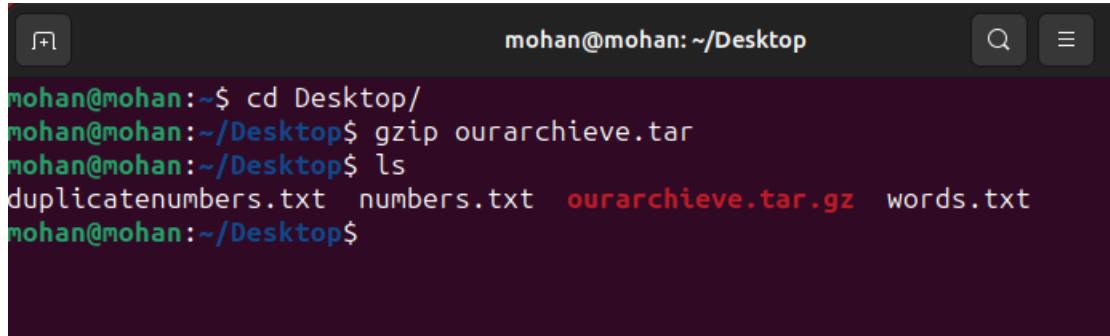
tar -xvf <tarballname> → option x stands for extract, v for verbose and f for files.

```
mohan@mohan:~/Desktop$ ls
ourarchieve.tar
mohan@mohan:~/Desktop$ tar -xvf ourarchieve.tar
duplicateNumbers.txt
numbers.txt
words.txt
mohan@mohan:~/Desktop$ ls
duplicateNumbers.txt numbers.txt ourarchieve.tar words.txt
mohan@mohan:~/Desktop$
```

Even when we extract files from tarball, we will still have those files in tarball.

Compression

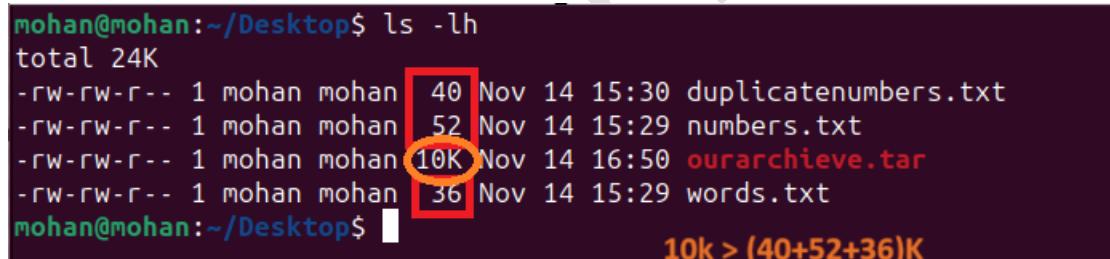
- Two compression algorithms are used in the Linux world. gzip & bzip2
- gzip is faster but has less compression power, bzip2 on the other hand can generally compress files to a smaller size than gzip but does require more computational time.
- Command to compress using gzip
gzip <tarfileName> → compress the data and the extension of compressed files would be .gz



```
mohan@mohan:~$ cd Desktop/
mohan@mohan:~/Desktop$ gzip ourarchieve.tar
mohan@mohan:~/Desktop$ ls
duplicateNumbers.txt  numbers.txt  ourarchieve.tar.gz  words.txt
mohan@mohan:~/Desktop$
```

- When tar file is compressed, size reduces to very much low. (try compressing large file to see the size less than sum of the compressed file sizes)

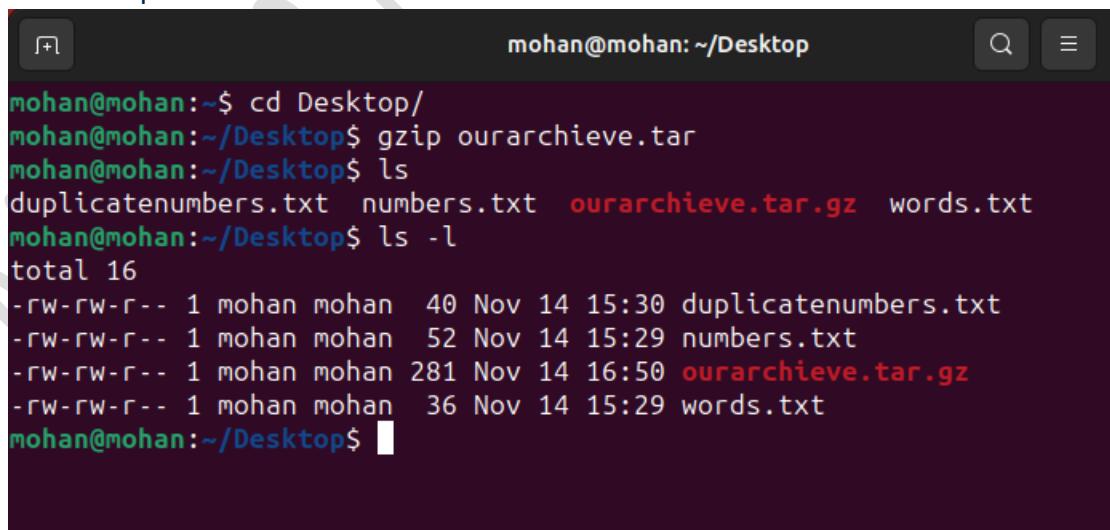
Before Compression:



```
mohan@mohan:~/Desktop$ ls -lh
total 24K
-rw-rw-r-- 1 mohan mohan 40 Nov 14 15:30 duplicateNumbers.txt
-rw-rw-r-- 1 mohan mohan 52 Nov 14 15:29 numbers.txt
-rw-rw-r-- 1 mohan mohan 10K Nov 14 16:50 ourarchieve.tar
-rw-rw-r-- 1 mohan mohan 36 Nov 14 15:29 words.txt
mohan@mohan:~/Desktop$
```

10k > (40+52+36)K

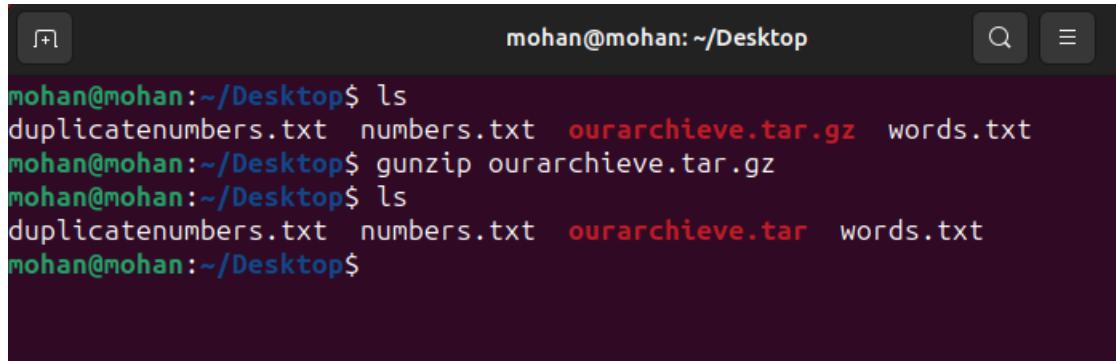
After Compression:



```
mohan@mohan:~$ cd Desktop/
mohan@mohan:~/Desktop$ gzip ourarchieve.tar
mohan@mohan:~/Desktop$ ls
duplicateNumbers.txt  numbers.txt  ourarchieve.tar.gz  words.txt
mohan@mohan:~/Desktop$ ls -l
total 16
-rw-rw-r-- 1 mohan mohan 40 Nov 14 15:30 duplicateNumbers.txt
-rw-rw-r-- 1 mohan mohan 52 Nov 14 15:29 numbers.txt
-rw-rw-r-- 1 mohan mohan 281 Nov 14 16:50 ourarchieve.tar.gz
-rw-rw-r-- 1 mohan mohan 36 Nov 14 15:29 words.txt
mohan@mohan:~/Desktop$
```

- Command to unzip compressed file.

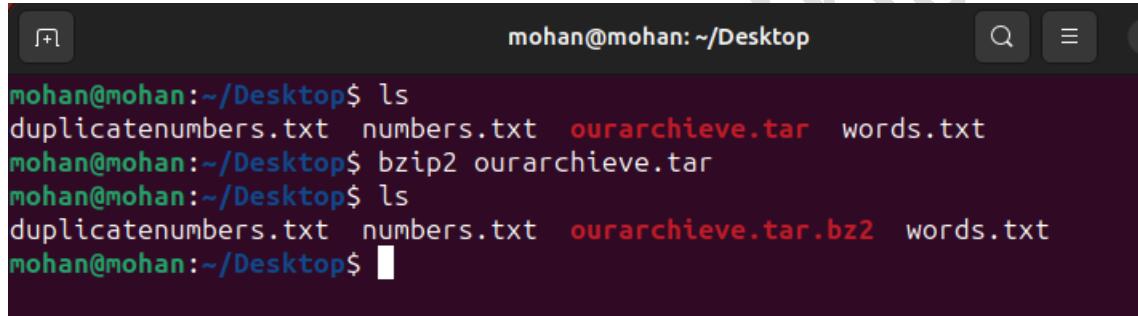
gunzip <compressed gip file path>



```
mohan@mohan:~/Desktop$ ls
duplicateNumbers.txt numbers.txt ourarchive.tar.gz words.txt
mohan@mohan:~/Desktop$ gunzip ourarchive.tar.gz
mohan@mohan:~/Desktop$ ls
duplicateNumbers.txt numbers.txt ourarchive.tar words.txt
mohan@mohan:~/Desktop$
```

- Compression using bzip2 algorithm.

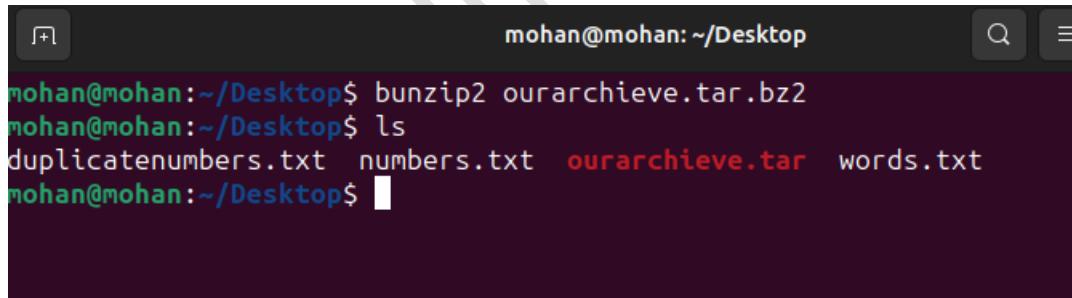
bzip2 <tarfilepath>



```
mohan@mohan:~/Desktop$ ls
duplicateNumbers.txt numbers.txt ourarchive.tar words.txt
mohan@mohan:~/Desktop$ bzip2 ourarchive.tar
mohan@mohan:~/Desktop$ ls
duplicateNumbers.txt numbers.txt ourarchive.tar.bz2 words.txt
mohan@mohan:~/Desktop$
```

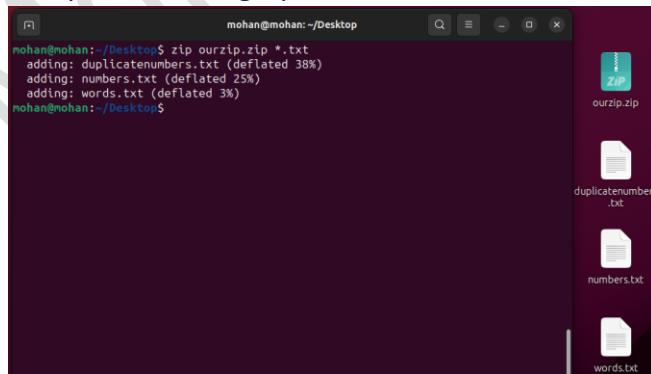
- Command to unzip bzip2 file

bunzip <tarfilepath>



```
mohan@mohan:~/Desktop$ bunzip2 ourarchive.tar.bz2
mohan@mohan:~/Desktop$ ls
duplicateNumbers.txt numbers.txt ourarchive.tar words.txt
mohan@mohan:~/Desktop$
```

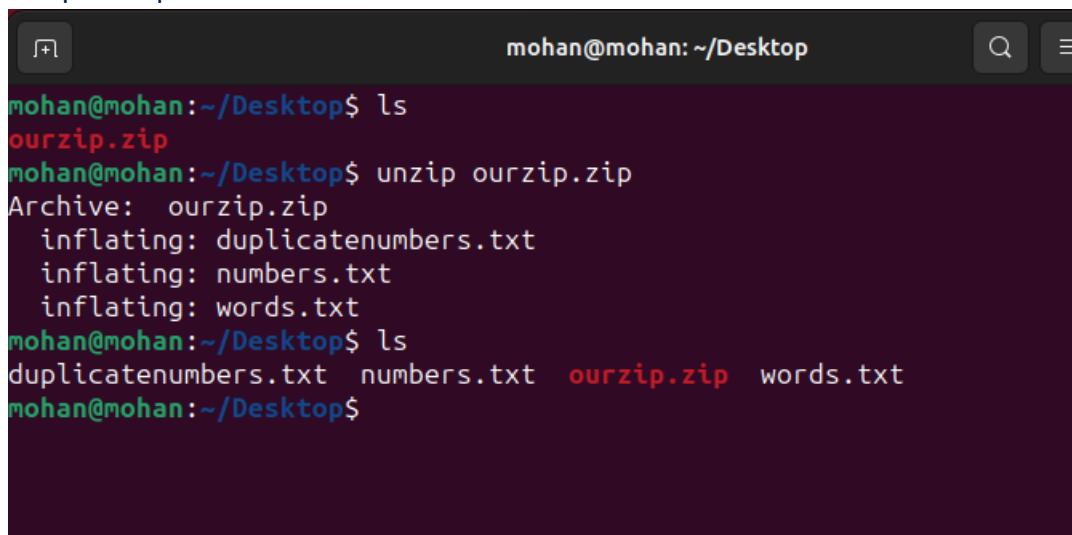
- Compression using zip



mohan@mohan:~/Desktop\$ zip ourzip.zip *.txt
adding: duplicateNumbers.txt (deflated 38%)
adding: numbers.txt (deflated 25%)
adding: words.txt (deflated 3%)
mohan@mohan: ~/Desktop\$

The file browser shows the contents of the 'ourzip.zip' file, which includes four files: 'duplicateNumbers.txt', 'numbers.txt', and 'words.txt'. Each file is represented by a small icon and its name.

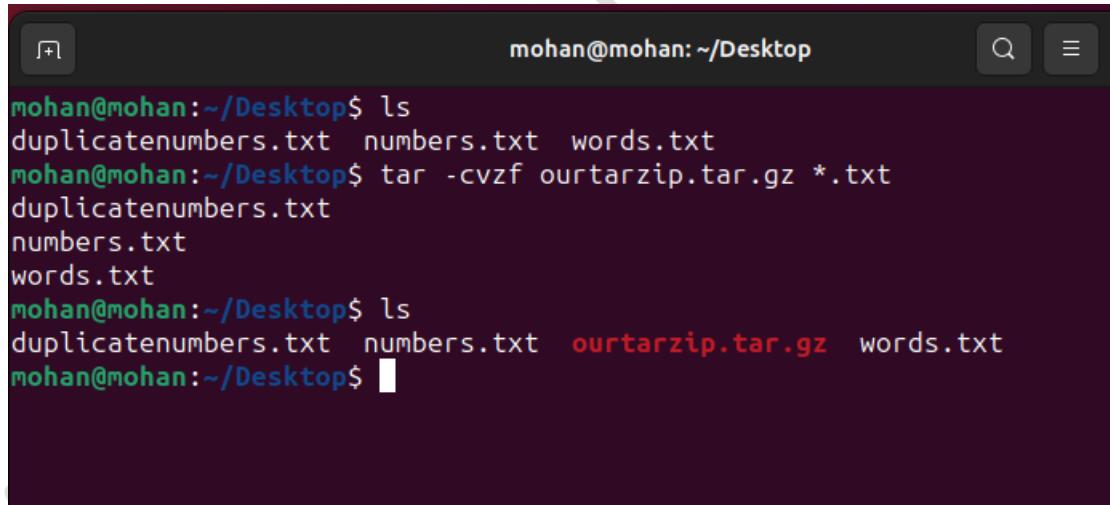
- unzip the zip file



```
mohan@mohan:~/Desktop$ ls  
ourzip.zip  
mohan@mohan:~/Desktop$ unzip ourzip.zip  
Archive: ourzip.zip  
  inflating: duplicatenumbers.txt  
  inflating: numbers.txt  
  inflating: words.txt  
mohan@mohan:~/Desktop$ ls  
duplicatenumbers.txt  numbers.txt  ourzip.zip  words.txt  
mohan@mohan:~/Desktop$
```

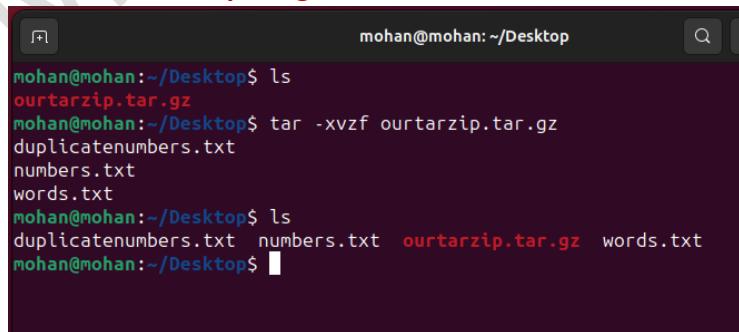
- till now compression in two steps, 1) Creating tarball 2) compressing the tarball to do both the steps with a single command, below is the command
Using gzip compression and Extraction:

tar -cvzf outtarfile.tar.gz *.txt → compression



```
mohan@mohan:~/Desktop$ ls  
duplicatenumbers.txt  numbers.txt  words.txt  
mohan@mohan:~/Desktop$ tar -cvzf ourtarzip.tar.gz *.txt  
duplicatenumbers.txt  
numbers.txt  
words.txt  
mohan@mohan:~/Desktop$ ls  
duplicatenumbers.txt  numbers.txt  ourtarzip.tar.gz  words.txt  
mohan@mohan:~/Desktop$
```

tar -xvzf ourtarzip.tar.gz → Extraction



```
mohan@mohan:~/Desktop$ ls  
ourtarzip.tar.gz  
mohan@mohan:~/Desktop$ tar -xvzf ourtarzip.tar.gz  
duplicatenumbers.txt  
numbers.txt  
words.txt  
mohan@mohan:~/Desktop$ ls  
duplicatenumbers.txt  numbers.txt  ourtarzip.tar.gz  words.txt  
mohan@mohan:~/Desktop$
```

Compression and Extraction using bzip2:

`tar -cvjf outbzfile.tar.bz2 *.txt` → compression

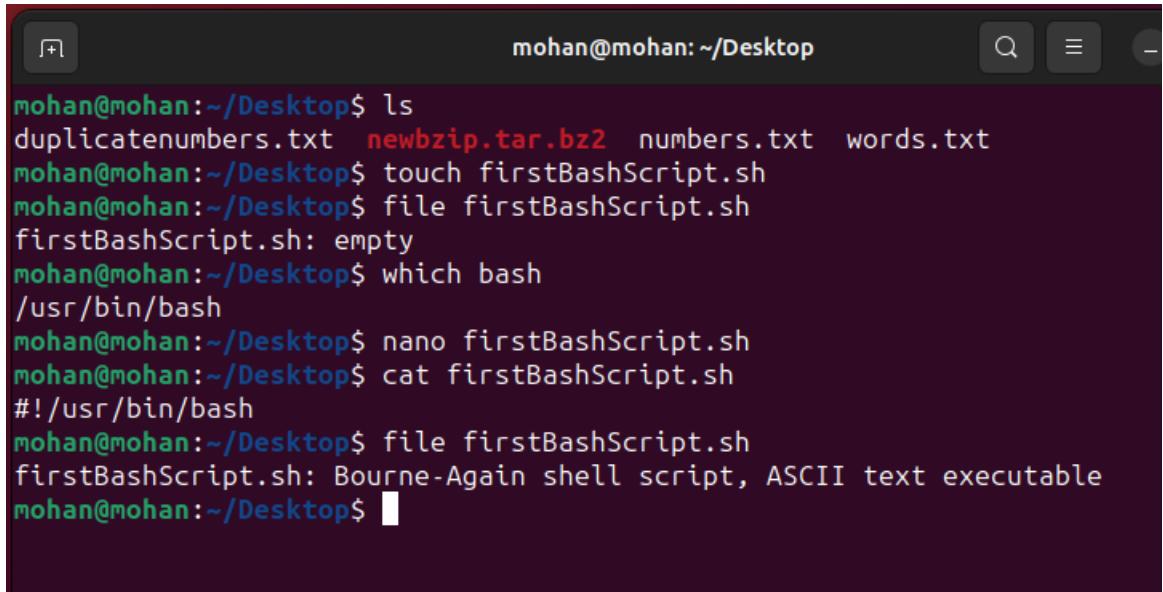
```
mohan@mohan:~/Desktop$ tar -cvjf newbzip.tar.bz2 *.txt
duplicatenumbers.txt
numbers.txt
words.txt
mohan@mohan:~/Desktop$ ls
duplicatenumbers.txt  newbzip.tar.bz2  numbers.txt  words.txt
mohan@mohan:~/Desktop$
```

`tar -xvf outbzfile.tar.bz2` → Extraction

```
mohan@mohan:~/Desktop$ ls
newbzip.tar.bz2
mohan@mohan:~/Desktop$ tar -xvf newbzip.tar.bz2
duplicatenumbers.txt
numbers.txt
words.txt
mohan@mohan:~/Desktop$ ls
duplicatenumbers.txt  newbzip.tar.bz2  numbers.txt  words.txt
mohan@mohan:~/Desktop$
```

Creating Bash Scripts

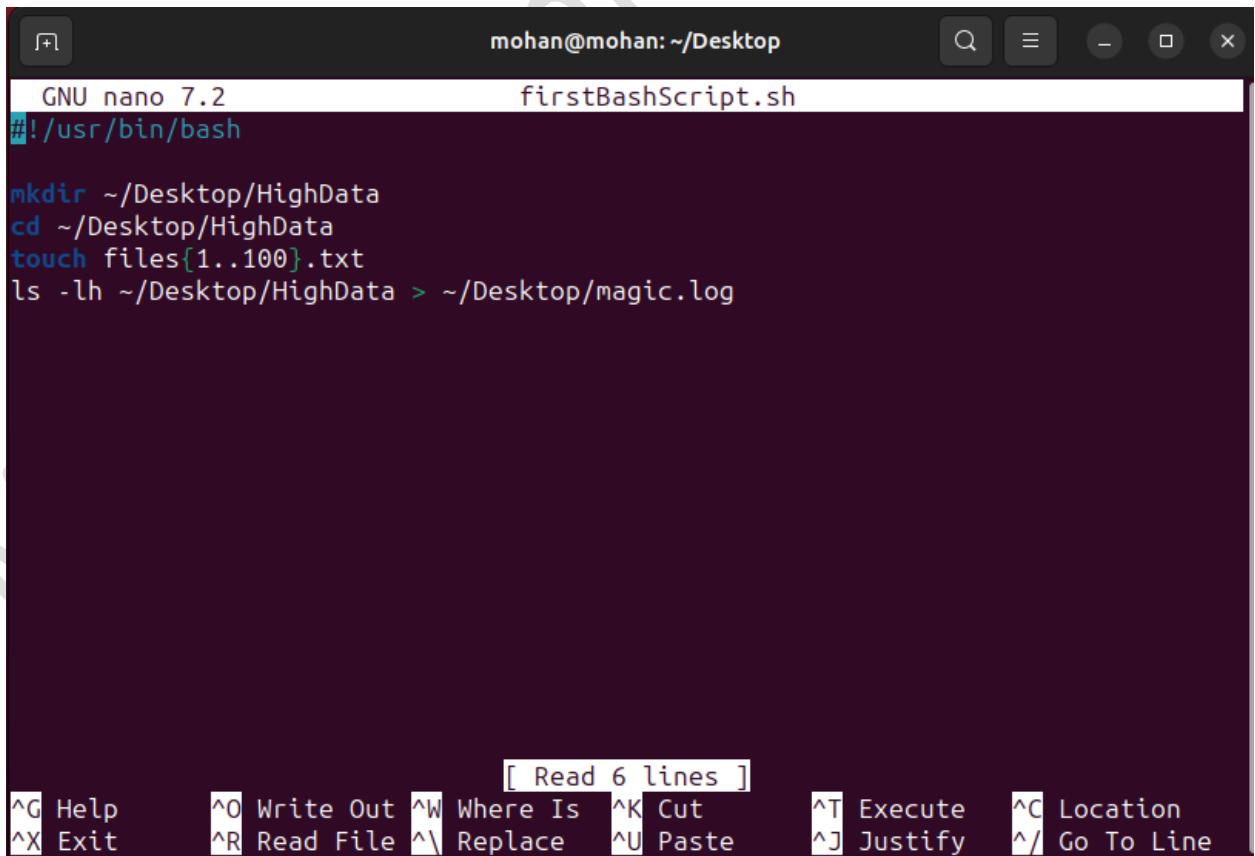
- Bash stands for Bourne-Again shell Script
- create a file with extension '.sh' , for example `our_script.sh` `touch 'our_script.sh'`
- in the terminal type '`which bash`' and take that path and use that path as first line in the `.sh` file to get that file recognized as bash script file else this will be recognized as an empty file.
- `#!/usr/bin/bash (#!{output of which bash})` → this says it's not a normal file, interpret this using bash shell interpreter, this should be first line in all bash scripts



```
mohan@mohan:~/Desktop$ ls
duplicateNumbers.txt  newbzip.tar.bz2  numbers.txt  words.txt
mohan@mohan:~/Desktop$ touch firstBashScript.sh
mohan@mohan:~/Desktop$ file firstBashScript.sh
firstBashScript.sh: empty
mohan@mohan:~/Desktop$ which bash
/usr/bin/bash
mohan@mohan:~/Desktop$ nano firstBashScript.sh
mohan@mohan:~/Desktop$ cat firstBashScript.sh
#!/usr/bin/bash
mohan@mohan:~/Desktop$ file firstBashScript.sh
firstBashScript.sh: Bourne-Again shell script, ASCII text executable
mohan@mohan:~/Desktop$
```

Now the .sh file created is treated as Bourne-Again shell script (bash script)

- Bash scripts are generally used to execute multiple command-line commands at one go.
To do that, add all the commands that we want to execute In an order in the .sh file.
As below.



```
GNU nano 7.2           firstBashScript.sh
#!/usr/bin/bash

mkdir ~/Desktop/HighData
cd ~/Desktop/HighData
touch files{1..100}.txt
ls -lh ~/Desktop/HighData > ~/Desktop/magic.log
```

[Read 6 lines]

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line

- To execute the bash script, the command is **bash <.sh filepath>**

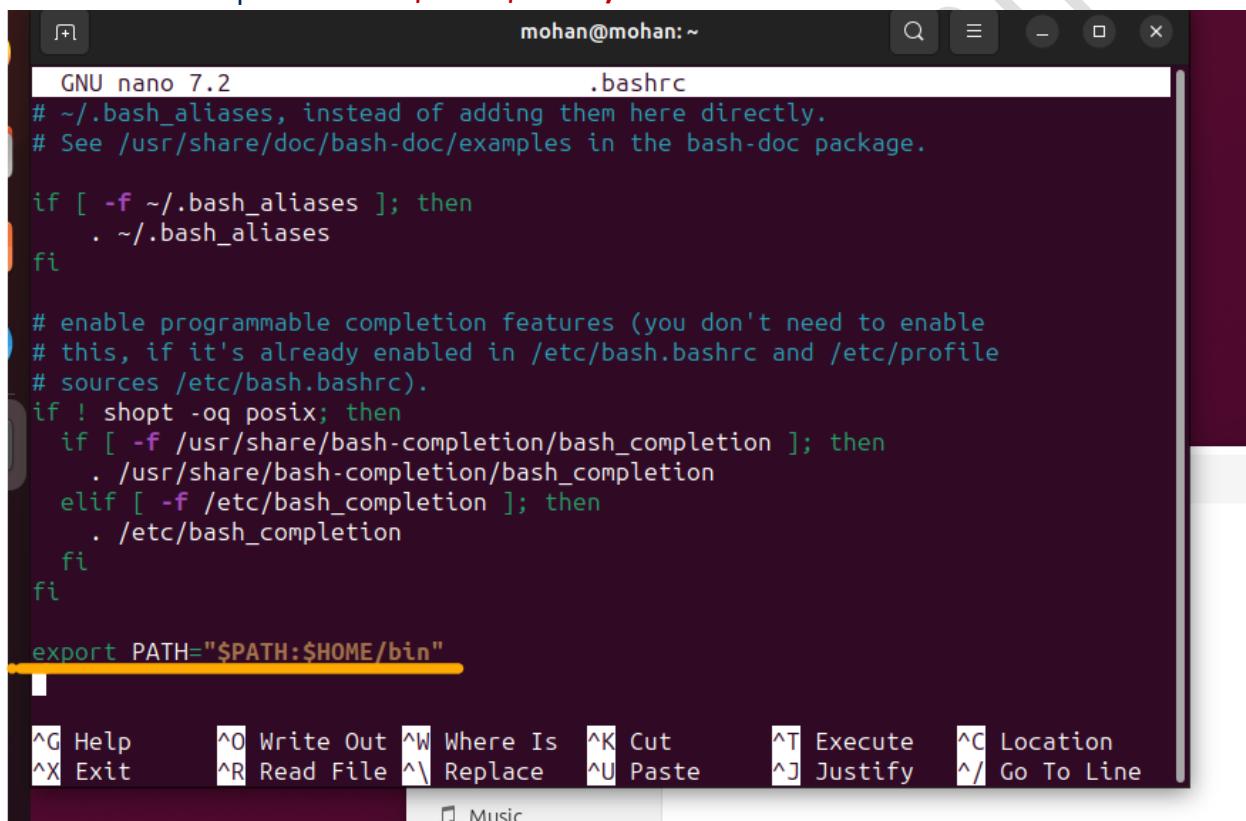
```
mohan@mohan:~/Desktop$ bash firstBashScript.sh
mohan@mohan:~/Desktop$
```

- To add the bash scripts to the shells search path, we have to follow below steps

- 1) Move all the bash scripts to ~bin folder
- 2) on the home directory, edit '.bashrc' file

nano .bashrc

- 3) Add ~bin folder to the path parameter by adding the below line to the .bashrc file → **export PATH = "\$PATH:\$HOME/bin"**



```
mohan@mohan:~/.bashrc
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

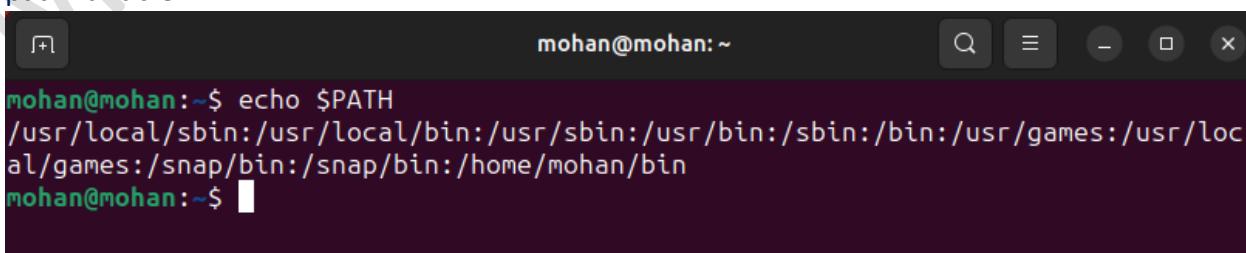
if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi

export PATH="$PATH:$HOME/bin"
```

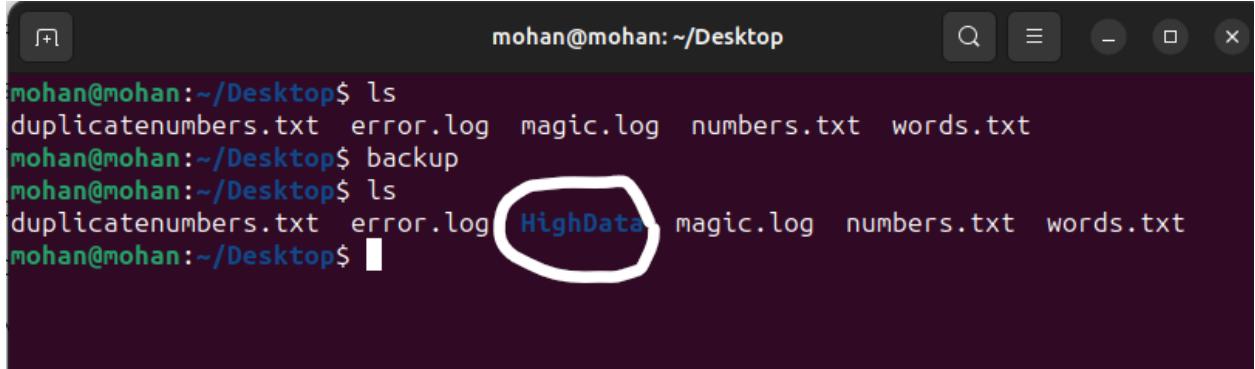
The terminal window shows the .bashrc file being edited in nano. The line `export PATH="$PATH:$HOME/bin"` is highlighted with a yellow bar. The bottom of the window shows the nano key bindings.

- Open new terminal (or) execute source `~/.bashrc` in the same terminal
- Execute “**echo \$PATH**” command and you could see home/mohan/bin path added to the path variable.



```
mohan@mohan:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/snap/bin:/home/mohan/bin
mohan@mohan:~$
```

- Once this is added, shell knows our bash scripts path, so we can directly execute the scripts from any place. i.e., I placed all my .sh files in ~/bin folder but when I execute **backup** shell script from home directory still we can see the results.



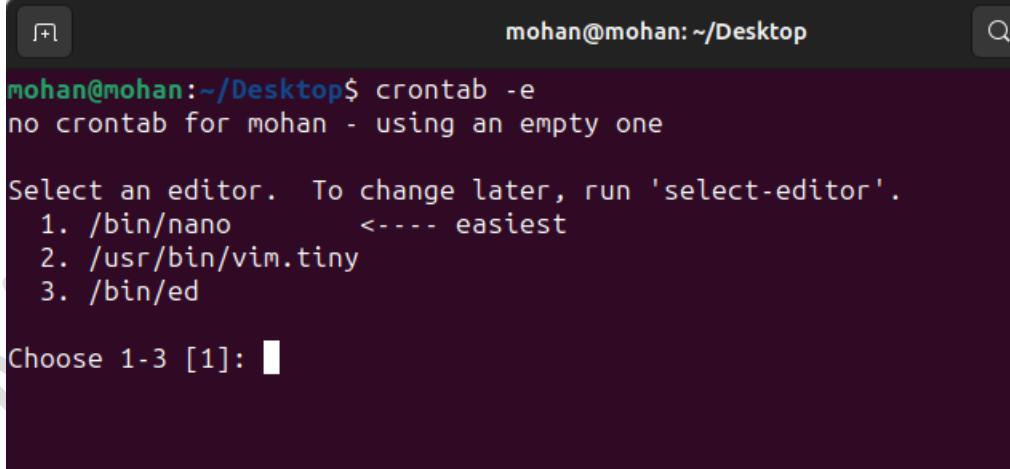
```
mohan@mohan:~/Desktop$ ls
duplicatenumbers.txt  error.log  magic.log  numbers.txt  words.txt
mohan@mohan:~/Desktop$ backup
mohan@mohan:~/Desktop$ ls
duplicatenumbers.txt  error.log  HighData  magic.log  numbers.txt  words.txt
mohan@mohan:~/Desktop$
```

- Aliases is just when we have a single command whereas bash scripts is when we have multiple line scripts.

Scheduling CRON JOBS

- Cron jobs is to run bash_scripts on schedule.
- Cron is a command line based program that is used to schedule tasks
- Cron gets its name from Greek word Kronos which means time.
- Each user will have a CronTab which is just a text file and each Crontab lists which commands or scripts will be automatically run by the user
- To open the Crontab, command is

crontab -e



```
mohan@mohan:~/Desktop$ crontab -e
no crontab for mohan - using an empty one

Select an editor. To change later, run 'select-editor'.
 1. /bin/nano      <---- easiest
 2. /usr/bin/vim.tiny
 3. /bin/ed

Choose 1-3 [1]:
```

When we open crontab for the first time, it asks for us to choose the editor.

```
GNU nano 7.2 /tmp/crontab.rB48Hm/crontab
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
```

[line 7/24 (29%), col 1/57 (1%), char 220/889]

^G Help	^O Write Out	^W Where Is	^K Cut	^T Execute
^X Exit	^R Read File	^V Replace	^U Paste	^J Justify

- We need to add one row for each job that we want to schedule. And the format of the row is as below

m h dom mon dow command

m → minutes of the hour (goes from 0 to 59), * means run any minute

h → hours of the day (goes from 0 to 23), * means run any hour of day

dom → date of the month (goes from 1 to 30/31)

mon → month (Range 1-12(1 being Jan) or JAN, FEB,MAR....DEC(three letter words))

dow → day of week (Range 0 to 6, where 0 being Sunday, 6 bring Saturday , we can also pass MON, TUE, WED.....SAT)

eg: 20 11 10 JUN SUN echo "Mohan Kumar" → job triggers on 10th June at 11:20 being 10th June as Sunday.



```
mohan@mohan:~
```

```
GNU nano 7.2          /tmp/crontab.YOHuVx/crontab
```

```
#  
# To define the time you can provide concrete values for  
# minute (m), hour (h), day of month (dom), month (mon),  
# and day of week (dow) or use '*' in these fields (for 'any').  
#  
# Notice that tasks will be started based on the cron's system  
# daemon's notion of time and timezones.  
#  
# Output of the crontab jobs (including errors) is sent through  
# email to the user the crontab file belongs to (unless redirected).  
#  
# For example, you can run a backup of all your user accounts  
# at 5 a.m every week with:  
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/  
#  
# For more information see the manual pages of crontab(5) and cron(8)  
#  
# m h dom mon dow   command  
* * * * * echo "Mohan Kumar Pulibanti" >> ~/Desktop/name.txt
```

```
^G Help      ^O Write Out  ^W Where Is  ^K Cut      ^T Execute  ^C Location  
^X Exit      ^R Read File  ^\ Replace   ^U Paste    ^J Justify  ^/ Go To Line
```

```
mohan@mohan:~
```

```
mohan@mohan:~$ crontab -e  
no crontab for mohan - using an empty one  
crontab: installing new crontab  
mohan@mohan:~$ crontab -e  
No modification made  
mohan@mohan:~$
```

mohan@mohan:~\$ crontab -e
no crontab for mohan - using an empty one
crontab: installing new crontab
mohan@mohan:~\$ crontab -e
No modification made
mohan@mohan:~\$

The terminal shows the user has run the command to edit the crontab, which creates a new empty crontab. Then they attempt to edit it again but make no changes. Below the terminal is a file viewer window titled "name.txt" located at "/Desktop". It contains three lines of text: "Mohan Kumar Pulibanti", each followed by a series of red asterisks. The file viewer interface includes tabs for ".bash_aliases", "duplicatenumbers.txt", and "magic.log", and a status bar showing "name.txt".

- If we want to change the editor for the crontab, we have to change in .selected_editor file in the home directory

```
mohan@mohan:~$ ls -a  
. .bashrc .gnupg .selected_editor  
.. bin .lessht snap  
.aspell.en.prepl .cache .local .ssh  
.aspell.en.pws .config Music .sudo_as_admin_successful  
.bash_aliases Desktop Pictures Templates  
.bash_history Documents .profile Videos  
.bash_logout Downloads Public  
mohan@mohan:~$ cat .selected_editor  
# Generated by /usr/bin/select-editor  
SELECTED_EDITOR="/bin/nano"  
mohan@mohan:~$
```

We can also change the editor by giving the command select-editor.

```
mohan@mohan:~$ select-editor  
Select an editor. To change later, run 'select-editor'.  
1. /bin/nano <---- easiest  
2. /usr/bin/vim.tiny  
3. /bin/ed  
Choose 1-3 [1]:
```

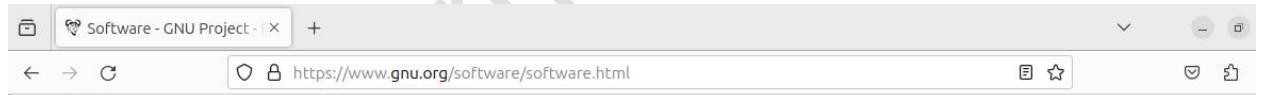
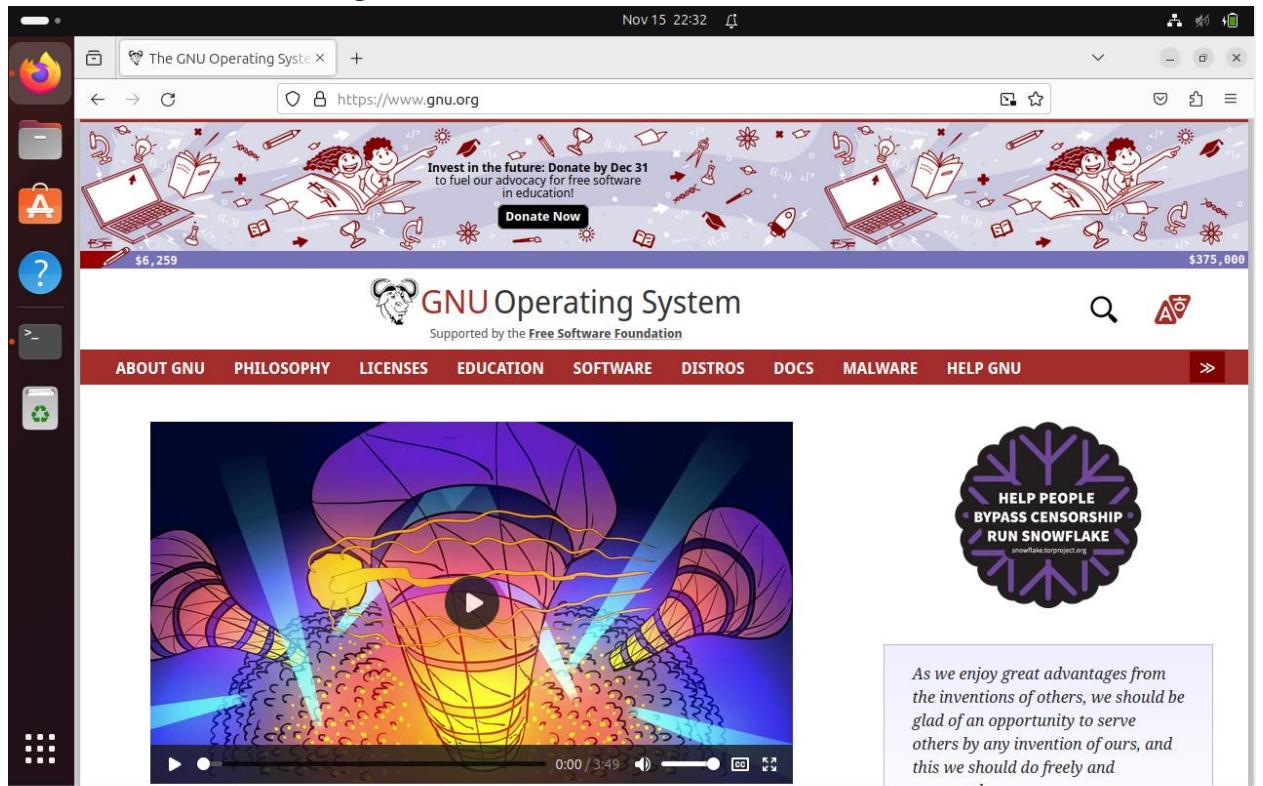
- **0,15,30,45 * * * * echo "hello"** → this runs at 0min, 15min, 30min, 45min at every hour, day, month, year.
- ***/15 * * * * echo "hello"** → this runs every 15min
- ***/15 * */3 * * echo "hello"** → this runs every 15min every 3 days
- **59 23 * JAN,DEC SUN echo "hello"** → this runs at 23:59 on every Sunday in the months of JAN and DEC

GNU Project

- Sep 1983 – Stallman announces GNU(open source operating System) Project
- Jan 1984 – Stallman quits MIT to work on GNU full time.
- Free Software – 4 Freedoms
 - 1) freedom to run programs as you wish, for any purpose
 - 2) freedom to study how the program works and change it so it does your computing as you wish (freedom 1). Access to source code is a precondition for this.
 - 3) The freedom to redistribute copies so you can help your neighbours.
 - 4) The freedom to distribute copies of your modified versions to others (freedom 3). By doing this you can give the whole community a chance to benefit from your changes. Access to the source code is a precondition.
- There is a license that provides with these freedoms and it's known as GNU public license(GPL)
- By 1991, GNU Almost Complete, but there was still one part missing from the GNU Operating System and it was an important part. This final piece is a major part of every operating system and is called kernel. Now, the Kernel is responsible for allocating the resources on the computer hardware that is required by the software that it's running. It's the interface layer between a computer's hardware and the applications that it's running.
- Now, at this time, in 1991, a man called Linus Torvalds invented a unix like kernel called Linux and it was called Linux after his own name. But everything in the Unix world kind of ends in an X. So Linus ending in an X is Linux.
- In 1991 – Linus released Linux Kernel under GPL v2.0 thereby making Linux kernel free software.
- Linux Kernel Completed GNU Project, a fully runnable operating system made of entirely free software.
- To know the details of the operating system the command is
uname -o

```
mohan@mohan:~/bin$ uname -o
GNU/Linux
mohan@mohan:~/bin$
```

- We can see all the open-source code at the GNU official website(www.gnu.org) →
 Software → All GNU Packages



Links to the home pages of all current GNU packages are given below, using their identifiers (rather than long names) for brevity. They are sorted alphabetically from left to right. If you have corrections to this list or questions about it, please email <maintainers@gnu.org>.

Also available are lists of:

- [short descriptions for all GNU packages](#);
- [documentation for GNU packages](#) (arranged by category);
- [GNU package logos](#); and
- [recent GNU releases](#).

3dldf	8sync	a2ps	acct	acm	adns
alive	anastasis	anubis	apl	archimedes	aris
artanis	aspell	auctex	autoconf	autoconf-archive	autogen
automake	avl	ballandpaddle	barcode	bash	bayonne
bazaar	bc	behistun	bfd	binutils	bison
bool	bpel2owfn	c-graph	ccaudio	ccd2cue	ccide
ccrtp	ccscript	cflow	cgicc	chess	cim
classpath	classpathx	clisp	combine	commoncpp	complexity
config	consensus	coreutils	cpio	cppi	cssc
cursynth	dap	datamash	dc	ddd	ddrescue
dejagnu	denemo	dia	dico	diction	diffutils
direvent	djgpp	dominion	dr-geo	easejs	ed
edma	electric	emacs	emacs-muse	emms	enscript
		ferret	findutils	fiscalab	foliot
https://www.gnu.org/software/emms/					

- Linux-libre is the free software version of the Linux Kernel itself, which we can find under the Linux Libre package. It contains all the source code.
- Under coreutils package, we can find source code for all the commands like ls, mkdir etc

Modifying and Installing from Source Code

- In this section, we download, compile and install a package (coreutils package that contains commands like ls, find etc)

Send general questions or suggestions to the mailing list at <coreutils@gnu.org>.
Send translation requests to the language team at the Translation Project.
Report bugs, including version and distribution variant, to the list at <bug-coreutils@gnu.org>. Before sending the bug, please consult the FAQ and mailing list archives (above). Often these perceived bugs are simply due to wrong program usage. To learn more about reporting bugs, see Getting help with GNU software.

Downloads

Stable source releases are available on the main GNU download server (via HTTPS, HTTP or FTP, and its mirrors. Please use a mirror if possible.

Source Code

The latest source with revision history can be browsed using cgit, gitee or GitHub. Assuming you have git installed, you can retrieve the latest version with this command:

```
git clone git://git.sv.gnu.org/coreutils
```

A Coreutils code structure overview is available, which is useful for educational purposes, or for those interested in contributing changes. To build from the latest sources please follow the instructions in README-hacking. Please note that we do not suggest using test versions of Coreutils for production use.

Index of /gnu/coreutils

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
Parent Directory		-	
coreutils-5.0.tar.bz2	2003-04-02 16:26	3.8M	
coreutils-5.0.tar.bz2.sig	2003-04-02 16:26	65	
coreutils-5.0.tar.gz	2003-04-02 16:25	5.7M	
coreutils-5.0.tar.gz.sig	2003-04-02 16:26	65	
coreutils-5.1.3-5.2.0.xdelta	2004-02-19 16:48	223K	
coreutils-5.1.3-5.2.0.xdelta.sig	2004-02-19 16:48	189	
coreutils-5.2.0-5.2.1.xdelta	2004-03-12 14:37	297K	
coreutils-5.2.0-5.2.1.xdelta.sig	2004-03-12 14:37	189	
coreutils-5.2.0.tar.bz2	2004-02-19 16:52	4.0M	
coreutils-5.2.0.tar.bz2.sig	2004-02-19 16:52	189	
coreutils-5.2.0.tar.gz	2004-02-19 16:59	6.2M	
coreutils-5.2.0.tar.gz.sig	2004-02-19 16:59	189	
coreutils-5.2.1.tar.bz2	2004-03-12 14:48	4.1M	
coreutils-5.2.1.tar.bz2.sig	2004-03-12 14:48	189	
coreutils-5.2.1.tar.gz	2004-03-12 14:44	6.3M	
coreutils-5.2.1.tar.gz.sig	2004-03-12 14:44	189	
coreutils-5.91-5.92.xdelta	2005-10-22 18:14	148K	

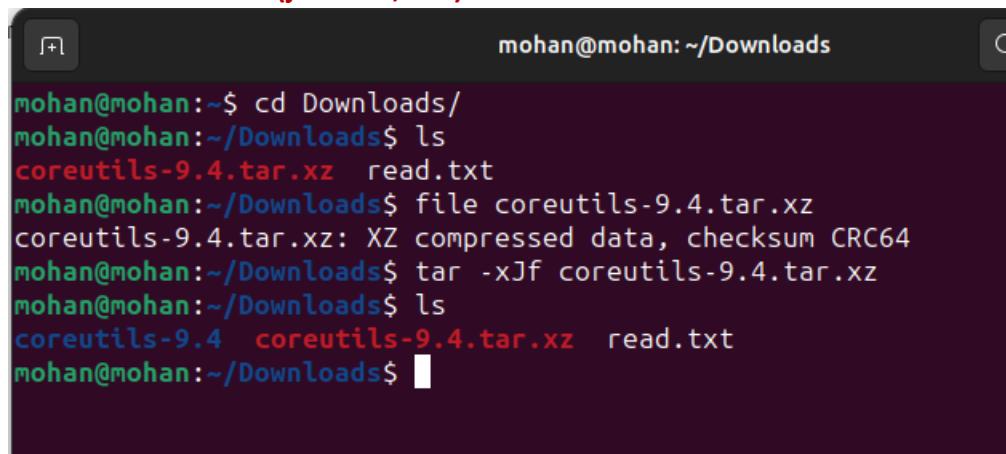
- Download the latest version that's ending with .xz

 [coreutils-9.4.tar.xz](#)

- Unzip the tar file compressed using XZ.

command to unzip for XZ is

tar -xJf <filename> (j for bz2, J xz)



```
mohan@mohan:~$ cd Downloads/
mohan@mohan:~/Downloads$ ls
coreutils-9.4.tar.xz  read.txt
mohan@mohan:~/Downloads$ file coreutils-9.4.tar.xz
coreutils-9.4.tar.xz: XZ compressed data, checksum CRC64
mohan@mohan:~/Downloads$ tar -xJf coreutils-9.4.tar.xz
mohan@mohan:~/Downloads$ ls
coreutils-9.4  coreutils-9.4.tar.xz  read.txt
mohan@mohan:~/Downloads$
```

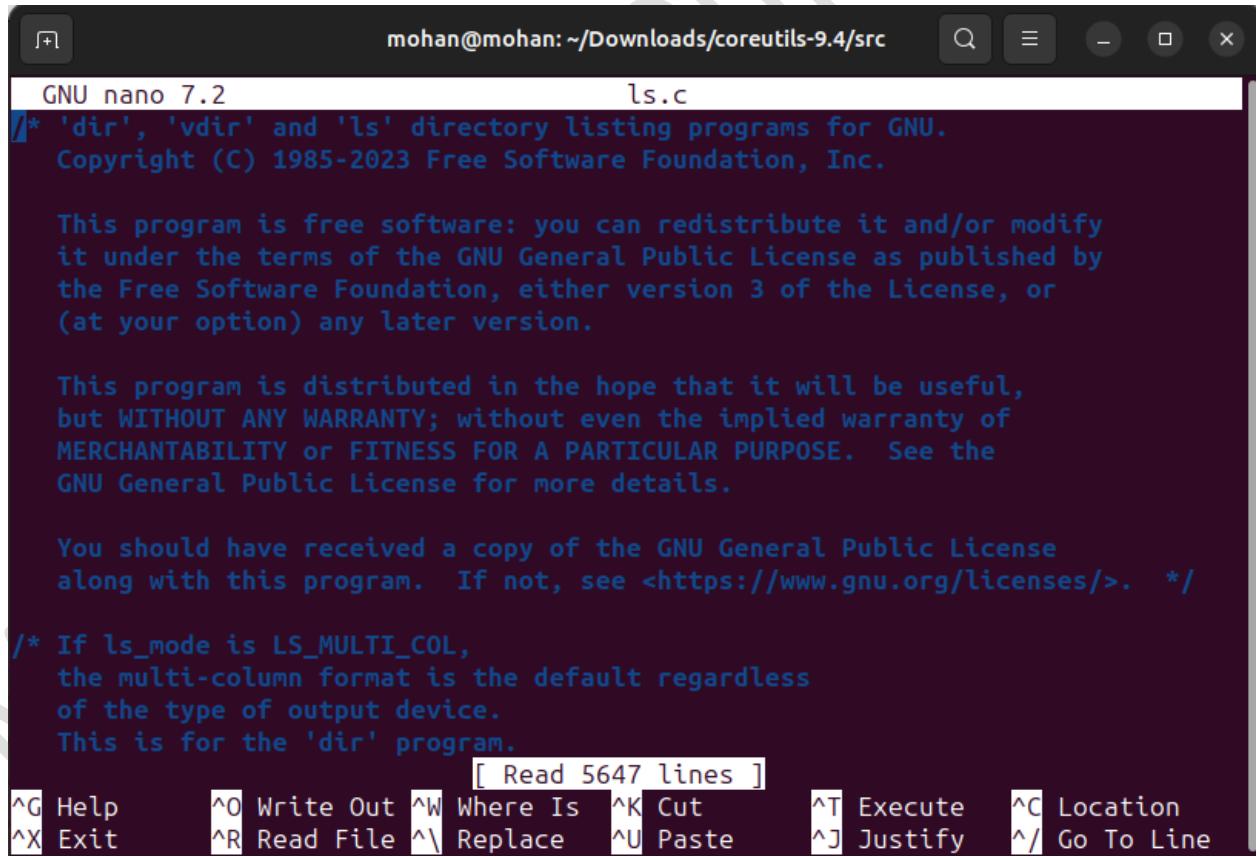
69 | Page

- Search ls using grep and nano 'ls.c' file to edit the source code



```
mohan@mohan:~/Downloads/coreutils-9.4/src$ ls | grep ls
coreutils-arch.c
coreutils.c
coreutils-dir.c
coreutils-vdir.c
false.c
ls.c
ls-dir.c
ls.h
ls-ls.c
ls-vdir.c
mohan@mohan:~/Downloads/coreutils-9.4/src$
```

nano ls.c → and modify the code by adding some print statement



```
GNU nano 7.2                               ls.c
/* 'dir', 'vdir' and 'ls' directory listing programs for GNU.
   Copyright (C) 1985-2023 Free Software Foundation, Inc.

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program. If not, see <https://www.gnu.org/licenses/>. */

/* If ls_mode is LS_MULTI_COL,
   the multi-column format is the default regardless
   of the type of output device.
   This is for the 'dir' program.

[ Read 5647 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut      ^T Execute  ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste    ^J Justify  ^/ Go To Line
```

- Adding print statement to the main method, save and exit

```

GNU nano 7.2                               ls.c *
signal_setup (true);
}

static void
signal_restore (void)
{
    signal_setup (false);
}

int
main (int argc, char **argv)
{
printf("Mohan Kumar Pulibanti \n")
    int i;
    struct pending *thispend;
    int n_files;

    initialize_main (&argc, &argv);
    set_program_name (argv[0]);
    setlocale (LC_ALL, "");

^G Help      ^O Write Out ^W Where Is  ^K Cut      ^T Execute   ^C Location
^X Exit      ^R Read File ^Y Replace   ^U Paste      ^J Justify   ^/ Go To Line

```

- The next step is to compile the code change. GCC is the compiler (GNU c compiler) to compile .c files in GNU + Linux OS.
- To install gcc, below is the command
`sudo apt-get install gcc`
- As different systems have different architectures, we need to configure the installation of the code to our specific machine. now, there is a script that comes called configure.

```

mohan@mohan:~/Downloads/coreutils-9.4$ ls
ABOUT-NLS      configure.ac   m4          src
aclocal.m4      COPYING        maint.mk    tests
AUTHORS        dist-check.mk  Makefile.am THANKS
bootstrap       doc           Makefile.in thanks-gen
bootstrap.conf  gnutest       man         THANKS.in
build-aux       GNUMakefile   NEWS        THANKS-to-translators
cfg.mk         init.cfg      po          THANKStt.in
ChangeLog      INSTALL       README     TODO
configure      lib           README-install
mohan@mohan:~/Downloads/coreutils-9.4$ 

```

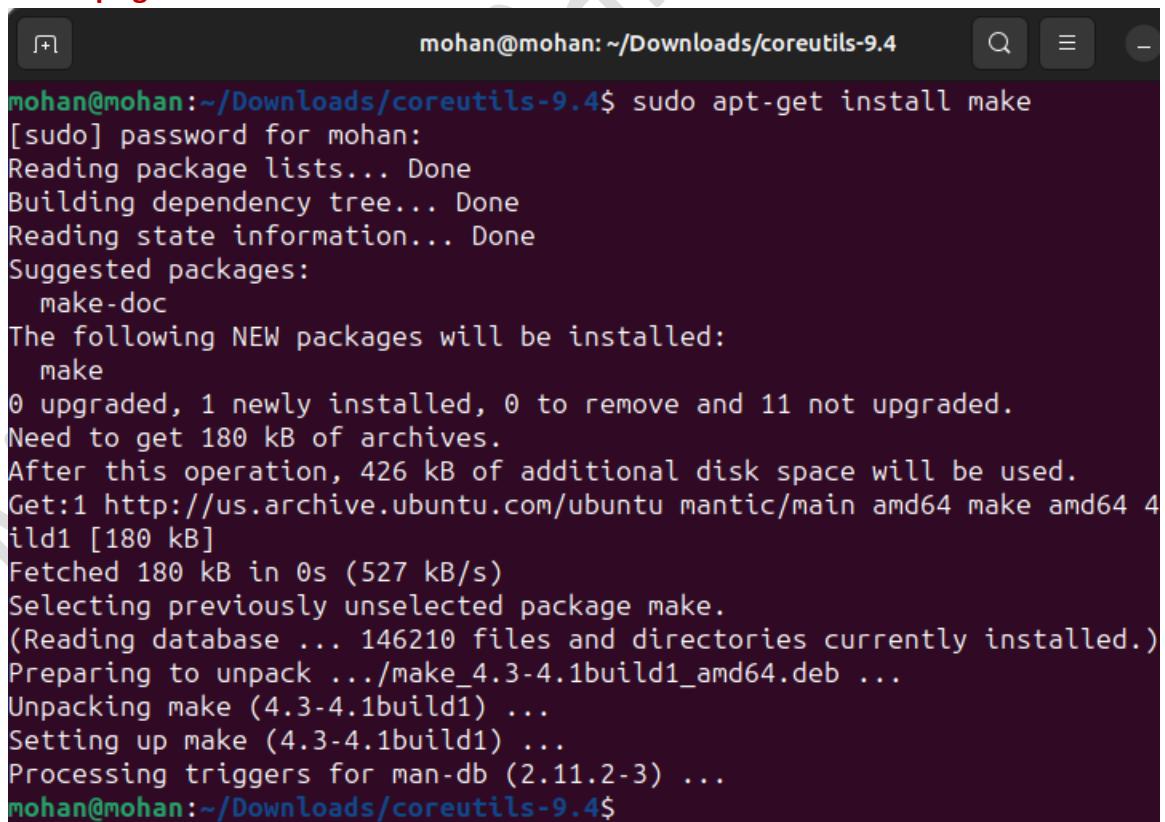
- Run the configure (it's in green color which means it's a script) with the command **bash configure** → this configures the GNU C compiler, the GCC, to make sure that when it compiles all this that we tell it to, it's gonna do it in a way that's appropriate for our computer's architecture and it's making the appropriate adjustments. Besides compiling this creates a new file called the make file. Make file is responsible for the installation of this new software package. To make the make file work, we need a new command called make.

- Once we ran this configure file we should see the **Makefile** in the same directory

```
mohan@mohan:~/Downloads/coreutils-9.4$ ls
ABOUT-NLS      configure     m4          src
acllocal.m4    configure.ac  maint.mk   tests
AUTHORS        COPYING       Makefile   THANKS
bootstrap      dist-check.mk Makefile.am THANKS-gen
bootstrap.conf doc          Makefile.in THANKS.in
build-aux      gnulib-tests man        THANKS-to-translators
cfg.mk         GNUmakefile  NEWS       THANKSt.in
ChangeLog      init.cfg    po         TODO
config.log     INSTALL     README    README-install
config.status  lib         README-install
```

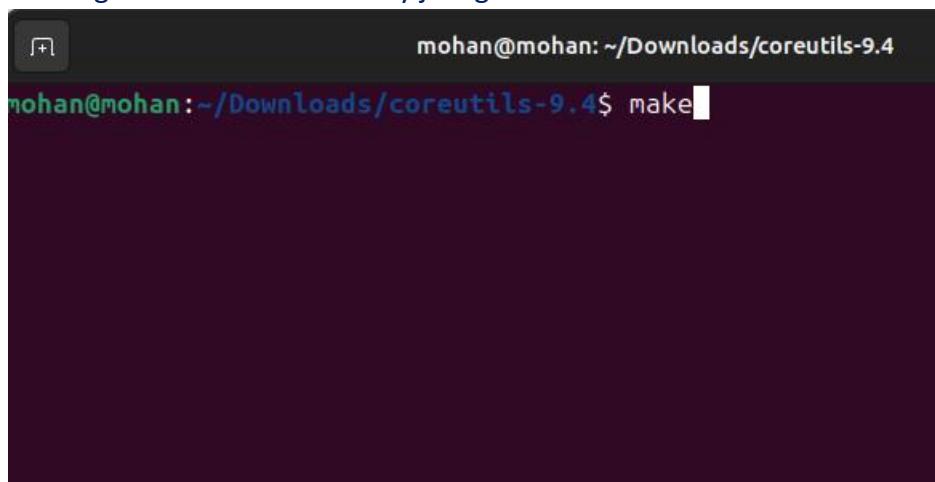
- to be able to run make file, we need make command, Installing the make command.

sudo apt-get install make



```
mohan@mohan:~/Downloads/coreutils-9.4$ sudo apt-get install make
[sudo] password for mohan:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  make-doc
The following NEW packages will be installed:
  make
0 upgraded, 1 newly installed, 0 to remove and 11 not upgraded.
Need to get 180 kB of archives.
After this operation, 426 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu mantic/main amd64 make amd64 4
ild1 [180 kB]
Fetched 180 kB in 0s (527 kB/s)
Selecting previously unselected package make.
(Reading database ... 146210 files and directories currently installed.)
Preparing to unpack .../make_4.3-4.1build1_amd64.deb ...
Unpacking make (4.3-4.1build1) ...
Setting up make (4.3-4.1build1) ...
Processing triggers for man-db (2.11.2-3) ...
mohan@mohan:~/Downloads/coreutils-9.4$
```

- running the make command by just give make in the terminal.



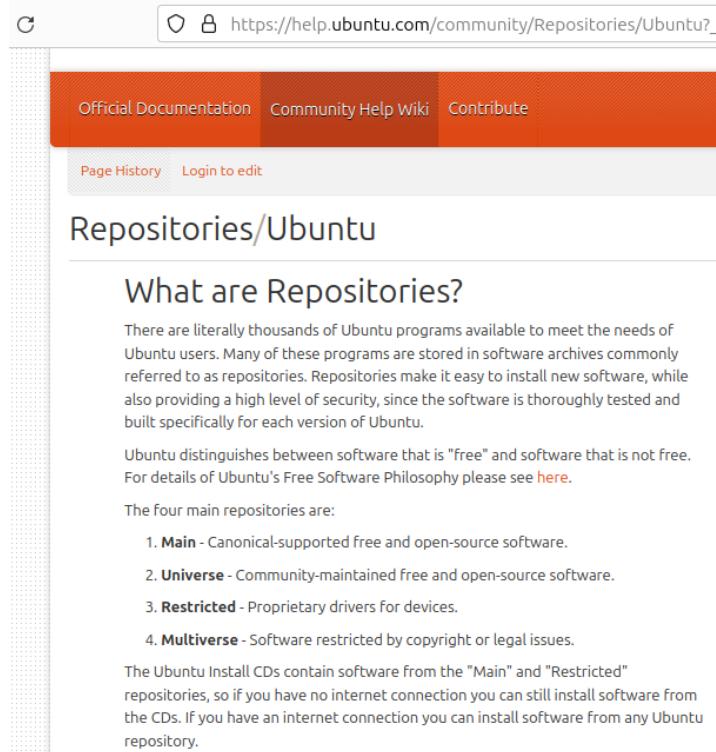
A screenshot of a terminal window titled "mohan@mohan: ~/Downloads/coreutils-9.4". The command "make" is typed at the prompt, and the terminal is waiting for the user to press Enter. The background of the terminal is dark, and the text is white.

make → it's gonna compile all of the C files that it finds have not been compiled recently and there are still some changes outstanding which is effectively all of the C code in this folder including our ls file. It's compiling all into machine code (binary code) that can be run on the computer. When this is done, we will be able to install the newly compiled programs.

- The last step that remained is, now that we've got this machine code, is to install that machine code in the required places on our path for it to work. we can do that using the command **sudo make install**

Software repositories

- Think Software repository as a big library filled with software.
- In ubuntu, there are 4 different types of repositories



The screenshot shows a web browser displaying the Ubuntu Help page for 'Repositories/Ubuntu'. The URL in the address bar is https://help.ubuntu.com/community.Repositories/Ubuntu?_. The page has a red header with links for 'Official Documentation', 'Community Help Wiki', and 'Contribute'. Below the header, there are links for 'Page History' and 'Login to edit'. The main content area is titled 'Repositories/Ubuntu' and contains a section titled 'What are Repositories?'. It explains that there are thousands of Ubuntu programs available and describes the four main repositories: Main, Universe, Restricted, and Multiverse. It also mentions that the Ubuntu Install CDs contain software from the Main and Restricted repositories.

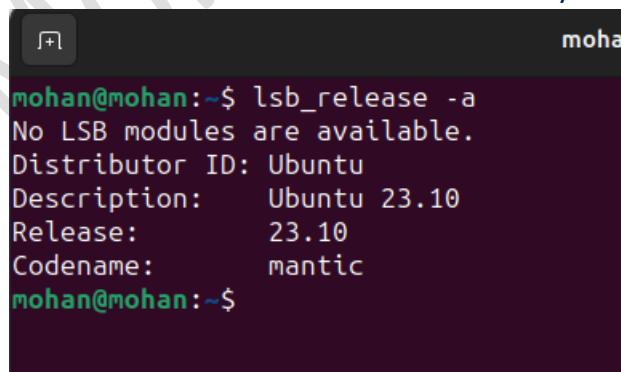
Main → Maintained by Canonical (Free, open Source)

Universe → Maintained by Community (Free, open source)

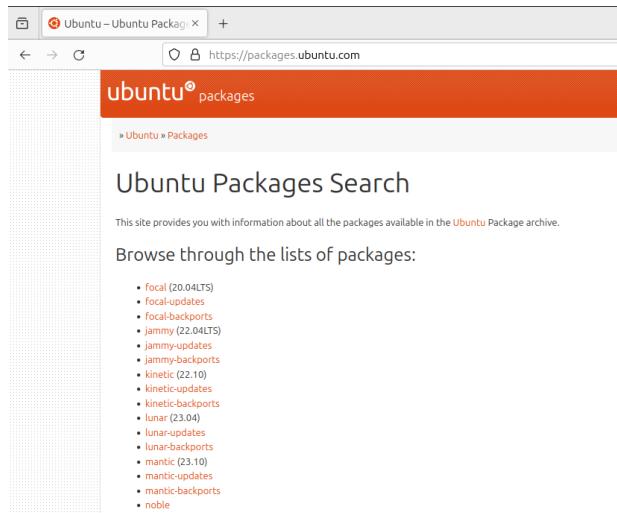
Restricted → Proprietary Drivers (Proprietary, Maybe Open Source)

Multiverse → Limited by Copyright of Legal Issues (proprietary, Maybe Open Source)

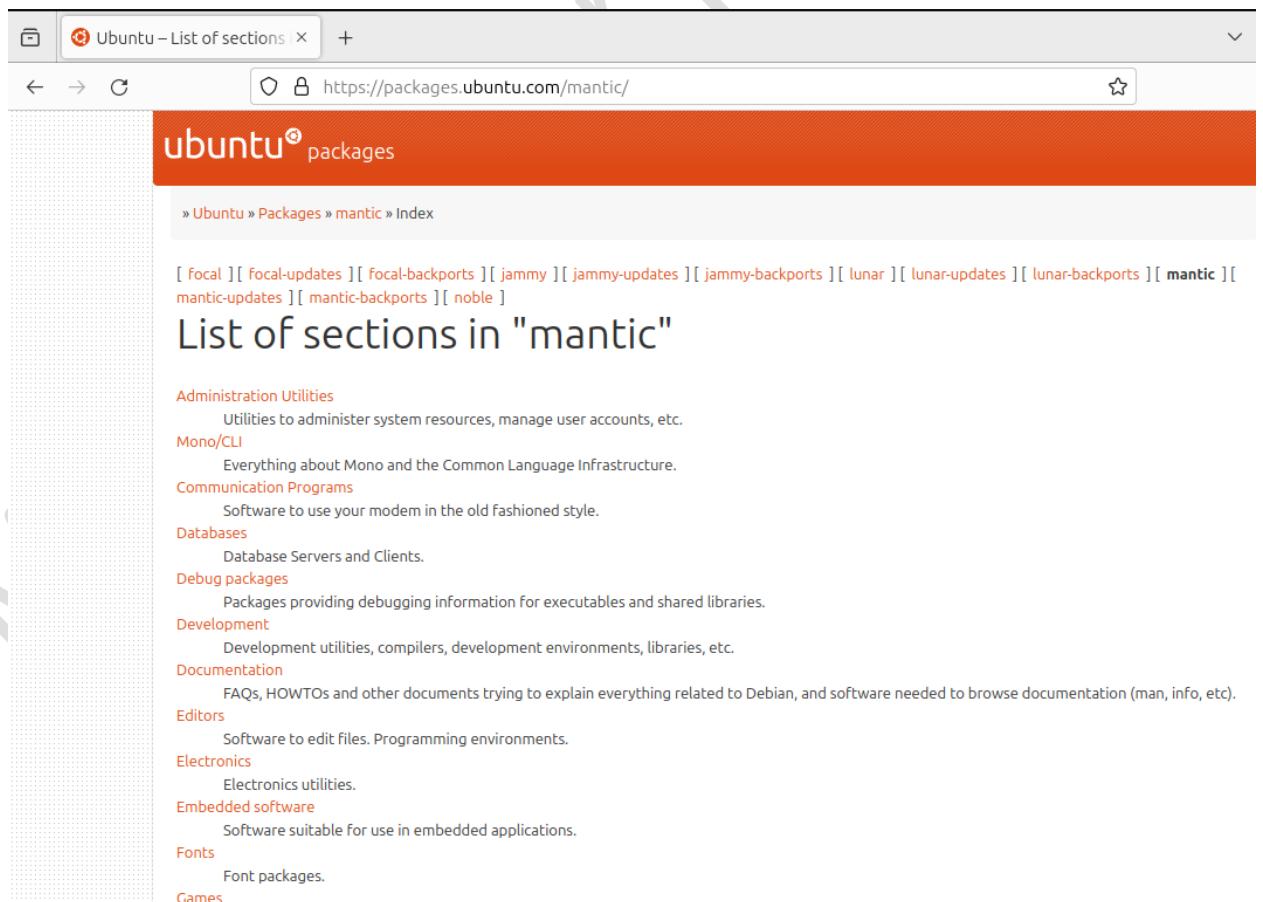
- packages.ubuntu.com → is the website where we see list of packages for different versions of ubuntu. For every new release of ubuntu the name gets changed. You can see the code name of distribution by using the command **lsb_release -a**



```
mohan@mohan:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 23.10
Release:        23.10
Codename:       mantic
mohan@mohan:~$
```



- click on the relevant link based on your ubuntu version, mine is mantic. Once clicked, we will see a whole bunch of categories in here



- At the bottom of this page , we will see All packages and compact Compressed textlist Compact compressed textlist → this is a list of every single one of the packages.
- When we click on all packages, we will see all the packages and from which repository it came from in a square brace. If you don't see square braces mentioning the package, that means it's from main package.

Software Packages in "mantic"

Oad (0.0.26-4) [universe]
 Real-time strategy game of ancient warfare

Oad-data (0.0.26-1) [universe]
 Real-time strategy game of ancient warfare (data files)

Oad-data-common (0.0.26-1) [universe]
 Real-time strategy game of ancient warfare (common data files)

Oinstall (2.18-2build2) [universe]
 cross-distribution packaging system

Oinstall-core (2.18-2build2) [universe]
 cross-distribution packaging system (non-GUI parts)

- Click on any of the package, we will get all detail of the packages and what other packages this package is dependent on.

[Source: Oad]

Package: Oad (0.0.26-4) [universe]

Real-time strategy game of ancient warfare

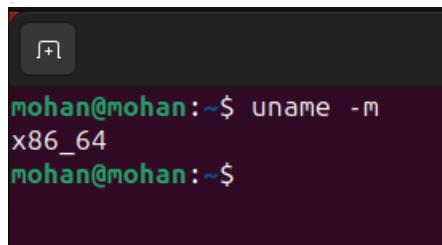
Other Packages Related to Oad

● depends	◆ recommends	■ suggests	• enhances
● dpkg (>= 1.15.6~) Debian package management system			
● Oad-data (<= 0.0.26-4) Real-time strategy game of ancient warfare (data files)			
● Oad-data (>= 0.0.26)			
● Oad-data-common (<= 0.0.26-4) Real-time strategy game of ancient warfare (common data files)			
● Oad-data-common (>= 0.0.26)			
● libboost-filesystem1.74.0 (>= 1.74.0+ds1) Filesystem operations (portable paths, iteration over directories, etc) in C++			
● libc6 (>= 2.34) GNU C Library: Shared libraries			
● libcurl3-gnutls (>= 7.32.0) Easy-to-use client-side URL transfer library (GnuTLS flavour)			

- Depends – mandatory for the package to function
- recommends – needs for normal use
- Suggested – Suggested for Interest
- Enhances – Enhances the package in some way
- When we scroll down to the bottom of page, we see download links and size for different architecture. To know your computer architecture, use command **uname -m**
X86_64 means amd64

Download Oad

Architecture	Package Size	Installed Size	Files
amd64	7,298.3 kB	21,921.0 kB	[list of files]
arm64	8,205.7 kB	26,950.0 kB	[list of files]
armhf	7,617.2 kB	16,831.0 kB	[list of files]



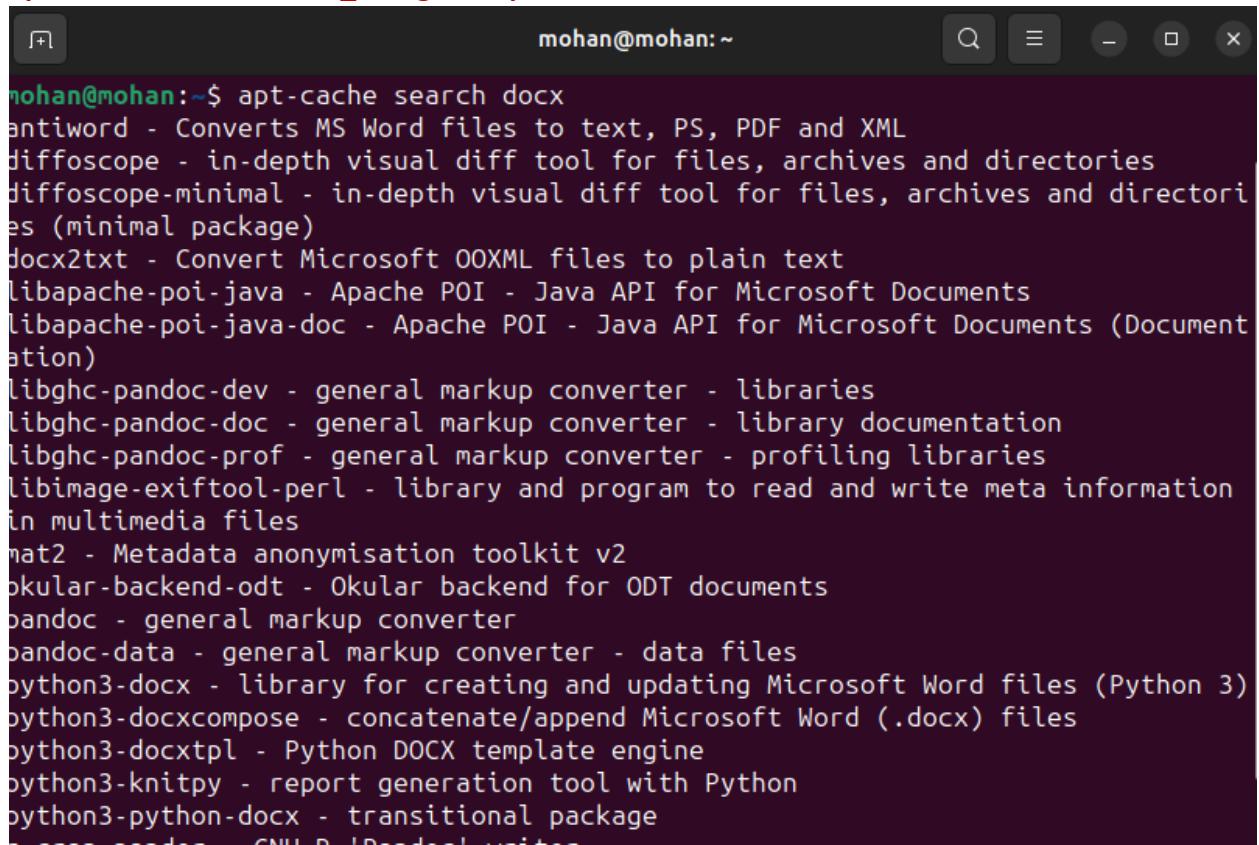
```
mohan@mohan:~$ uname -m
x86_64
mohan@mohan:~$
```

- In Ubuntu, the package manager is called apt (which stands for advanced packaging tool) which takes care of installing the packages we need and all its dependencies.

Searching for Packages Using apt

- Package manager (Apt) is used to download and install software's in Linux.
- Command to search packages (lets search package for Microsoft word file)

apt-cache search <search_string> → apt-cache search docx



```
mohan@mohan:~$ apt-cache search docx
antiword - Converts MS Word files to text, PS, PDF and XML
diffoscope - in-depth visual diff tool for files, archives and directories
diffoscope-minimal - in-depth visual diff tool for files, archives and directories (minimal package)
docx2txt - Convert Microsoft OOXML files to plain text
libapache-poi-java - Apache POI - Java API for Microsoft Documents
libapache-poi-java-doc - Apache POI - Java API for Microsoft Documents (Documentation)
libghc-pandoc-dev - general markup converter - libraries
libghc-pandoc-doc - general markup converter - library documentation
libghc-pandoc-prof - general markup converter - profiling libraries
libimage-exiftool-perl - library and program to read and write meta information in multimedia files
mat2 - Metadata anonymisation toolkit v2
okular-backend-odt - Okular backend for ODT documents
pandoc - general markup converter
pandoc-data - general markup converter - data files
python3-docx - library for creating and updating Microsoft Word files (Python 3)
python3-docxcompose - concatenate/append Microsoft Word (.docx) files
python3-docxtpl - Python DOCX template engine
python3-knitrpy - report generation tool with Python
python3-python-docx - transitional package
```

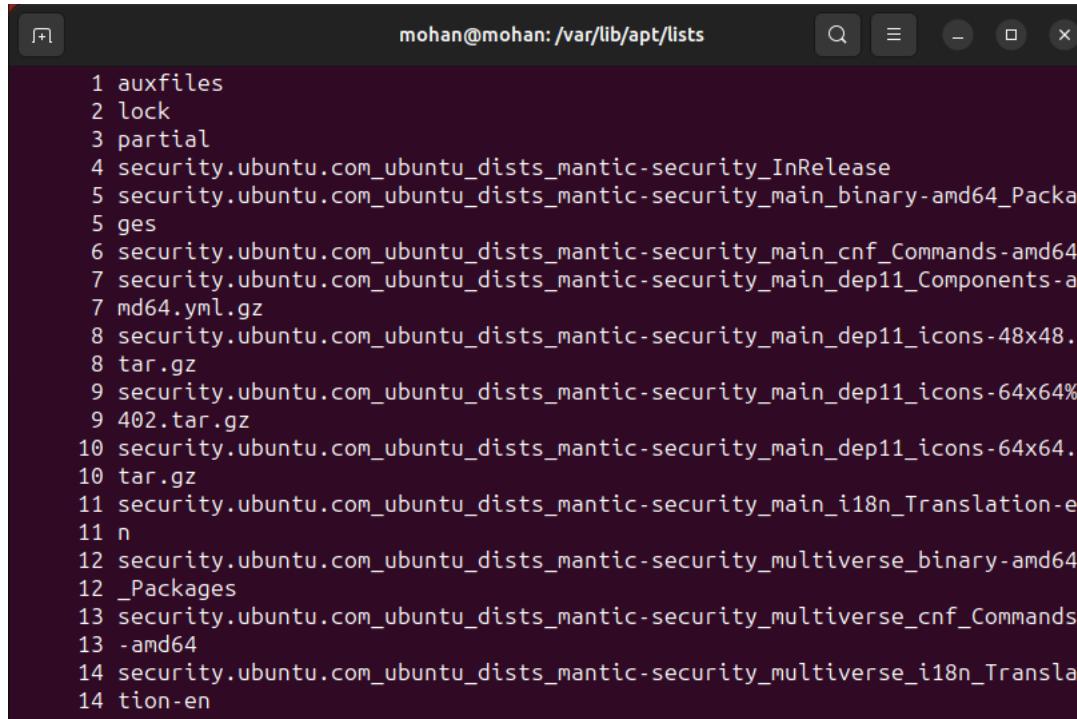
- To get more details of a specific package, the command is
apt-cache show <package_name> → **apt-cache show docx2txt**

```
mohan@mohan:~$ apt-cache show docx2txt
Package: docx2txt
Architecture: all
Version: 1.4-5
Multi-Arch: foreign
Priority: extra
Section: universe/text
Origin: Ubuntu
Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Original-Maintainer: Barak A. Pearlmuter <bap@debian.org>
Bugs: https://bugs.launchpad.net/ubuntu/+filebug
Installed-Size: 46
Depends: unzip
Filename: pool/universe/d/docx2txt/docx2txt_1.4-5_all.deb
Size: 15728
MD5sum: 24e534318c4c73dce57bd045a88b5e63
SHA1: a8d1b4496755b25ec37fce2e0c01b02b3d9bc1a4
SHA256: 3c67fad4140fee70f3f033f53495622488a777eaad3da3e05044c8cec8905900
SHA512: 2228dd8981c3a342ad784dd9e2c862a51d0bab05a235ab2cb652599a188fa954ab85fb51
647a15fa6a319f3803e2949fddb4e49be3c0c9479f52a8c7e456d99d
Homepage: http://docx2txt.sourceforge.net
Description-en: Convert Microsoft OOXML files to plain text
  This tool attempts to generate equivalent plain text files from
```

- Cache is kind of local storage and it needs to be update to get the right details of the package. In Linux, the cache is located at location **/var/lib/apt/lists**

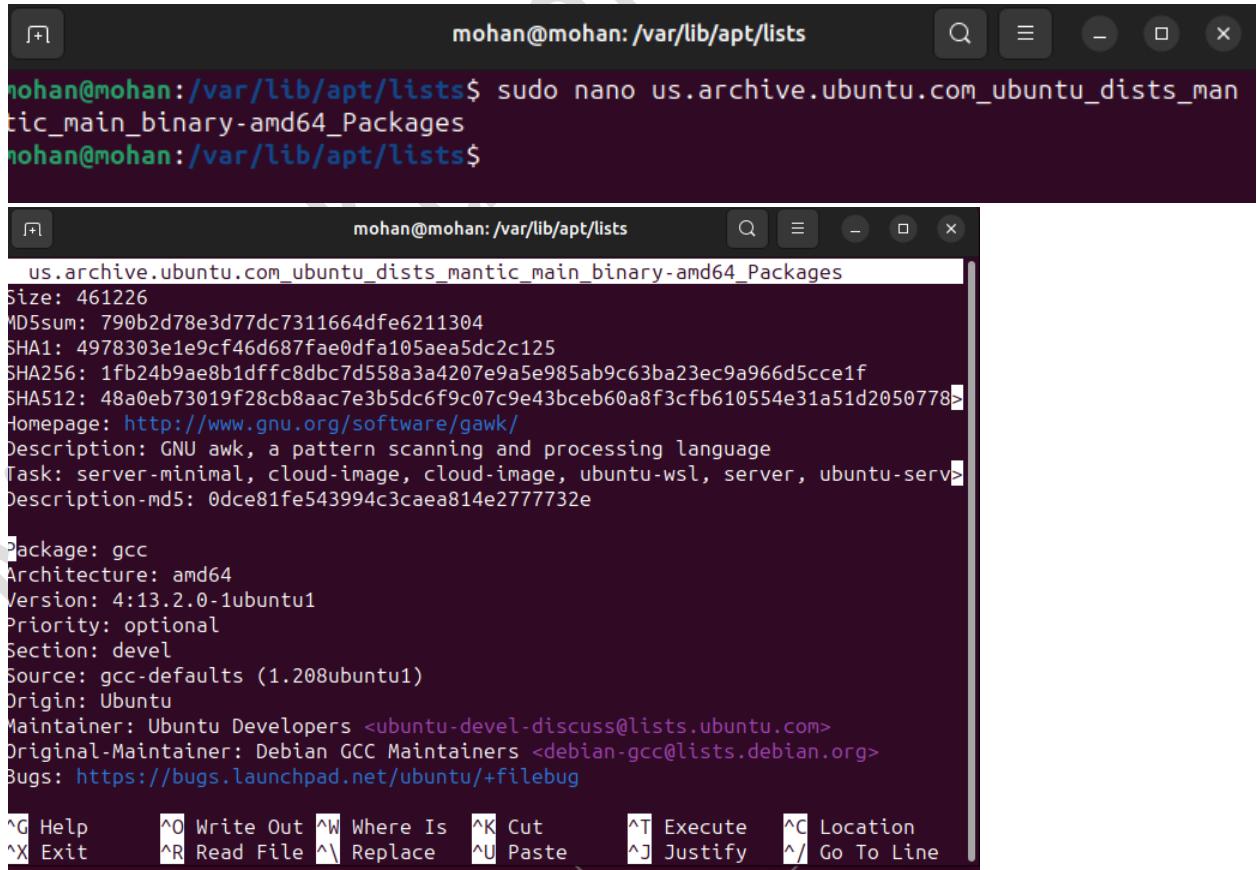
```
mohan@mohan: /var/lib/apt/lists
mohan@mohan:~$ cd /var/lib/apt/lists/
mohan@mohan:/var/lib/apt/lists$ ls -l | less -N
```

Option N for less is to show numbers before every result.



```
mohan@mohan: /var/lib/apt/lists
1 auxfiles
2 lock
3 partial
4 security.ubuntu.com_ubuntu_dists_mantic-security_InRelease
5 security.ubuntu.com_ubuntu_dists_mantic-security_main_binary-amd64_Packa
5 ges
6 security.ubuntu.com_ubuntu_dists_mantic-security_main_cnf_Commands-amd64
7 security.ubuntu.com_ubuntu_dists_mantic-security_main_dep11_Components-a
7 md64.yml.gz
8 security.ubuntu.com_ubuntu_dists_mantic-security_main_dep11_icons-48x48.
8 tar.gz
9 security.ubuntu.com_ubuntu_dists_mantic-security_main_dep11_icons-64x64%
9 402.tar.gz
10 security.ubuntu.com_ubuntu_dists_mantic-security_main_dep11_icons-64x64.
10 tar.gz
11 security.ubuntu.com_ubuntu_dists_mantic-security_main_i18n_Translation-e
11 n
12 security.ubuntu.com_ubuntu_dists_mantic-security_multiverse_binary-amd64
12 _Packages
13 security.ubuntu.com_ubuntu_dists_mantic-security_multiverse_cnf_Commands
13 -amd64
14 security.ubuntu.com_ubuntu_dists_mantic-security_multiverse_i18n_Transla
14 tion-en
```

- We can see all the packages by just opening any of this.



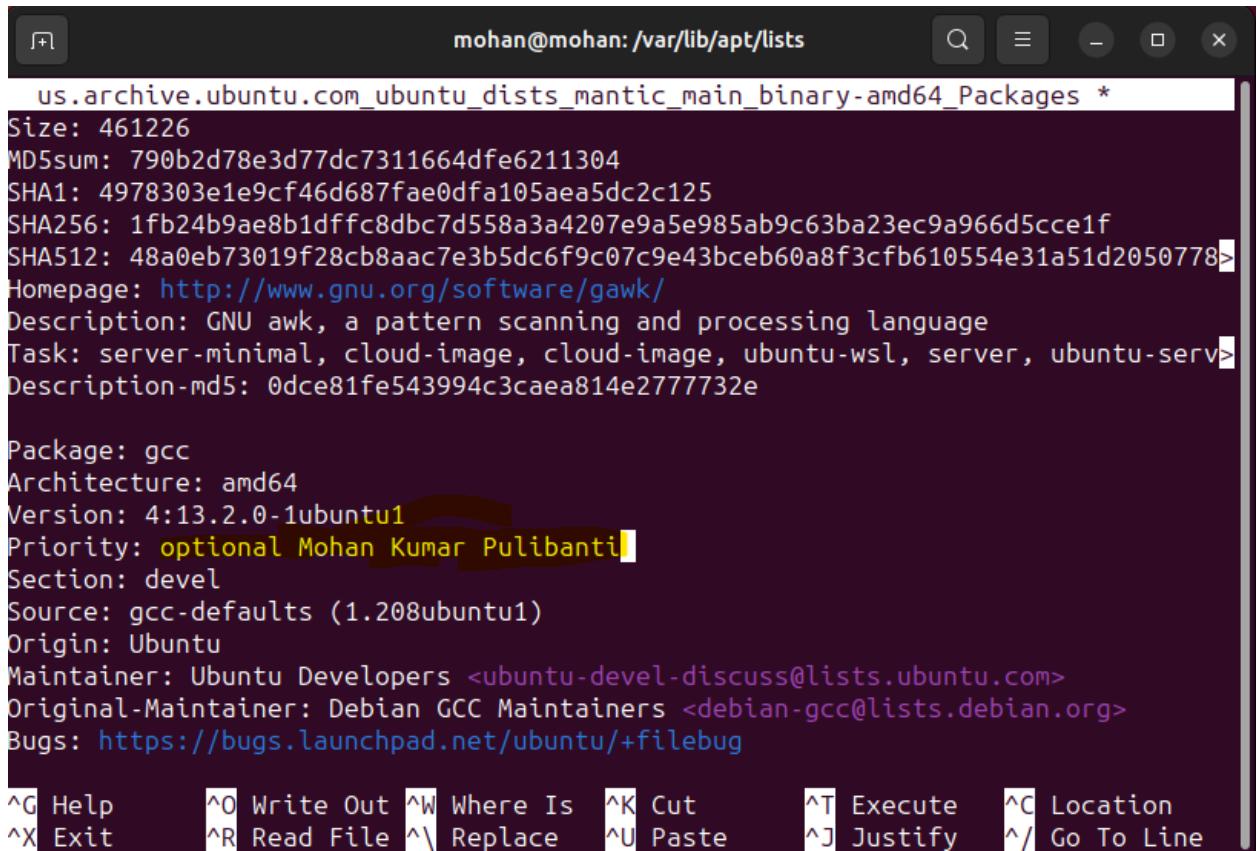
```
mohan@mohan: /var/lib/apt/lists
mohan@mohan:/var/lib/apt/lists$ sudo nano us.archive.ubuntu.com_ubuntu_dists_mantic_main_binary-amd64_Packages
mohan@mohan:/var/lib/apt/lists$
```

```
mohan@mohan: /var/lib/apt/lists
us.archive.ubuntu.com_ubuntu_dists_mantic_main_binary-amd64_Packages
Size: 461226
MD5sum: 790b2d78e3d77dc7311664dfe6211304
SHA1: 4978303e1e9cf46d687fae0dfa105aea5dc2c125
SHA256: 1fb24b9ae8b1dfffc8dbc7d558a3a4207e9a5e985ab9c63ba23ec9a966d5cce1f
SHA512: 48a0eb73019f28cb8aac7e3b5dc6f9c07c9e43bceb60a8f3cfb610554e31a51d2050778
Homepage: http://www.gnu.org/software/gawk/
Description: GNU awk, a pattern scanning and processing language
Task: server-minimal, cloud-image, cloud-image, ubuntu-wsl, server, ubuntu-serv
Description-md5: 0dce81fe543994c3cae814e2777732e

Package: gcc
Architecture: amd64
Version: 4:13.2.0-1ubuntu1
Priority: optional
Section: devel
Source: gcc-defaults (1.208ubuntu1)
Origin: Ubuntu
Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Original-Maintainer: Debian GCC Maintainers <debian-gcc@lists.debian.org>
Bugs: https://bugs.launchpad.net/ubuntu/+filebug

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line
```

- Edited optional value for package and saved it

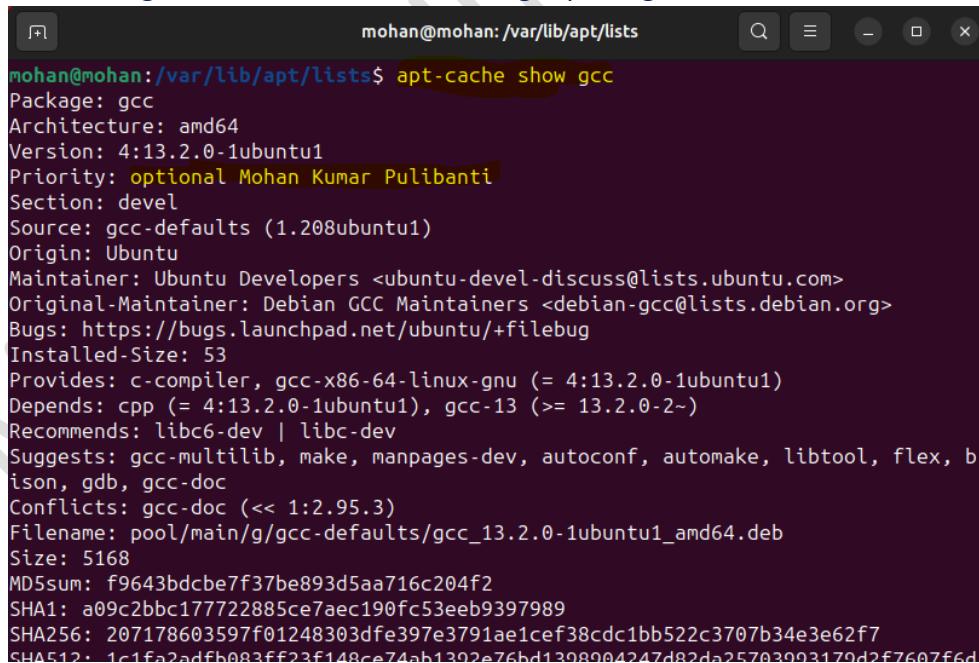


```
mohan@mohan:/var/lib/apt/lists
us.archive.ubuntu.com_ubuntu_dists_mantic_main_binary-amd64_Packages *
Size: 461226
MD5sum: 790b2d78e3d77dc7311664dfe6211304
SHA1: 4978303e1e9cf46d687fae0dfa105aea5dc2c125
SHA256: 1fb24b9ae8b1dfffc8dbc7d558a3a4207e9a5e985ab9c63ba23ec9a966d5cce1f
SHA512: 48a0eb73019f28cb8aac7e3b5dc6f9c07c9e43bceb60a8f3cfb610554e31a51d2050778>
Homepage: http://www.gnu.org/software/gawk/
Description: GNU awk, a pattern scanning and processing language
Task: server-minimal, cloud-image, cloud-image, ubuntu-wsl, server, ubuntu-serv>
Description-md5: 0dce81fe543994c3caea814e2777732e

Package: gcc
Architecture: amd64
Version: 4:13.2.0-1ubuntu1
Priority: optional Mohan Kumar Pulibanti
Section: devel
Source: gcc-defaults (1.208ubuntu1)
Origin: Ubuntu
Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Original-Maintainer: Debian GCC Maintainers <debian-gcc@lists.debian.org>
Bugs: https://bugs.launchpad.net/ubuntu/+filebug

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line
```

- The same got reflected when searched gcc package.

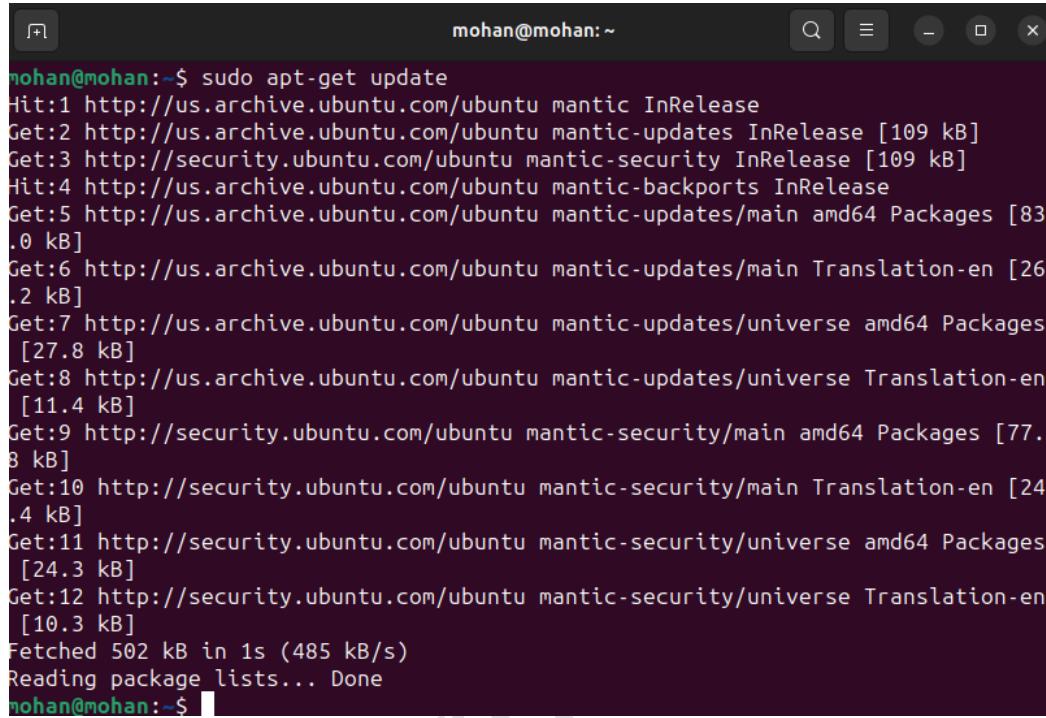


```
mohan@mohan:/var/lib/apt/lists$ apt-cache show gcc
Package: gcc
Architecture: amd64
Version: 4:13.2.0-1ubuntu1
Priority: optional Mohan Kumar Pulibanti
Section: devel
Source: gcc-defaults (1.208ubuntu1)
Origin: Ubuntu
Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Original-Maintainer: Debian GCC Maintainers <debian-gcc@lists.debian.org>
Bugs: https://bugs.launchpad.net/ubuntu/+filebug
Installed-Size: 53
Provides: c-compiler, gcc-x86-64-linux-gnu (= 4:13.2.0-1ubuntu1)
Depends: cpp (= 4:13.2.0-1ubuntu1), gcc-13 (>= 13.2.0-2~)
Recommends: libc6-dev | libc-dev
Suggests: gcc-multilib, make, manpages-dev, autoconf, automake, libtool, flex, bison, gdb, gcc-doc
Conflicts: gcc-doc (<< 1:2.95.3)
Filename: pool/main/g/gcc-defaults/gcc_13.2.0-1ubuntu1_amd64.deb
Size: 5168
MD5sum: f9643bdbe7f37be893d5aa716c204f2
SHA1: a09c2bbc177722885ce7aec190fc53eeb9397989
SHA256: 207178603597f01248303dfe397e3791ae1cef38cdc1bb522c3707b34e3e62f7
SHA512: 1c1fa2adfb083ff23f148ce74ab1392e76bd1398004247d82da25703993179d2f7607fed
```

Updating the cache and the software

- Command to update the cache (download the most updated package lists to the cache).
sudo apt-get update

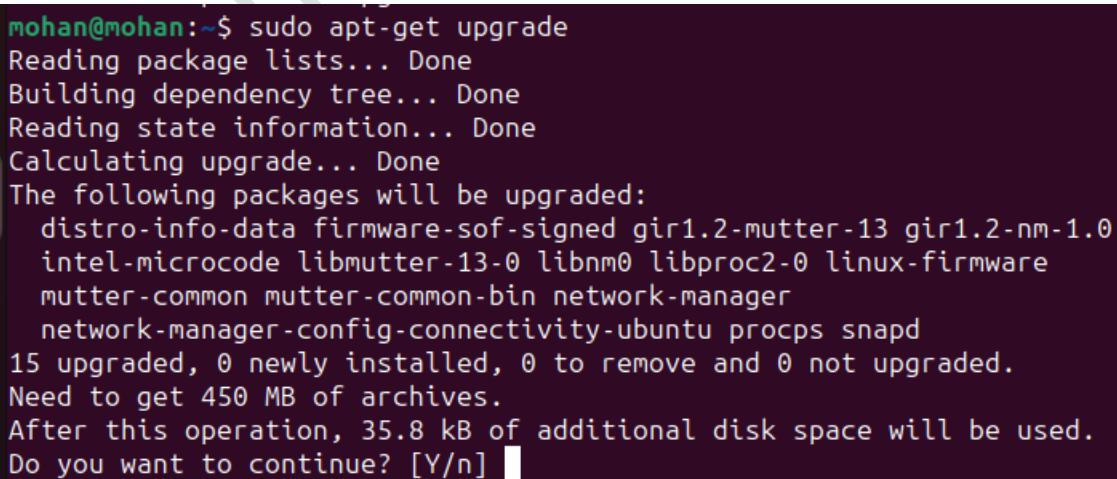
apt-get because we are getting data from the repositories.



```
mohan@mohan:~$ sudo apt-get update
Hit:1 http://us.archive.ubuntu.com/ubuntu mantic InRelease
Get:2 http://us.archive.ubuntu.com/ubuntu mantic-updates InRelease [109 kB]
Get:3 http://security.ubuntu.com/ubuntu mantic-security InRelease [109 kB]
Hit:4 http://us.archive.ubuntu.com/ubuntu mantic-backports InRelease
Get:5 http://us.archive.ubuntu.com/ubuntu mantic-updates/main amd64 Packages [83 .0 kB]
Get:6 http://us.archive.ubuntu.com/ubuntu mantic-updates/main Translation-en [26 .2 kB]
Get:7 http://us.archive.ubuntu.com/ubuntu mantic-updates/universe amd64 Packages [27.8 kB]
Get:8 http://us.archive.ubuntu.com/ubuntu mantic-updates/universe Translation-en [11.4 kB]
Get:9 http://security.ubuntu.com/ubuntu mantic-security/main amd64 Packages [77.8 kB]
Get:10 http://security.ubuntu.com/ubuntu mantic-security/main Translation-en [24 .4 kB]
Get:11 http://security.ubuntu.com/ubuntu mantic-security/universe amd64 Packages [24.3 kB]
Get:12 http://security.ubuntu.com/ubuntu mantic-security/universe Translation-en [10.3 kB]
Fetched 502 kB in 1s (485 kB/s)
Reading package lists... Done
mohan@mohan:~$
```

- To upgrade the software's on your system. This makes package manager look all the softwares that are currently on your system and compare them to what is available in the package-lists. Pacakage manager will automatically download and upgrade all the packages.

sudo apt-get upgrade



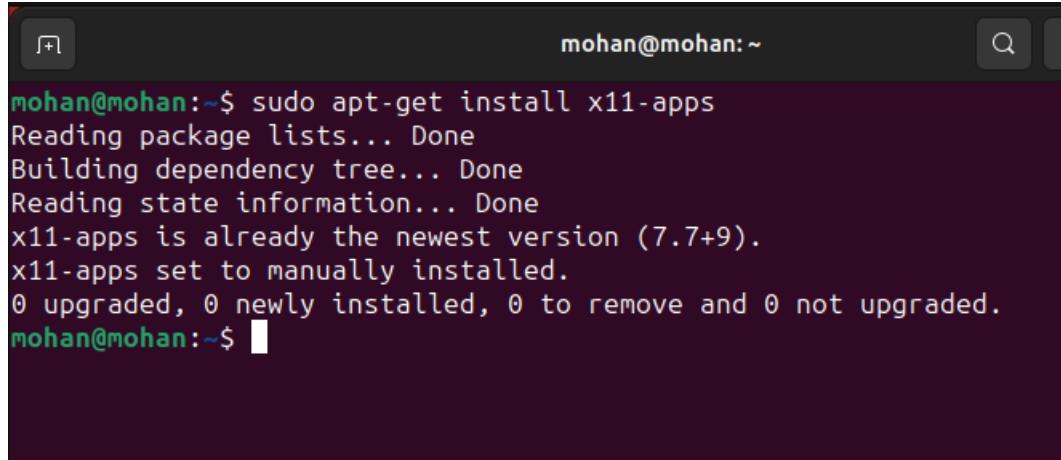
```
mohan@mohan:~$ sudo apt-get upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages will be upgraded:
  distro-info-data firmware-sof-signed gir1.2-mutter-13 gir1.2-nm-1.0
    intel-microcode libmutter-13-0 libnm0 libproc2-0 linux-firmware
    mutter-common mutter-common-bin network-manager
    network-manager-config-connectivity-ubuntu procps snapd
15 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 450 MB of archives.
After this operation, 35.8 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

- Always update package before upgrade software

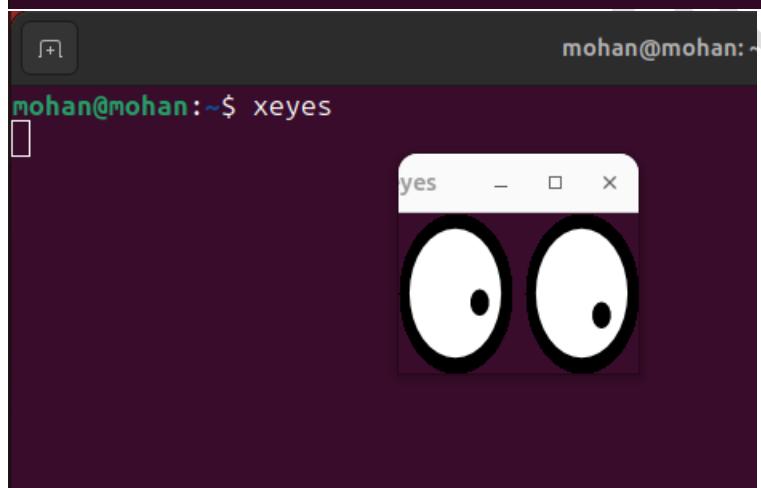
Installing New packages

➤ **sudo apt-get install <package_name>**

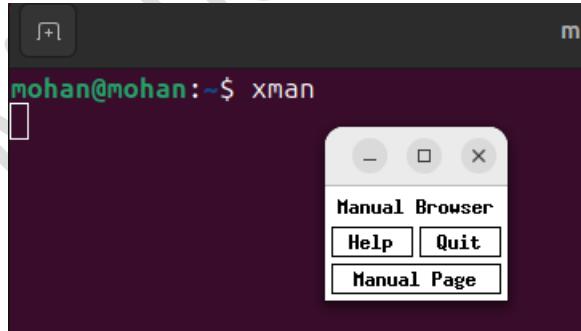
sudo apt-get install x11-apps

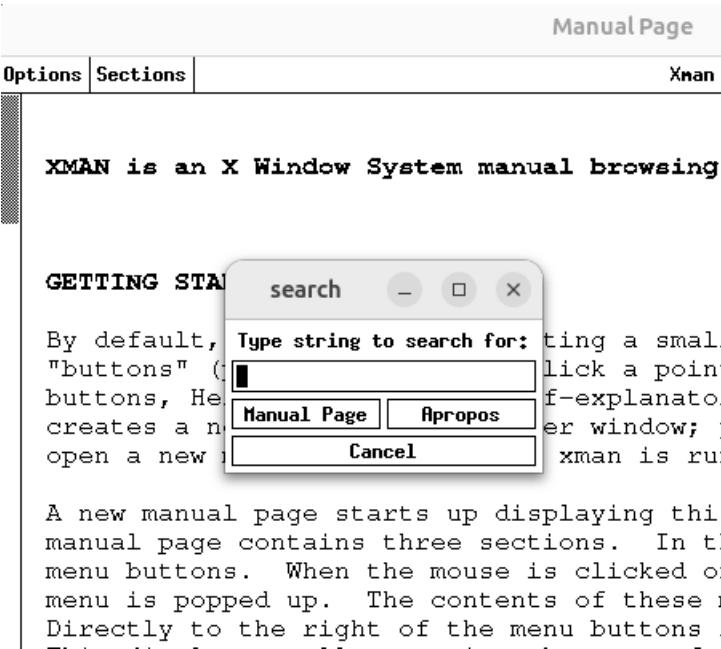


```
mohan@mohan:~$ sudo apt-get install x11-apps
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
x11-apps is already the newest version (7.7+9).
x11-apps set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
mohan@mohan:~$
```



➤ **xman** → to see the manual pages





Download Source Code for Packages

- 1) **sudo nano /etc/apt/sources.list** → uncomment all the lines starting deb-src, this will activate the ability of package manager to be able to download source code.

The screenshot shows a terminal window with the command 'mohan@mohan:~\$ sudo nano /etc/apt/sources.list' run. The file contains several commented-out lines starting with '# deb-src'. These lines are uncommented by removing the '#' character. The terminal also shows a note about unsupported software from the universe repository. At the bottom, nano key bindings are listed.

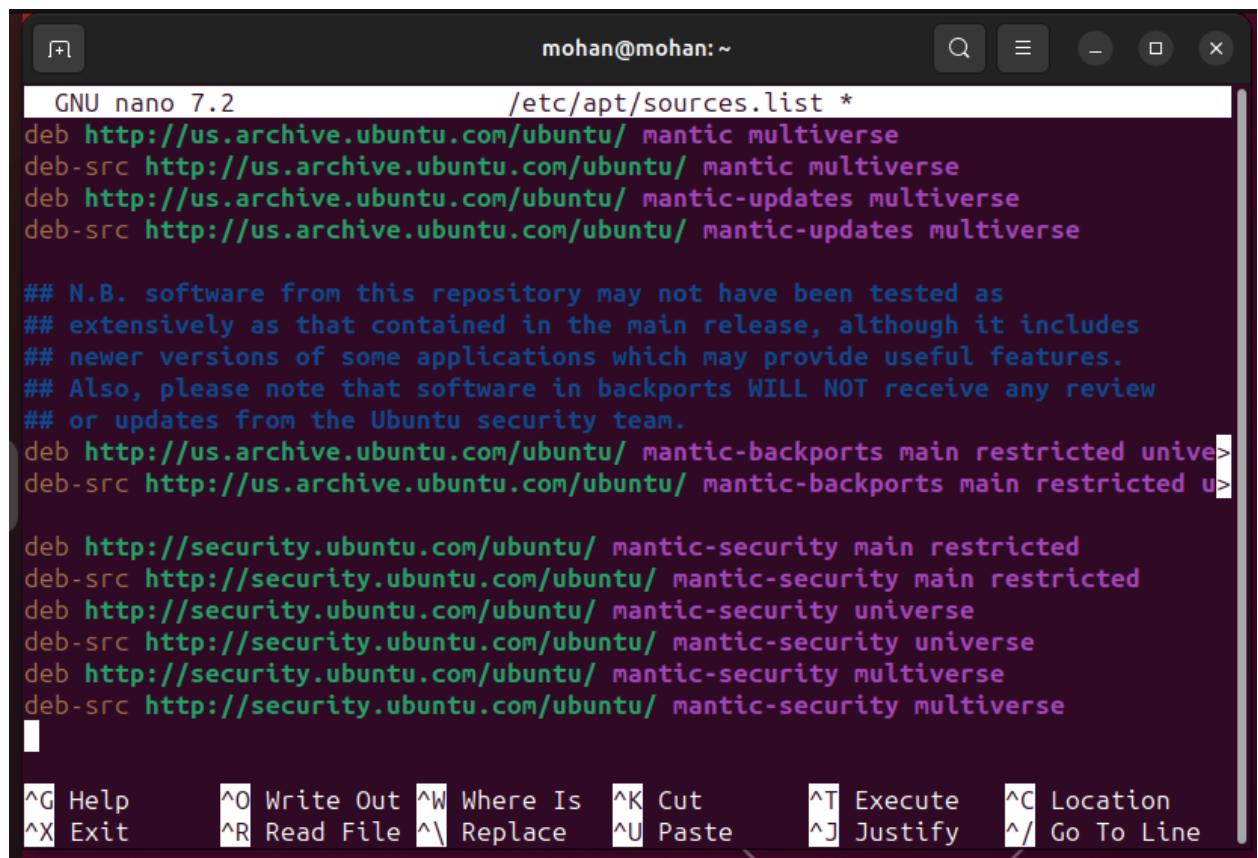
```
mohan@mohan:~$ sudo nano /etc/apt/sources.list
# See http://help.ubuntu.com/community/UpgradeNotes for how to upgrade to
# newer versions of the distribution.
deb http://us.archive.ubuntu.com/ubuntu/ mantic main restricted
# deb-src http://us.archive.ubuntu.com/ubuntu/ mantic main restricted

## Major bug fix updates produced after the final release of the
## distribution.
deb http://us.archive.ubuntu.com/ubuntu/ mantic-updates main restricted
# deb-src http://us.archive.ubuntu.com/ubuntu/ mantic-updates main restricted

## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team. Also, please note that software in universe WILL NOT receive any
## review or updates from the Ubuntu security team.
deb http://us.archive.ubuntu.com/ubuntu/ mantic universe
# deb-src http://us.archive.ubuntu.com/ubuntu/ mantic universe
deb http://us.archive.ubuntu.com/ubuntu/ mantic-updates universe
# deb-src http://us.archive.ubuntu.com/ubuntu/ mantic-updates universe

## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team, and may not be under a free licence. Please satisfy yourself as to
[ Read 42 lines ]
^G Help      ^O Write Out  ^W Where Is  ^K Cut      ^T Execute   ^C Location
^X Exit      ^R Read File  ^\ Replace   ^U Paste    ^J Justify   ^/ Go To Line
```

Uncommented and saved



The screenshot shows a terminal window titled "mohan@mohan:~". The command "nano /etc/apt/sources.list" is running. The content of the file is displayed in green and blue text. It includes repository definitions for "mantic", "mantic-security", and "mantic-updates" from "us.archive.ubuntu.com" and "security.ubuntu.com". A note about N.B. software from the repository is present. The bottom of the screen shows the nano editor's command bar with various keyboard shortcuts.

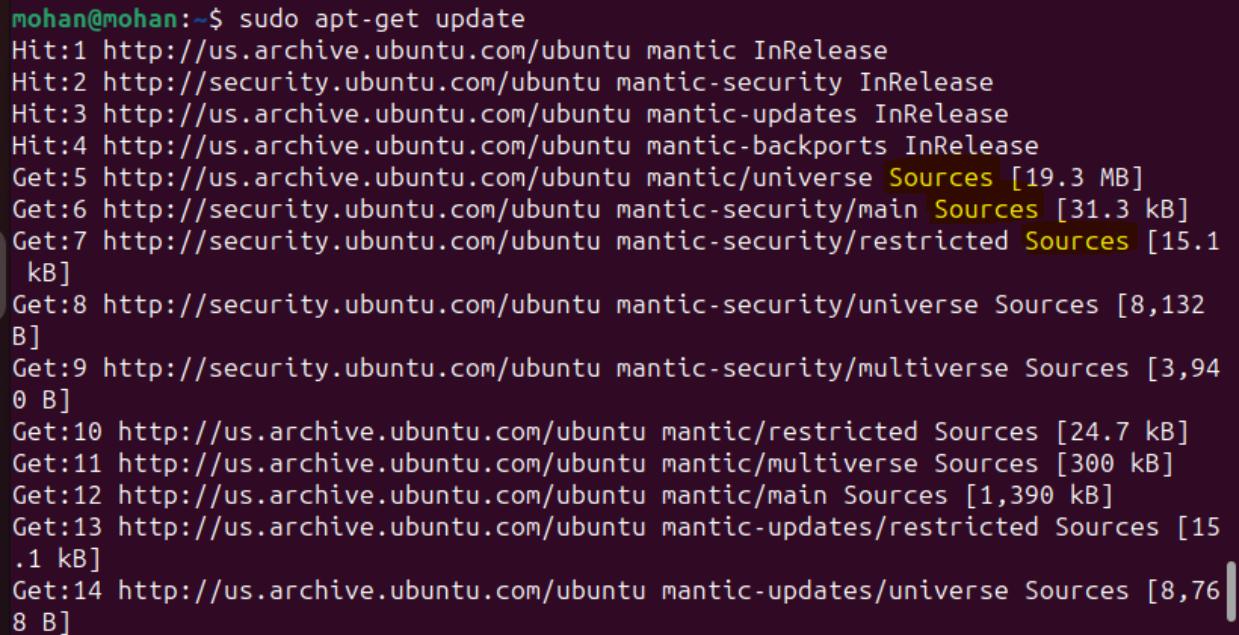
```
GNU nano 7.2 /etc/apt/sources.list *
deb http://us.archive.ubuntu.com/ubuntu/ mantic multiverse
deb-src http://us.archive.ubuntu.com/ubuntu/ mantic multiverse
deb http://us.archive.ubuntu.com/ubuntu/ mantic-updates multiverse
deb-src http://us.archive.ubuntu.com/ubuntu/ mantic-updates multiverse

## N.B. software from this repository may not have been tested as
## extensively as that contained in the main release, although it includes
## newer versions of some applications which may provide useful features.
## Also, please note that software in backports WILL NOT receive any review
## or updates from the Ubuntu security team.
deb http://us.archive.ubuntu.com/ubuntu/ mantic-backports main restricted universe
deb-src http://us.archive.ubuntu.com/ubuntu/ mantic-backports main restricted universe

deb http://security.ubuntu.com/ubuntu/ mantic-security main restricted
deb-src http://security.ubuntu.com/ubuntu/ mantic-security main restricted
deb http://security.ubuntu.com/ubuntu/ mantic-security universe
deb-src http://security.ubuntu.com/ubuntu/ mantic-security universe
deb http://security.ubuntu.com/ubuntu/ mantic-security multiverse
deb-src http://security.ubuntu.com/ubuntu/ mantic-security multiverse

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line
```

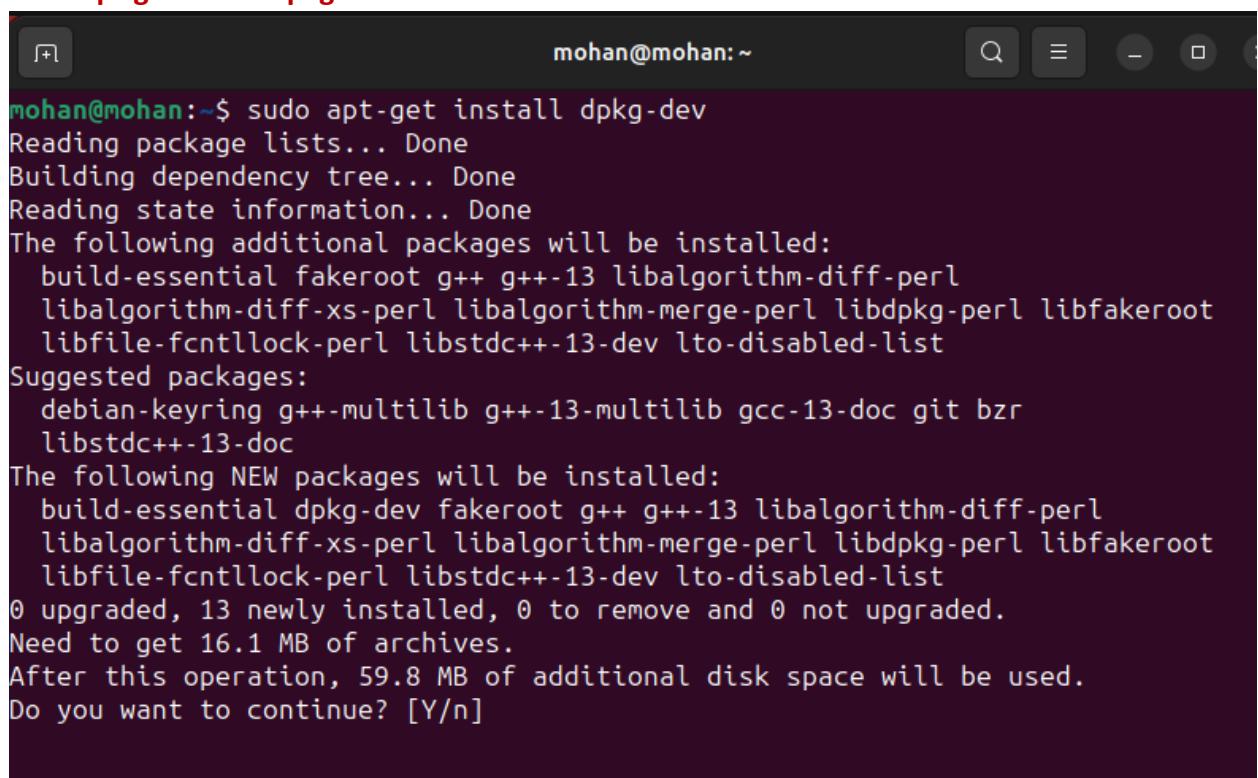
2) sudo apt-get update → to get the sources to our cache



The screenshot shows a terminal window with the command "sudo apt-get update" run. The output lists numerous package sources being downloaded from "us.archive.ubuntu.com" and "security.ubuntu.com" in both "Sources" and "InRelease" formats, with sizes ranging from 8 kB to 19.3 MB.

```
mohan@mohan:~$ sudo apt-get update
Hit:1 http://us.archive.ubuntu.com/ubuntu mantic InRelease
Hit:2 http://security.ubuntu.com/ubuntu mantic-security InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu mantic-updates InRelease
Hit:4 http://us.archive.ubuntu.com/ubuntu mantic-backports InRelease
Get:5 http://us.archive.ubuntu.com/ubuntu mantic/universe Sources [19.3 MB]
Get:6 http://security.ubuntu.com/ubuntu mantic-security/main Sources [31.3 kB]
Get:7 http://security.ubuntu.com/ubuntu mantic-security/restricted Sources [15.1 kB]
Get:8 http://security.ubuntu.com/ubuntu mantic-security/universe Sources [8,132 B]
Get:9 http://security.ubuntu.com/ubuntu mantic-security/multiverse Sources [3,940 B]
Get:10 http://us.archive.ubuntu.com/ubuntu mantic/restricted Sources [24.7 kB]
Get:11 http://us.archive.ubuntu.com/ubuntu mantic/multiverse Sources [300 kB]
Get:12 http://us.archive.ubuntu.com/ubuntu mantic/main Sources [1,390 kB]
Get:13 http://us.archive.ubuntu.com/ubuntu mantic-updates/restricted Sources [15.1 kB]
Get:14 http://us.archive.ubuntu.com/ubuntu mantic-updates/universe Sources [8,768 B]
```

3) sudo apt-get install dpkg-dev

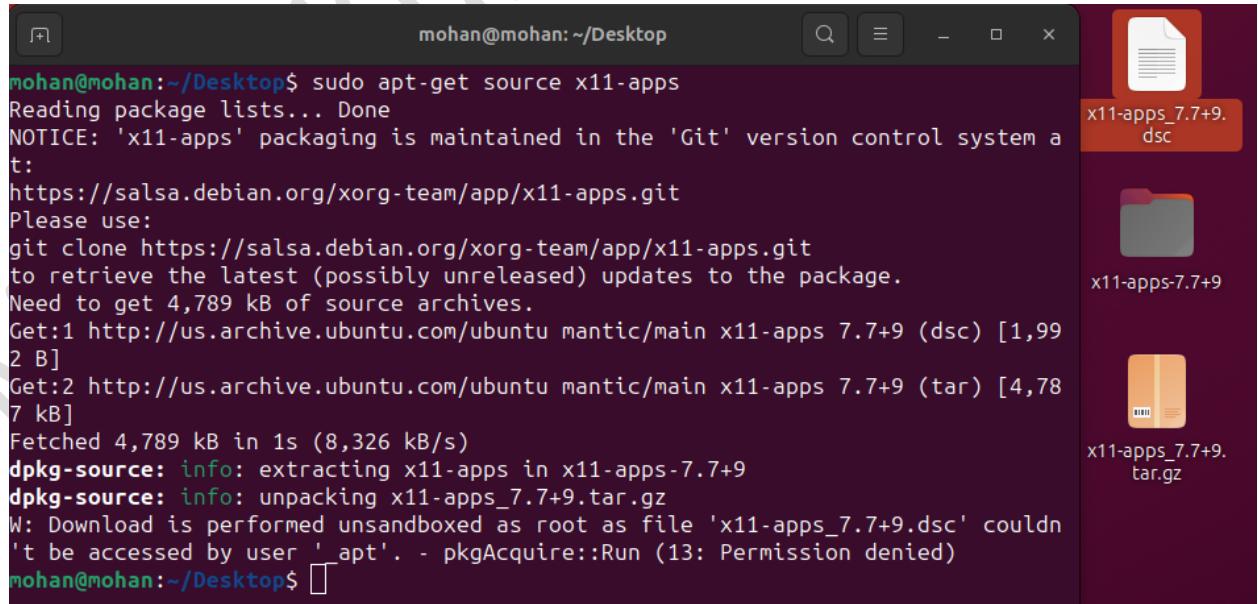


```
mohan@mohan:~$ sudo apt-get install dpkg-dev
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  build-essential fakeroot g++ g++-13 libalgorithm-diff-perl
  libalgorithm-diff-xs-perl libalgorithm-merge-perl libdpkg-perl libfakeroot
  libfile-fcntllock-perl libstdc++-13-dev lto-disabled-list
Suggested packages:
  debian-keyring g++-multilib g++-13-multilib gcc-13-doc git bzr
  libstdc++-13-doc
The following NEW packages will be installed:
  build-essential dpkg-dev fakeroot g++ g++-13 libalgorithm-diff-perl
  libalgorithm-diff-xs-perl libalgorithm-merge-perl libdpkg-perl libfakeroot
  libfile-fcntllock-perl libstdc++-13-dev lto-disabled-list
0 upgraded, 13 newly installed, 0 to remove and 0 not upgraded.
Need to get 16.1 MB of archives.
After this operation, 59.8 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

- 4) Now we are good at downloading source code. Cd to the folder where you want to save the source code and execute the below command.

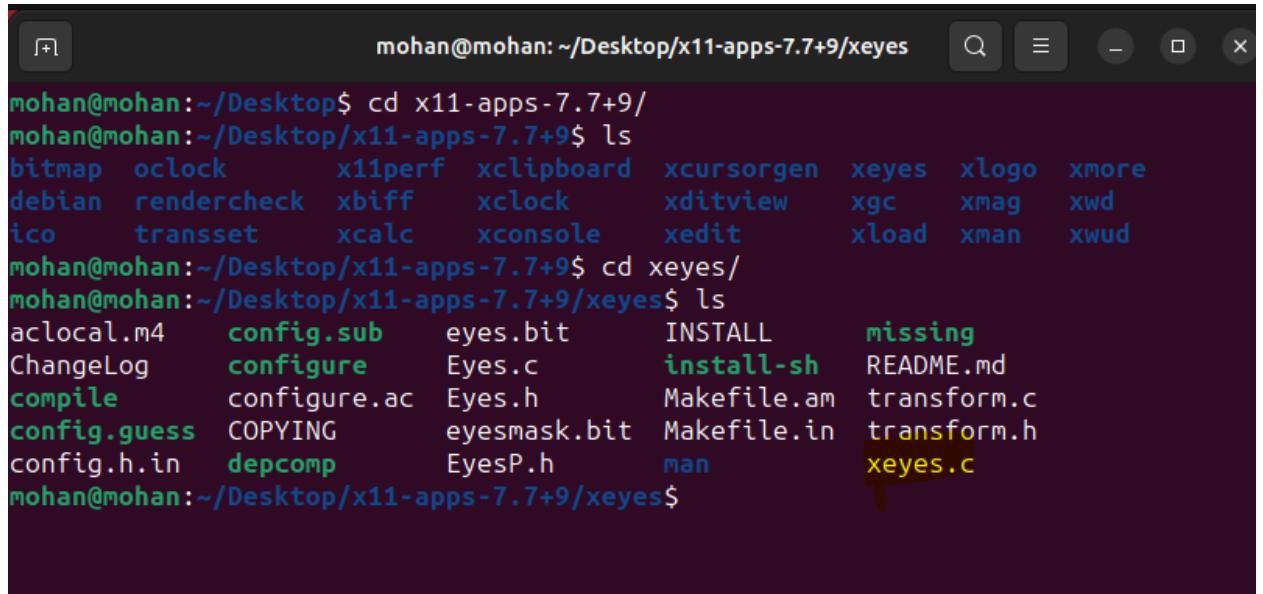
sudo apt-get source <package_name>

sudo apt-get source x11-apps → to get source code for x-apps



```
mohan@mohan:~/Desktop$ sudo apt-get source x11-apps
Reading package lists... Done
NOTICE: 'x11-apps' packaging is maintained in the 'Git' version control system at:
https://salsa.debian.org/xorg-team/app/x11-apps.git
Please use:
git clone https://salsa.debian.org/xorg-team/app/x11-apps.git
to retrieve the latest (possibly unreleased) updates to the package.
Need to get 4,789 kB of source archives.
Get:1 http://us.archive.ubuntu.com/ubuntu mantic/main x11-apps 7.7+9 (dsc) [1,992 B]
Get:2 http://us.archive.ubuntu.com/ubuntu mantic/main x11-apps 7.7+9 (tar) [4,787 kB]
Fetched 4,789 kB in 1s (8,326 kB/s)
dpkg-source: info: extracting x11-apps in x11-apps-7.7+9
dpkg-source: info: unpacking x11-apps_7.7+9.tar.gz
W: Download is performed unsandboxed as root as file 'x11-apps_7.7+9.dsc' couldn't be accessed by user '_apt'. - pkgAcquire::Run (13: Permission denied)
mohan@mohan:~/Desktop$
```

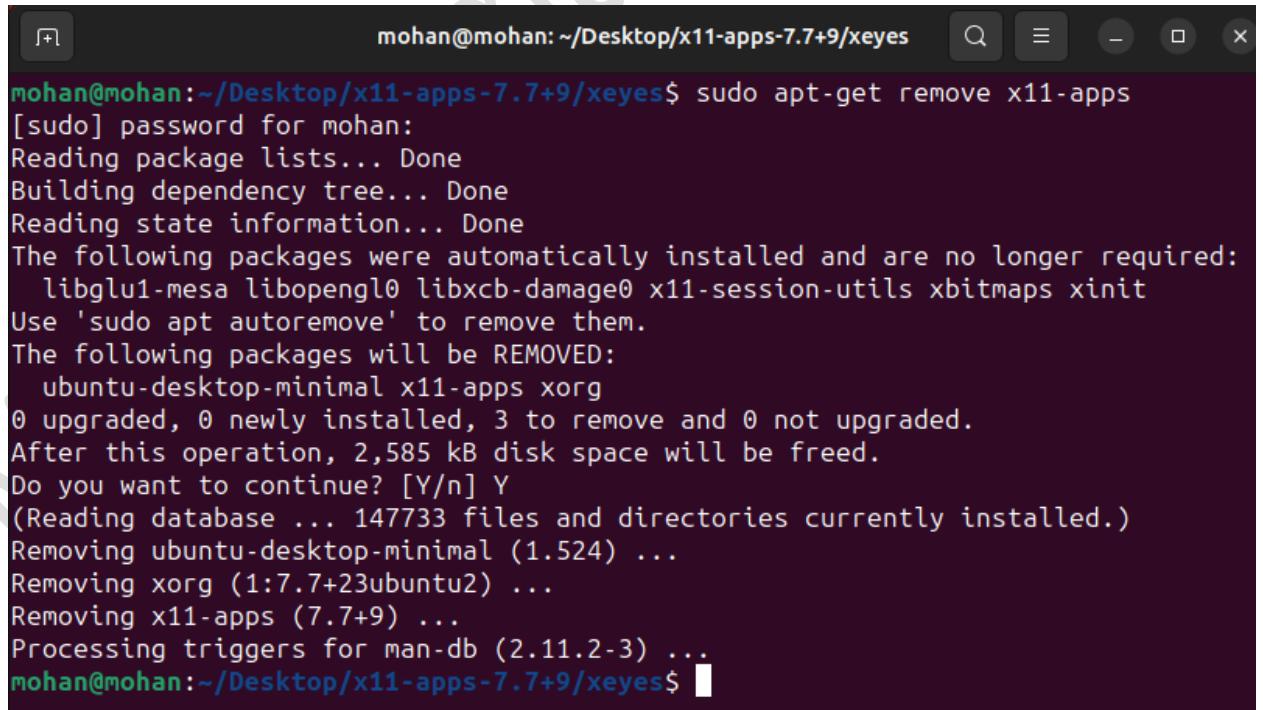
The terminal shows the command being run and its output. To the right of the terminal, there is a file browser window showing three files: 'x11-apps_7.7+9.dsc' (a deb source control file), 'x11-apps-7.7+9' (the directory containing the extracted source code), and 'x11-apps_7.7+9.tar.gz' (the tarball of the source code).



```
mohan@mohan:~/Desktop$ cd x11-apps-7.7+9/
mohan@mohan:~/Desktop/x11-apps-7.7+9$ ls
bitmap o'clock x11perf xclipboard xcursorgen xeyes xlogo xmore
debian rendercheck xbiff xclock xditview xgc xmag xwd
ico transset xcalc xconsole xedit xload xman xwud
mohan@mohan:~/Desktop/x11-apps-7.7+9$ cd xeyes/
mohan@mohan:~/Desktop/x11-apps-7.7+9/xeyes$ ls
aclocal.m4 config.sub eyes.bit INSTALL missing
ChangeLog configure Eyes.c install-sh README.md
compile configure.ac Eyes.h Makefile.am transform.c
config.guess COPYING eyesmask.bit Makefile.in transform.h
config.h.in depcomp EyesP.h man xeyes.c
mohan@mohan:~/Desktop/x11-apps-7.7+9/xeyes$
```

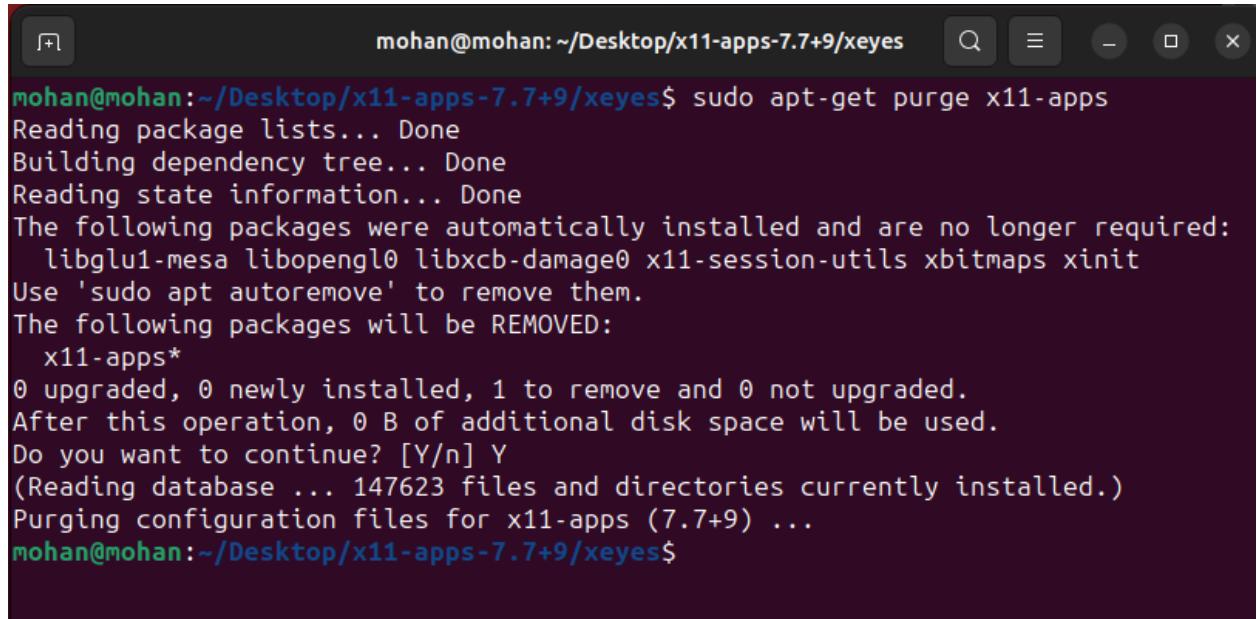
Uninstalling Packages

- **sudo apt-get remove <package_name>** → this is just to uninstall the package, but it will not remove any configuration files that may occupy space on your system unnecessarily. This is preferred only when you are going to reinstall the package later.



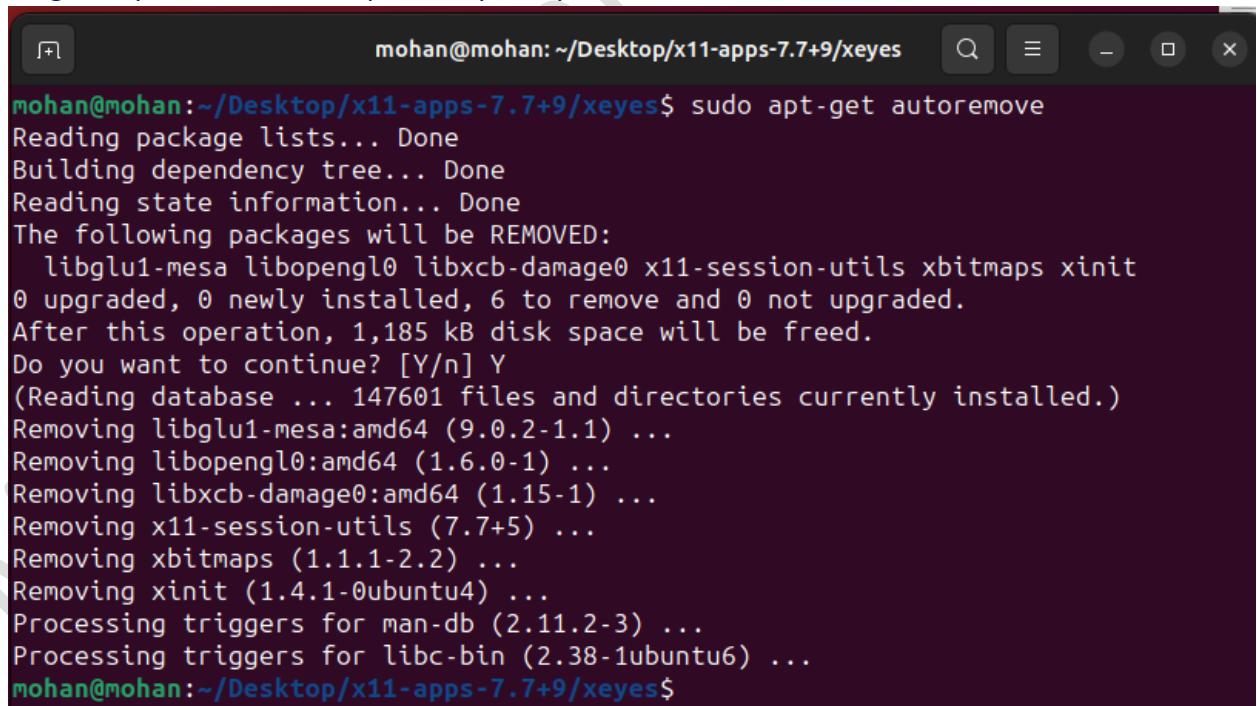
```
mohan@mohan:~/Desktop/x11-apps-7.7+9/xeyes$ sudo apt-get remove x11-apps
[sudo] password for mohan:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libglu1-mesa libopengl0 libxcb-damage0 x11-session-utils xbitmaps xinit
Use 'sudo apt autoremove' to remove them.
The following packages will be REMOVED:
  ubuntu-desktop-minimal x11-apps xorg
0 upgraded, 0 newly installed, 3 to remove and 0 not upgraded.
After this operation, 2,585 kB disk space will be freed.
Do you want to continue? [Y/n] Y
(Reading database ... 147733 files and directories currently installed.)
Removing ubuntu-desktop-minimal (1.524) ...
Removing xorg (1:7.7+23ubuntu2) ...
Removing x11-apps (7.7+9) ...
Processing triggers for man-db (2.11.2-3) ...
mohan@mohan:~/Desktop/x11-apps-7.7+9/xeyes$
```

- **sudo apt-get purge <package_name>** → this will remove package and configuration files as well. Also, use this when you want to uninstall a package.



```
mohan@mohan:~/Desktop/x11-apps-7.7+9/xeyes$ sudo apt-get purge x11-apps
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libglu1-mesa libopengl0 libxcb-damage0 x11-session-utils xbitmaps xinit
Use 'sudo apt autoremove' to remove them.
The following packages will be REMOVED:
  x11-apps*
0 upgraded, 0 newly installed, 1 to remove and 0 not upgraded.
After this operation, 0 B of additional disk space will be used.
Do you want to continue? [Y/n] Y
(Reading database ... 147623 files and directories currently installed.)
Purging configuration files for x11-apps (7.7+9) ...
mohan@mohan:~/Desktop/x11-apps-7.7+9/xeyes$
```

- **sudo apt-get autoremove** → this will remove any dangling dependencies that are no longer required that saves space on your system.



```
mohan@mohan:~/Desktop/x11-apps-7.7+9/xeyes$ sudo apt-get autoremove
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages will be REMOVED:
  libglu1-mesa libopengl0 libxcb-damage0 x11-session-utils xbitmaps xinit
0 upgraded, 0 newly installed, 6 to remove and 0 not upgraded.
After this operation, 1,185 kB disk space will be freed.
Do you want to continue? [Y/n] Y
(Reading database ... 147601 files and directories currently installed.)
Removing libglu1-mesa:amd64 (9.0.2-1.1) ...
Removing libopengl0:amd64 (1.6.0-1) ...
Removing libxcb-damage0:amd64 (1.15-1) ...
Removing x11-session-utils (7.7+5) ...
Removing xbitmaps (1.1.1-2.2) ...
Removing xinit (1.4.1-0ubuntu4) ...
Processing triggers for man-db (2.11.2-3) ...
Processing triggers for libc-bin (2.38-1ubuntu6) ...
mohan@mohan:~/Desktop/x11-apps-7.7+9/xeyes$
```

- When a software is installed, it downloads a compressed file and then extract the files and installs which is a two-step process. If we remove the compressed files, we can save massive amounts of space. cd to /var/cache/apt/archives.

sudo apt-get clean → this command will clean all these compressed files

```
mohan@mohan:~/Desktop/x11-apps-7.7+9/xeyes$ cd /var/cache/apt/archives/
mohan@mohan:/var/cache/apt/archives$ ls
binutils_2.41-5ubuntu1_amd64.deb
binutils-common_2.41-5ubuntu1_amd64.deb
binutils-x86-64-linux-gnu_2.41-5ubuntu1_amd64.deb
build-essential_12.10ubuntu1_amd64.deb
distro-info-data_0.58ubuntu0.1_all.deb
dpkg-dev_1.22.0ubuntu1_all.deb
fakeroot_1.32.1-1_amd64.deb
firmware-sof-signed_2.2.6-1ubuntu1.1_all.deb
g++-13_13.2.0-4ubuntu3_amd64.deb
g++_4%3a13.2.0-1ubuntu1_amd64.deb
gcc-13_13.2.0-4ubuntu3_amd64.deb
gcc_4%3a13.2.0-1ubuntu1_amd64.deb
gir1.2-mutter-13_45.0-3ubuntu3.1_amd64.deb
gir1.2-nm-1.0_1.44.2-1ubuntu1.2_amd64.deb
intel-microcode_3.20231114.0ubuntu0.23.10.1_amd64.deb
libalgorithm-diff-perl_1.201-1_all.deb
libalgorithm-diff-xs-perl_0.04-8_amd64.deb
libalgorithm-merge-perl_0.08-5_all.deb
libasan8_13.2.0-4ubuntu3_amd64.deb
libbinutils_2.41-5ubuntu1_amd64.deb
libc6-dev_2.38-1ubuntu6_amd64.deb
```

```
mohan@mohan:/var/cache/apt/archives$ sudo apt-get clean
mohan@mohan:/var/cache/apt/archives$ cd /var/cache/apt/archives/
mohan@mohan:/var/cache/apt/archives$ ls
lock partial
mohan@mohan:/var/cache/apt/archives$ 
```

- **sudo apt-get autoclean** → will checks in and cleans the compressed files which can no-longer be downloaded from repositories

```
mohan@mohan:~$ sudo apt-get autoclean
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
mohan@mohan:~$ 
```

Happy Learning

Mohan Kumar Pulibanti