50 Most Used Linux Commands

Suppose, you've started to use **Linux** recently, and you're quite new to the command line interface. But one thing is certain: if you want to learn about some frequently used **Linux** commands and practice them on your own then you've found the right place. Here, we've discussed the **50 most used Linux commands** with a brief description and practical examples.

List of 50 Most Used Linux Commands

1. sudo command

Sudo can be referred as the supreme command. It is the abbreviation for **"S**uper **U**ser **DO".** It allows a user to act as a **superuser** and run commands accordingly. One can run certain commands prefixed by **sudo** with boosted rights. It is considered analogous to the "**run as administrator**" process of Windows.

Synopsis

```
sudo -h | -K | -k | -V
sudo -v [-ABknS] [-g group] [-h host] [-p prompt] [-u user]
sudo -l [-ABknS] [-g group] [-h host] [-p prompt] [-U user]
     [-u user] [command]
sudo [-ABbEHnPS] [-C num] [-D directory] [-g group]
     [-h host] [-p prompt] [-R directory] [-r role]
     [-t type] [-T timeout] [-u user] [VAR=value] [-i | -s]
     [command]
sudoedit [-ABknS] [-C num] [-D directory] [-g group]
     [-h host] [-p prompt] [-R directory] [-r role]
     [-t type] [-T timeout] [-u user] file ...
```

Useful Options

-D directory, --chdir=directory (executes the command in the specific directory)
-e (edits one or multiple files instead of executing commands)
-I (runs specific commands as the root user)
-u user, --user=user (executes the command as a user other than the specific default user)

Example

```
softeko@ubuntu:~/Desktop> apt install XYZ
E: Could not open lock file /var/lib/dpkg/lock-frontend - open (13:
 Permission denied)
E: Unable to acquire the dpkg frontend lock (/var/lib/dpkg/lock-fro
ntend), are you root?
```

Any general user cannot install any packages on the machine. However, with **sudo** prefixed with the command, the user can execute his/her task by providing his/her password.

2. pwd command

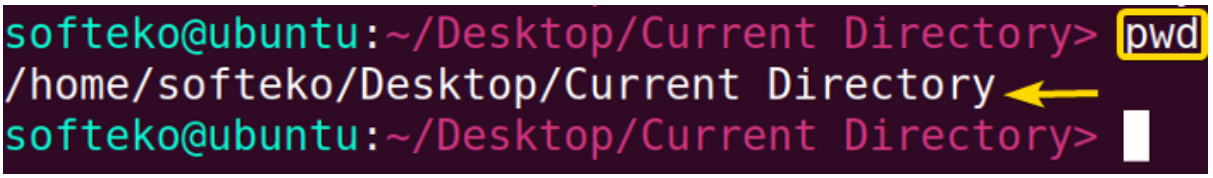
Pwd is the abbreviation for **P**rint **W**orking **D**irectory. As the name suggests, it prints the name of the **current/working** directory all the way beginning from the root(/) directory.

Synopsis

Useful Options

-L, --logical (Even as it carries symlinks, PWD utilizes from the environment.)
-P, --physical (avoid the symlinks)
When no option is mentioned, it is assumed that option -P is being used.

Example



Generally, Terminal prompts have a complete directory in the name. Otherwise, pwd becomes a handy command to get insights about the current working directory.

3. cd command

Change **D**irectory(**cd**) allows one to change one's current directory to the desired directory within the **terminal**.

Synopsis

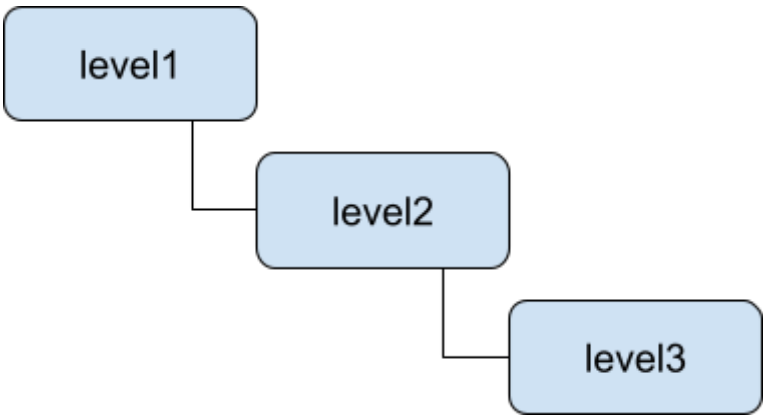
```
softeko@ubuntu:~/Desktop/Current Directory> type cd
cd is a shell builtin
softeko@ubuntu:~/Desktop/Current Directory> help cd
cd: cd [-L|[-P [-e]] [-@]] [dir]
    Change the shell working directory.

    Change the current directory to DIR.  The default DIR is the va
lue of the
    HOME shell variable.

    The variable CDPATH defines the search path for the directory c
ontaining
    DIR.  Alternative directory names in CDPATH are separated by a
colon (:).
```

Note: **cd** is a shell built-in command, it doesn't have a dedicated man(manual) page. However, you can get help using command help!

Useful Options

cd ~[username] — change the directory to the home directory of the specified user.
cd .. — changes directory one directory up the current directory.
cd - — changes the directory to the previously changed directory.

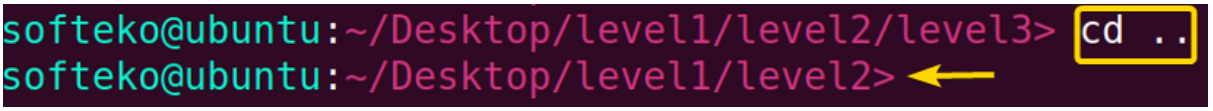
Example



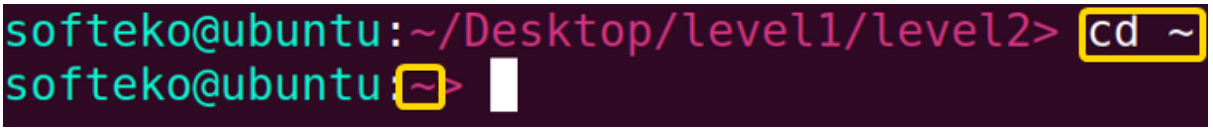
In our Desktop directory, we have a 3-level nested directory. we want to leap forward to the level 3 directory by jumping one level at a time.

```
softeko@ubuntu:~/Desktop> ls
'Current Directory'   FIFA_World_Cup_2022   file2   level1
 Examples             file1                file3   Locate
softeko@ubuntu:~/Desktop> cd level1/
softeko@ubuntu:~/Desktop/level1> cd level2/
softeko@ubuntu:~/Desktop/level1/level2> cd level3/
softeko@ubuntu ~/Desktop/level1/level2/level3>
```

We have changed our mind, now we want to be at level2. Therefore, we just need to go one directory backward which can easily be done by executing the command **cd ..**



Lastly, we have come to the conclusion that we want to be in the home(**~**) directory. This can be simply done by executing **cd ~** (tilde represents home directory).



4. ls command

Lists the contents, both files and subdirectories of the current directory by default. It is one of the most used commands, as one can view the contents of a directory without exiting the terminal and perform their desired tasks on the specific contents.

Synopsis

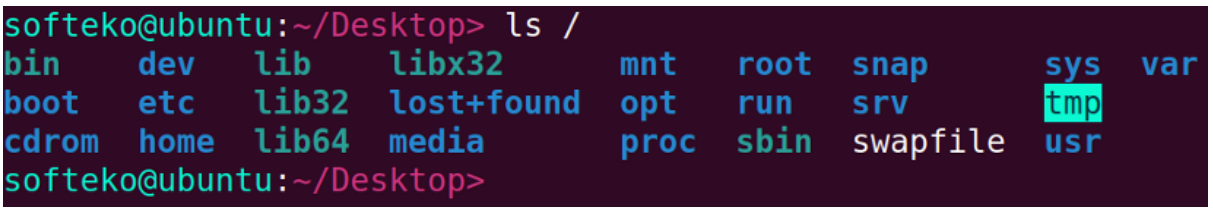
```
ls [OPTION]... [FILE]...
```

Useful Options

-a — doesn't ignore the hidden files (files named with **.**(dot) at the beginning).
-h — print sizes in human-readable forms.
-l — lists in a long form.
-S — Sorts according to file size, largest first.

Example

After being at the **root(/)** directory, if we run the **ls** command we can view the contents of the root directory.



5. cat command

Prints the contents of the file specified. Generally, **cat** (con**cat**enates) reads the contents of the files fed to its arguments and prints them serially on the terminal.

Synopsis

```
cat [OPTION]... [FILE]...
```

Useful Options

-n, --number — Displays line numbers when utilized.

Example

We can display the contents of a file simply by using the **cat** command followed by the file name.

```
softeko@ubuntu:~> cat Greetings
Welcome to LinuxSimply. One go-to-place to find
solutions of your problems related to Linux. Good
Luck! to your upcoming journey!!!
softeko@ubuntu:~>
```

6. mv command

Mv is the abbreviation for **mov**e. As the name suggests it moves things from one place to another place. **mv** moves one or multiple files to the specified destination directory. If the directory doesn't exist it just renames the files. **mv** can also be used to move directories and their contents.

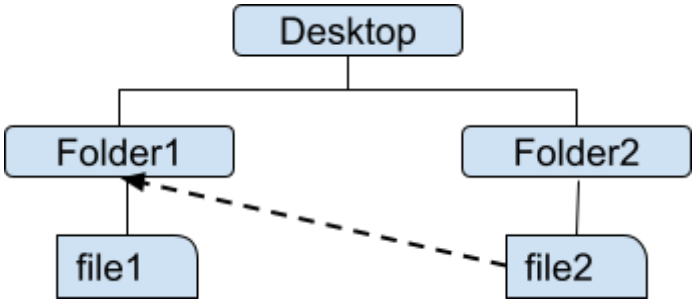
Synopsis

```
mv [OPTION]... [-T] SOURCE DEST
mv [OPTION]... SOURCE... DIRECTORY
mv [OPTION]... -t DIRECTORY SOURCE...
```

Useful Options

-i, --interactive (Displays interactive prompt before completing the modification)
-t, --target-directory=DIRECTORY (Moves every specified file to the targeted DIRECTORY)
-v, --verbose (Prints message of what is being performed.)

Example



In the desktop directory, there are two subirectories named Folder1 & Folder2 which respectively contains two files named file1 & file2. Now lets **mov**e file2 to folder1.

```
softeko@ubuntu:~/Desktop> ls
'Current Directory'     file1   Folder1   Locate
 Examples               file2   Folder2
 FIFA_World_Cup_2022    file3   level1
softeko@ubuntu:~/Desktop> ls Folder1/
file1
softeko@ubuntu:~/Desktop> ls Folder2/
file2
softeko@ubuntu:~/Desktop> mv Folder2/file2 Folder1/
softeko@ubuntu:~/Desktop> ls Folder2/
softeko@ubuntu:~/Desktop> ls Folder1/
file1  file2
softeko@ubuntu:~/Desktop>
```

7. cp command

Cp is the abbreviation for **co**p**y**. As the name suggests it copies things from one place to another place. **cp** copies one or multiple files to the specified destination directory. If the directory doesn't exist it just renames the files. **cp** can also be used to copy directories and their contents.

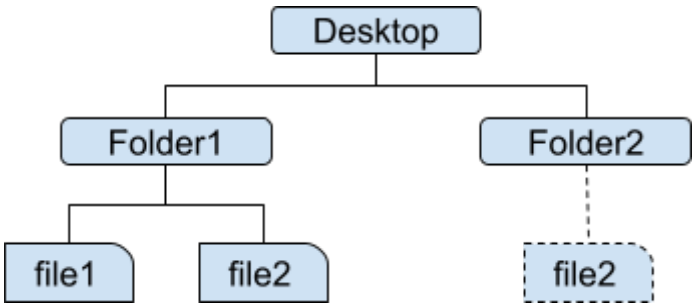
Synopsis

```
cp [OPTION]... [-T] SOURCE DEST
cp [OPTION]... SOURCE... DIRECTORY
cp [OPTION]... -t DIRECTORY SOURCE...
```

Useful Options

-i, --interactive (Displays interactive prompt before completing the modification)
-R, -r, --recursive (Copies the directories as well as its contents recursively)
-v, --verbose (Prints message of what is being performed.)

Example



In the desktop directory, there are two subdirectories named Folder1 & Folder2. In folder1 there is a file named file2. Let's make a **co**py of file2 in the Folder2 directory.

```
softeko@ubuntu:~/Desktop> ls
'Current Directory'     file1   Folder1   Locate
 Examples               file2   Folder2
 FIFA_World_Cup_2022    file3   level1
softeko@ubuntu:~/Desktop> ls Folder1/
file1  file2
softeko@ubuntu:~/Desktop> ls Folder2/
softeko@ubuntu:~/Desktop> cp Folder1/file2 Folder2/
softeko@ubuntu:~/Desktop> ls Folder2/
file2
softeko@ubuntu:~/Desktop>
```

8. rm command

Rm is the abbreviation for **rem**ove. As the name suggests it removes things and the removal is **permanent**, so be cautious while using it. **rm** can also be used to remove directories and their contents permanently.

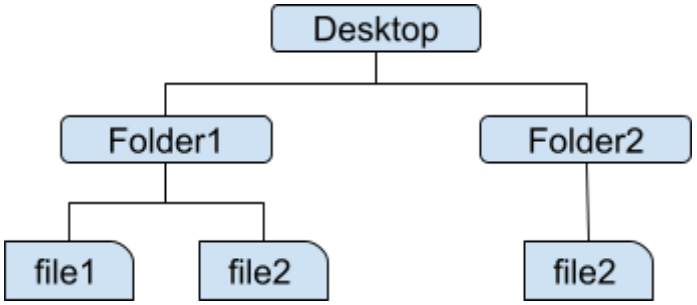
Synopsis

```
rm [OPTION]... [FILE]...
```

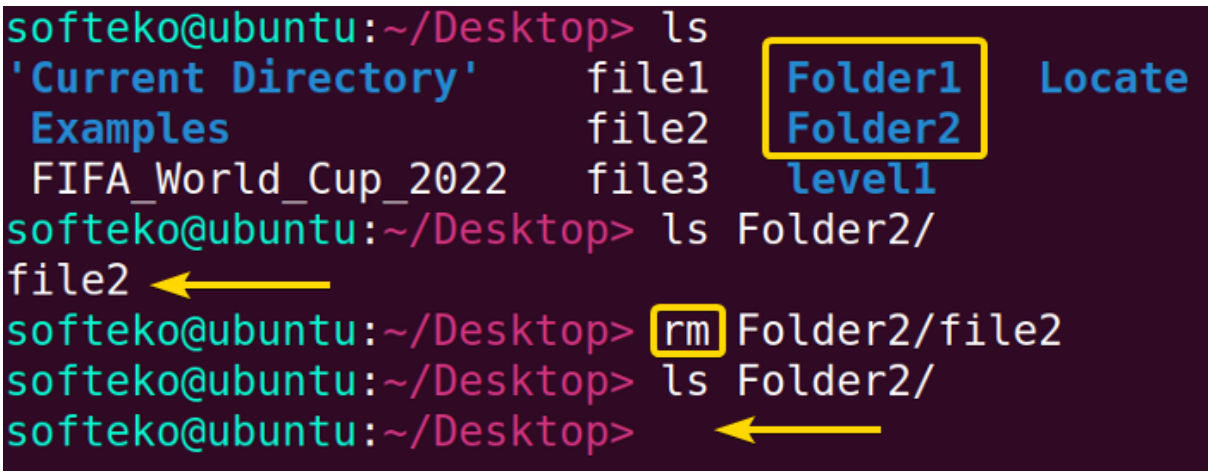
Useful Options

-i (Displays interactive prompt before completing the deletion each time)
-I (Less intuitive than **-i**, Only shows prompt while deleting 3 or more files or deleting recursively.)
-d, --dir (Removes the empty directories)
-R, -r, --recursive (Removes the directories as well as its contents recursively)
-v, --verbose (Prints message of what is being performed.)

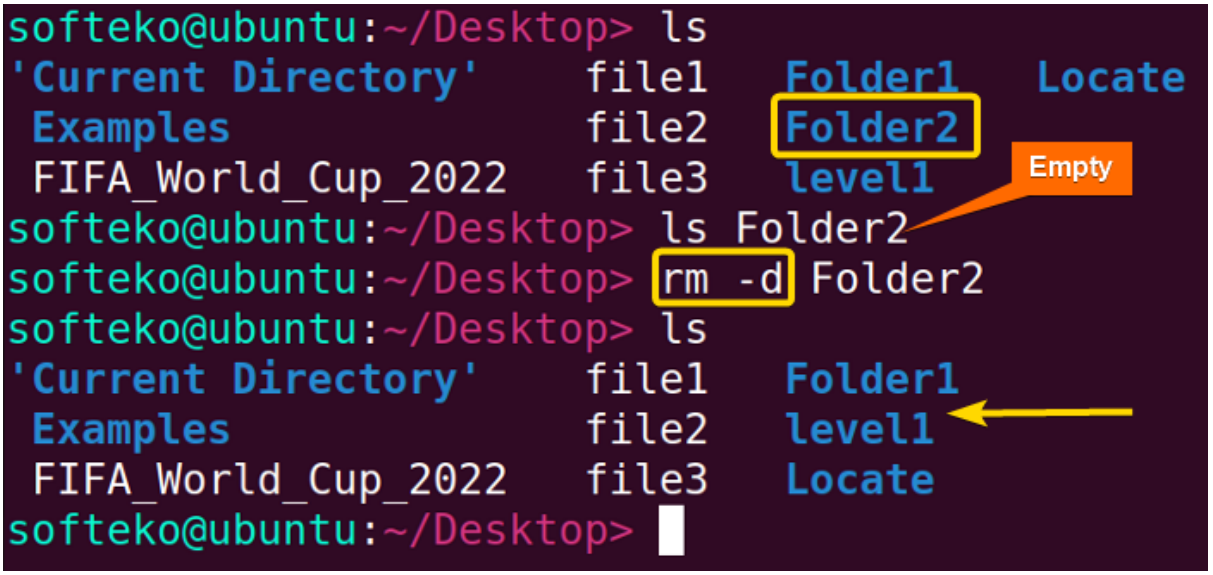
Example



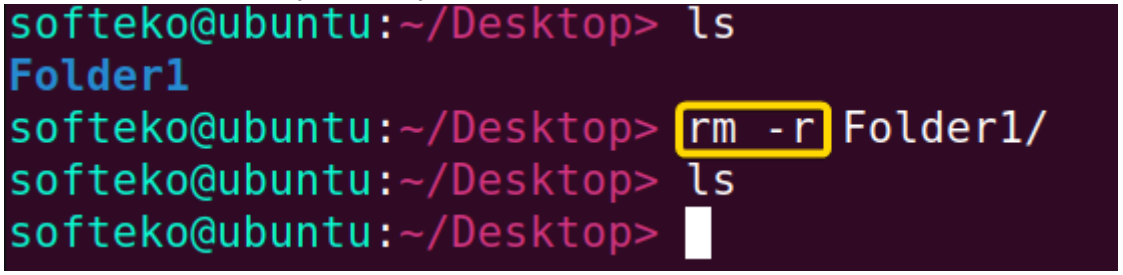
In the desktop directory, there are two subdirectories named Folder1 & Folder2. Folder2 contains a file named file2. let's **rem**ove file2.



Now, Folder2 is empty let's remove this directory. To remove an empty directory option **-d** has to be used.



Finally, we have decided that Folder1 is also of no use to us. Let's remove this directory. However, it is not an empty directory so **-d** will not work. Instead, we have to use **-r**.



Disclaimer: You cannot retrieve anything after removing it through the **rm** command. So, better be careful with **rm**.

9. touch command

The **touch** command allows us to update a file's access or modification time. However, if the file doesn't exist we can **create** that file. This ability to create files makes the **touch** command one of the most useful commands.

Synopsis

```
touch [OPTION]... FILE...
```

Useful Options

-a (changes only the access time)
-m (changes only the modification time)

Example

```
softeko@ubuntu:~/Desktop> ls
softeko@ubuntu:~/Desktop> touch file1 file2 file3
softeko@ubuntu:~/Desktop> ls
file1  file2  file3
softeko@ubuntu:~/Desktop>
```

We created 3 files in our current directory, Desktop named file1, file2 and file3 with the help of the **touch** command.

10. mkdir command

mkdir is the abbreviation for **m**a**k**e **dir**ectory. As the name suggests the **mkdir** command can be used to create one or more directories.

Synopsis

```
mkdir [OPTION]... DIRECTORY...
```

Useful Options

-p - -parents (Creates the necessary parent directories if required)
-v, --verbose (Prints message of what is being performed.)

Example

Let's make a directory named 'Examples' in our desktop directory which will contain one subdirectory named 'NoExamples'.

```
softeko@ubuntu:~/Desktop> ls
file1  file2  file3
softeko@ubuntu:~/Desktop> mkdir -p Examples/NoExamples
softeko@ubuntu:~/Desktop> ls
Examples  file1  file2  file3
softeko@ubuntu:~/Desktop> ls Examples/
NoExamples
softeko@ubuntu:~/Desktop>
```

Note: Option **-p** is used to create the necessary parent directory, in this case, which is 'Examples'.

11. locate command

The **locate** command performs the search operation from an **existing database** and prints the results with the exact directory path.

Synopsis

```
plocate [OPTION]...  PATTERN...
```

Useful Options

-c (Does not print any matches rather total occurring number)
-e (Only prints the existing match)
-i (Ignores the sensitivity of cases)

Example

We have a file named Greetings in our '/home' directory. After changing our current directory from '/home' to '/Desktop/Locate' we operated the **locate** command.

```
softeko@ubuntu:~> ls
Desktop    Documents  Greetings  Pictures  snap       Videos
Directory  Downloads  Music      Public    Templates
softeko@ubuntu:~> cd Desktop/Locate/
softeko@ubuntu:~/Desktop/Locate> locate Greetings
/home/softeko/Greetings
softeko@ubuntu:~/Desktop/Locate>
```

12. find command

The **find** command searches in real-time not like the **locate** command from an existing file.

Synopsis

```
find  [-H]  [-L]  [-P] [-D debugopts] [-Olevel] [starting-
point...] [expression]
```

Useful Options

-type d/f (**d**(limits the search to only directories), **f**(limits the search to only files)).
-size +n,**n**,**-n** (Finds for a specific size **n**.)
-name pattern (Will search for the given pattern)
-exec (can be used to perform our customized tasks on the matches.)

Example

We will only limit the find operation to directory type.

```
softeko@ubuntu:~/Desktop> ls
Examples  file1  file2  file3  Locate
softeko@ubuntu:~/Desktop> find  -type d
.
./Locate
./Examples
./Examples/NoExamples
softeko@ubuntu:~/Desktop>
```

13. grep command

The **grep** command can be used to search patterns in specific files or each file. Then prints the entire line containing the match.

Synopsis

```
grep [OPTION...] PATTERNS [FILE...]
grep [OPTION...] -e PATTERNS ... [FILE...]
grep [OPTION...] -f PATTERN_FILE ... [FILE...]
```

Useful Options

-c (Does not print any matches rather total occurring number)
-i (Ignores the sensitivity of cases)
-w (search for a whole word)

Example

We have a text file named “FIFA_World_Cup_2022” in our desktop directory. Let’s search for ‘FIFA’ in the file.

```
softeko@ubuntu:~/Desktop> ls
Examples  FIFA_World_Cup_2022  file1  file2  file3  Locate
softeko@ubuntu:~/Desktop> grep FIFA FIFA_World_Cup_2022
The 2022 FIFA World Cup is an international association
of FIFA's member associations, and the 22nd FIFA World Cup.
4–2 in the 2018 FIFA World Cup final.
of bribery for hosting rights and wider FIFA
FIFA President Sepp Blatter has twice said that giving
Qatar hosting rights was a "mistake".[8][9] Current FIFA
softeko@ubuntu:~/Desktop>
```

Now, if we want to just know about the occurrence number, we have to use the **-c** option.

```
softeko@ubuntu:~/Desktop> grep -c FIFA FIFA_World_Cup_2022
6
```

14. head command

The **head** command prints the **first**(by default 10 lines) few lines of a file.

Synopsis

```
head [OPTION]... [FILE]...
```

Useful Options

-n (Prints the first n lines.)
-v, --verbose (Prints message of what is being performed.)

Example

We want to print the first 5 lines of the "FIFA_World_Cup_2022" text file. It can be simply done using the **head** command with **-n5** option

```
softeko@ubuntu:~/Desktop> ls
Examples  FIFA_World_Cup_2022  file1  file2  file3  Locate
softeko@ubuntu:~/Desktop> head -n5 FIFA_World_Cup_2022
The 2022 FIFA World Cup is an international association
football tournament contested by the men's national teams
of FIFA's member associations, and the 22nd FIFA World Cup.
The event is taking place in Qatar from 20 November to 18
December 2022. This is the first World Cup to be held in
softeko@ubuntu:~/Desktop>
```

15. tail command

The **tail** command prints the **last**(by default 10 lines) few lines of a file.

Synopsis

```
tail [OPTION]... [FILE]...
```

Useful Options

-n (Prints the last n lines.)
-v, --verbose (Prints message of what is being performed.)

Example

We want to print the first 5 lines of the "FIFA_World_Cup_2022" text file. It can be simply done using the **tail** command with **-n3** option

```
softeko@ubuntu:~/Desktop> ls
Examples  FIFA_World_Cup_2022  file1  file2  file3  Locate
softeko@ubuntu:~/Desktop> tail -n3 FIFA_World_Cup_2022
liberal democracies.[6][7] Others noted it as
representative of declining Western influence in
association football and geopolitics.[5][11]
softeko@ubuntu:~/Desktop>
```

16. man command

You are whether a beginner or a professional Linux CLI(command line interface) user, the command you will interact with for the most is certainly the **man** command. The **man** command enables the user to learn more about a specific command i.e. syntax, options and arguments of that command.

Synopsis

```
man [man options] [[section] page ...] ...
man -k [apropos options] regexp ...
man -K [man options] [section] term ...
man -f [whatis options] page ...
man -l [man options] file ...
man -w|-W [man options] page ...
```

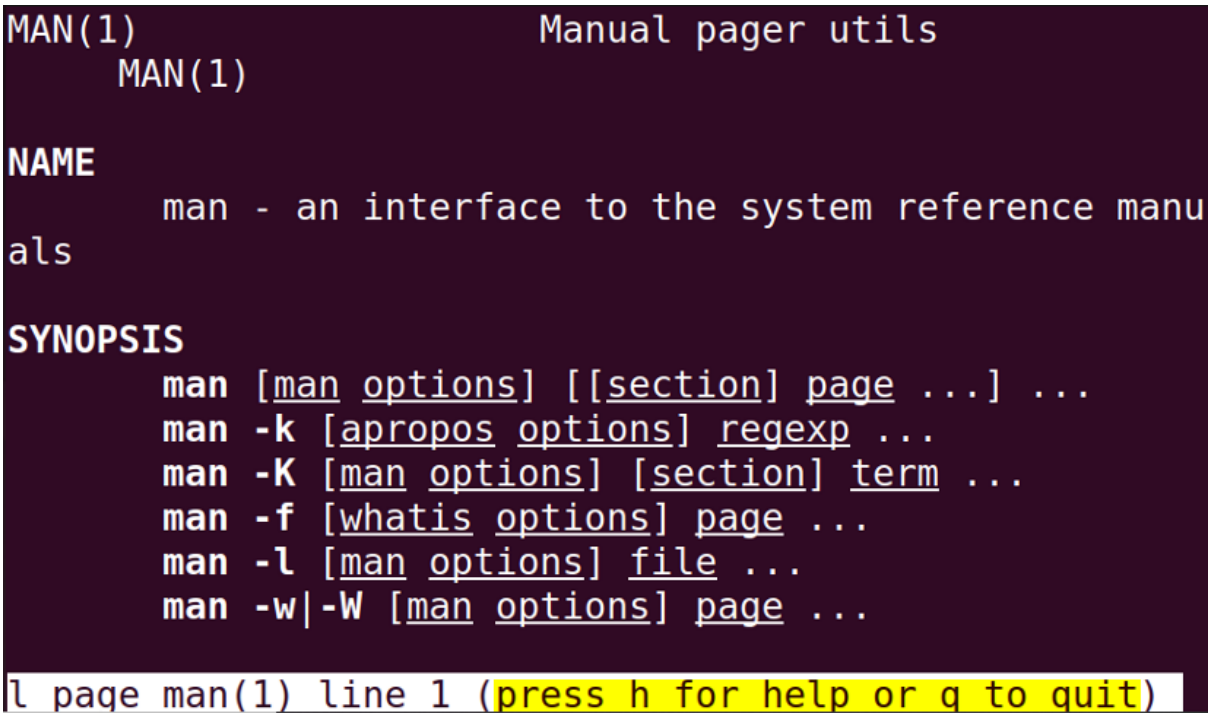
Useful Options & Shortcuts

-k, --apropos (Displays the short manual page on the terminal)
-I, --match-case (Performs case sensitive manual page search)
/pattern (It is a useful search shortcut built-in inside the man page.)

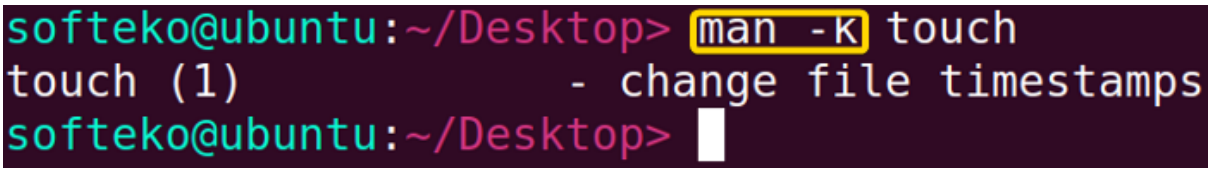
Example

You can **navigate** through the man page using arrow keys and other specific keys. To know more about the **man** page navigation try pressing **h** on the **man** page of any command.

$ softeko@ubuntu:~/Desktop> man man



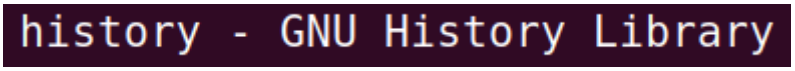
You can print short information from the **man** page using option **-k**.



17. history command

Executing this command one can view the previously used commands in the terminal.

Synopsis



Example

let's view some of our previously executed commands using the **history** command.

```
softeko@ubuntu:~> history
    1  nano .bashrc
    2  source bash
    3  bash
    4  nano .bashrc
    5  bash
    6  man grep
    7  mkdir Folder1 Folder2
    8  rm Folder1 Folder2
    9  rm Folder1 Folder2 -d
   10  ls
```

18. chmod command

chmod is the abbreviation for **ch**ange **mod**e. The **chmod** command can be used to alter the permission attributes of system contents.

Synopsis

```
chmod [OPTION]... MODE[,MODE]... FILE...
chmod [OPTION]... OCTAL-MODE FILE...
chmod [OPTION]... --reference=RFILE FILE...
```

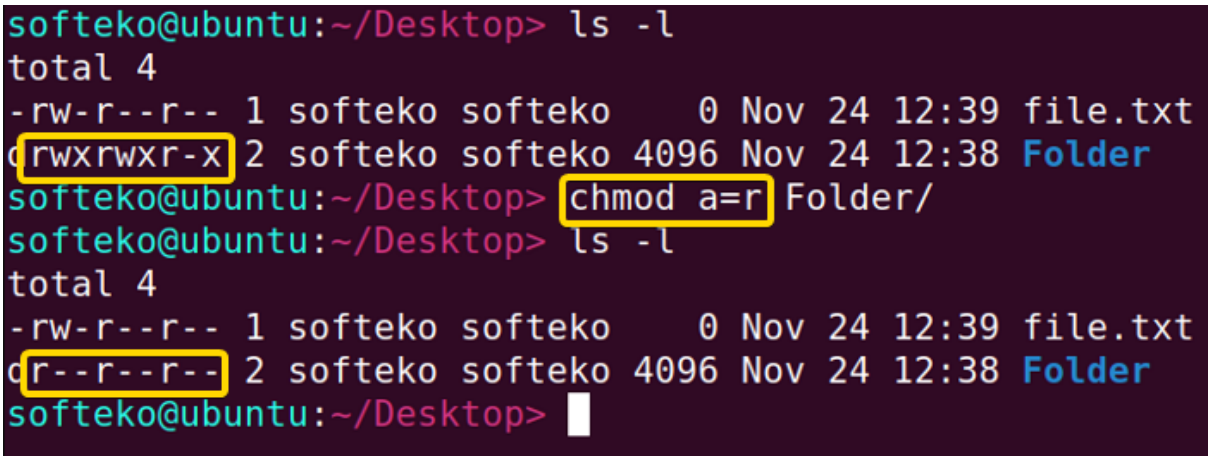
Useful Options

u represents **user** (**u+x** will empowers user with executable permission.**)**
g represents **groups** (**g-w** will revoked the modification power of members of the group.)
o represents **others** (**o+r** will empower others to read the contents.)
a represents **all** (**a=r** will grant everyone accessing power however it will revoke everyone from write and execution permissions.)
-c, --change (reports the occurrence of change.)

Example

The modification permission represented by character w of groups for filem named "file.txt" can be revoked using the command **chmod g-w**.

Now, for the directory named “Folder”, we want that everyone only be able to read the file. This canbe simply be done by using the command **chmod a=r**.



19. chown command

The **chown (ch**ange **own**er**)** command can be used to alter the owner of system files & directories.

Synopsis

```
chown [OPTION]... [OWNER][:[GROUP]] FILE...
chown [OPTION]... --reference=RFILE FILE...
```

Useful Options

-c, --change (reports the occurrence of the change.)
-R, --recursive (operates recursivelyon the desired files & directories)

Example

We want to change the ownership of a directory named “Folder”. It can be easily done using the **chown** command. However, we need to use **sudo** as a prefix of **chown** as only the **root** user has the permission to do so.

```
softeko@ubuntu:~/Desktop> ls -l
total 32
-rw-rw-r-- 1 softeko softeko     0 Nov 24 13:17 file.txt
drwxr-xr-x 2 softeko softeko  4096 Nov 24 12:38 Folder
-rw-rw-r-- 1 softeko softeko 25229 Nov 24 14:51 World-Cup
softeko@ubuntu:~/Desktop> sudo chown sowad Folder/
softeko@ubuntu:~/Desktop> ls -l
total 32
-rw-rw-r-- 1 softeko softeko     0 Nov 24 13:17 file.txt
drwxr-xr-x 2 sowad  softeko  4096 Nov 24 12:38 Folder
-rw-rw-r-- 1 softeko softeko 25229 Nov 24 14:51 World-Cup
softeko@ubuntu:~/Desktop>
```

20. echo command

The **echo** common is one of the simplest commands, it prints whatever we give to it.

Synopsis

```
echo [SHORT-OPTION]... [STRING]...
echo LONG-OPTION
```

Example

Let's print whatever we want!!!.

```
softeko@ubuntu:~/Desktop> echo Whatever we want!
Whatever we want!
softeko@ubuntu:~/Desktop>
```

Let's print today’s date. However, if we feed ‘**date**’ to **echo** it will just literally print ‘date’ so we have to use special syntax **$(command)**.

```
softeko@ubuntu:~/Desktop> echo Today is date
Today is date
softeko@ubuntu:~/Desktop> echo Today is $(date)
Today is Thu Nov 24 04:26:40 PM +06 2022
softeko@ubuntu:~/Desktop>
```

21. alias command

The **alias** command can be used to replace a command with user-modified instructions while running the command.

Synopsis

```
softeko@ubuntu:~> type alias
alias is a shell builtin
softeko@ubuntu:~> help alias
alias: alias [-p] [name[=value] ... ]
    Define or display aliases.

    Without arguments, `alias' prints the list of aliases in the reusable
    form `alias NAME=VALUE' on standard output.

    Otherwise, an alias is defined for each NAME whose VALUE is given.
    A trailing space in VALUE causes the next word to be checked for
    alias substitution when the alias is expanded.

    Options:
      -p        print all defined aliases in a reusable format

    Exit Status:
    alias returns true unless a NAME is supplied for which no alias has been
    defined.
softeko@ubuntu:~>
```

Note: Shell builtin commands don't have **man** pages. The **help** command can be used here.

Example

If we want to remove directories/files with the **rm** command while adding a confirmation prompt we need to run **rm -ri**. Now, we can make just **rm** sufficient enough to do the task of **rm -ri**, using the **alias** command.

```
softeko@ubuntu:~/Desktop> ls
file.txt  Folder
softeko@ubuntu:~/Desktop> rm -ri Folder/
rm: descend into directory 'Folder/'? n
softeko@ubuntu:~/Desktop> alias rm='rm -ri'
softeko@ubuntu:~/Desktop> rm Folder/
rm: descend into directory 'Folder/'? y
rm: remove regular empty file 'Folder/file'? n
rm: remove directory 'Folder/'? n
softeko@ubuntu:~/Desktop>
```

22. passwd command

The **passwd** command can be used to change the **password** of a specific user.

Synopsis

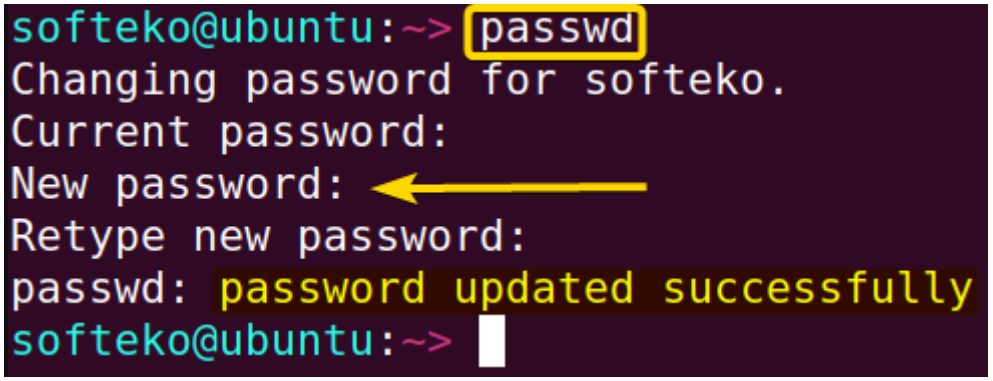
```
passwd - change user password
```

Useful Options

-d --delete User's password can be deleted.
-e --expire Immediately makes the password expired.
-i --inactive INACTIVE makes the password inactivate after a specific INACTIVATE period.

Example

We can simply change our password for the current user with **passwd** command.



23. less command

The **less** command is used to display the contents of a file on the terminal screen in page by page manner.

Synopsis

```
less -?
less --help
less -V
less --version
less [-[+]aABcCdeEfFgGiIJKLmMnNqQrRsSuUVwWX~]
     [-b space] [-h lines] [-j line] [-k keyfile]
     [-{oO} logfile] [-p pattern] [-P prompt] [-t tag]
     [-T tagsfile] [-x tab,...] [-y lines] [-[z] lines]
     [-# shift] [+[+]cmd] [--] [filename]...
(See the OPTIONS section for alternate option syntax with long option names.)
```

Useful Options

-n --line-numbers (when enabled it stops showing line numbers)
-N --LINE-NUMBERS (displays line numbers at starting points of each line.)

Example

We want to view the file named “World_Cup” with line numbers. This can be done by using the **less -N** command.

```
softeko@ubuntu:~/Desktop> ls
file.txt  Folder  World-Cup
softeko@ubuntu:~/Desktop> less -N World-Cup
```

Arrow keys can be used to navigate through the less view.
Press q to exit the less display.

24. whoami command

The **whoami** command simply displays the currently logged-in user.

Synopsis

```
whoami [OPTION]...
```

Example

The **whomai** command displays the current user in our case which is softeko.

```
softeko@ubuntu:~/Desktop> whoami
softeko
softeko@ubuntu:~/Desktop>
```

25. kill command

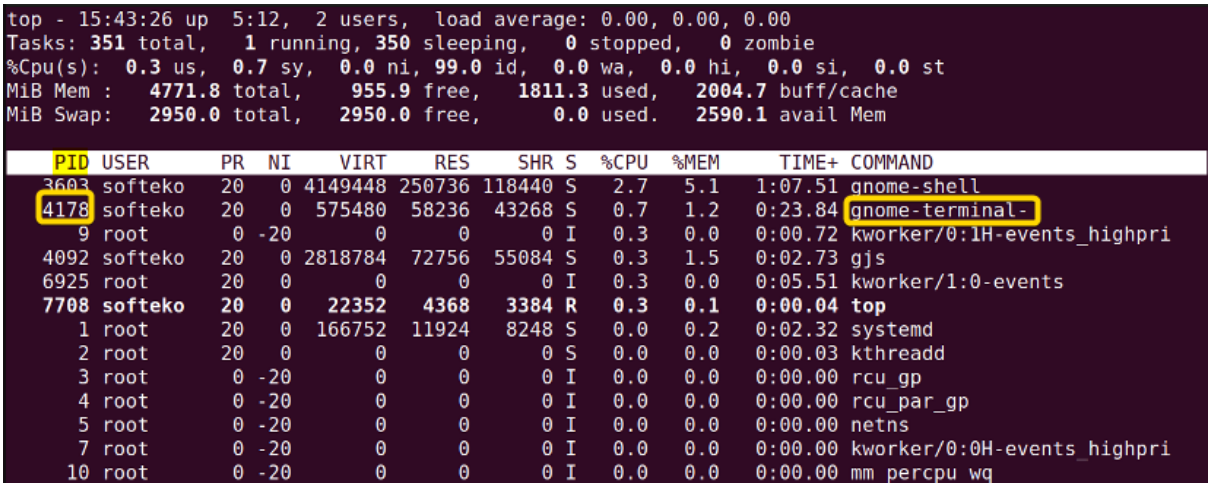
Synopsis

```
kill [options] <pid> [...]
```

Example

```
softeko@ubuntu:~/Desktop> top
```

The **top** command can be referred to as the task manager of linux. It shows information regarding CPU and memory optimization.



The process ID(**PID**) of the terminal is 4178 in this case. Now using the **kill** command with the PID of the terminal as its argument we can close the terminal.

```
softeko@ubuntu:~/Desktop> kill 4178
```

26. zip Command

zip command is used to compress files or folders into a .**zip** file in UNIX and UNIX-like operating systems. This allows for reducing the size of files and disk usage. Converting many files, and folders into a .zip file allows sharing and maintaining disk location by reducing file zie without any loss.

Synopsis

```
zip  [-aABcdDeEfFghjklLmoqrRSTuvVwXyz!@$] [--longoption ...]  [-b path]
[-n suffixes] [-t date] [-tt date] [zipfile [file ...]]  [-xi list]

zipcloak (see separate man page)

zipnote (see separate man page)

zipsplit (see separate man page)
```

Useful Options

-e, --encrypt (encrypt a file with a password which is entered through the prompt)
-i *.txt (include the only files with some conditions)
-m (moves the files into the zip files, deletes the original files after compression)
-r, --recurse-paths (compression throughout all the files and folders recursively)
-s, --split-size SIZE (the zip file is split into many different files into a specific size)

Example

We can use **zip** command to compress a single file.

```
softeko@UBUNTU:~/Desktop$ zip COMPRESSED.zip file_1
  adding: file_1 (deflated 62%)
```

We can also compress multiple files.

```
softeko@UBUNTU:~/Desktop$ zip COMPRESSED.zip file_1 file_2 file_3
updating: file_1 (deflated 62%)
  adding: file_2 (deflated 59%)
  adding: file_3 (deflated 62%)
```

We can encrypt a file with password with **-e** option.

```
softeko@UBUNTU:~/Desktop$ zip -e COMPRESSED.zip file_1
Enter password:
Verify password:
  adding: file_1 (deflated 62%)
```

We can move the file into a .zip file and delete the original with **-m** option.

```
softeko@UBUNTU:~/Desktop$ zip -m COMPRESSED.zip file_1
updating: file_1 (deflated 62%)
```

27. unzip Command

unzip command is used to extract files from a .zip file. This command creates a new folder in the current directory where the field inside the **zip** files is extracted.

Synopsis

```
unzip  [-Z] [-cflptTuvz[abjnoqsCDKLMUVWX$/:^]] file[.zip] [file(s) ...]
[-x xfile(s) ...] [-d exdir]
```

Useful Options

-u (update existing files inside the folder and create new ones if required)

-i (shows the files or folders inside the compressed document)

Example

To simply extract files from a compressed folder, we use **unzip** command.

```
softeko@UBUNTU:~/Desktop$ unzip COMPRESSED.zip
Archive:  COMPRESSED.zip
  inflating: file_1
  inflating: file_2
  inflating: file_3
```

We can use **-l** option too view the files inside the compressed files.

```
softeko@UBUNTU:~/Desktop$ unzip -l COMPRESSED.zip
Archive:  COMPRESSED.zip
  Length      Date    Time    Name
---------  ---------- -----   ----
   595123  2022-11-23 11:03   file_1
   418307  2022-11-23 11:04   file_2
   389179  2022-11-23 11:04   file_3
---------                     -------
  1402609                     3 files
```

28. wget Command

wget is a download command which downloads files or webpages non-interactively from the network.

Synopsis

```
wget [option]... [URL]...
```

Useful Options

-b (downloads a file in the background)
-c (continues a partially downloaded file)

Example

To download a webpage, we can use **wget** command.

```
softeko@UBUNTU:~/Desktop$ wget www.linuxsimply.com
URL transformed to HTTPS due to an HSTS policy
--2022-11-23 12:17:49--  https://www.linuxsimply.com/
Resolving www.linuxsimply.com (www.linuxsimply.com)... 172.67.219.37, 104.21.67.
91, 2606:4700:3031::ac43:db25, ...
Connecting to www.linuxsimply.com (www.linuxsimply.com)|172.67.219.37|:443... co
nnected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://linuxsimply.com/ [following]
--2022-11-23 12:17:51--  https://linuxsimply.com/
Resolving linuxsimply.com (linuxsimply.com)... 104.21.67.91, 172.67.219.37, 2606
:4700:3035::6815:435b, ...
Connecting to linuxsimply.com (linuxsimply.com)|104.21.67.91|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: 'index.html'

index.html              [ <=>                ]  56.96K  --.-KB/s    in 0.1s

2022-11-23 12:17:52 (486 KB/s) - 'index.html' saved [58324]
```

To download a webpage in the background requires **-b** option.

```
softeko@UBUNTU:~/Desktop$ wget -b www.linuxsimply.com
Continuing in background, pid 8861.
Output will be written to 'wget-log'.
```

29. df command

df (**d**isk **f**ree) command shows the size, used, available space and mounted on the information of the filesystem.

Synopsis

```
df [OPTION]... [FILE]...
```

Useful Options

-a, --all (displays all file systems including inaccessible or hidden files)
-h, --human-readable (displays information in human-readable format)
-t, --type=file_type (displays files of a certain file type)
-l, --local (displays local file system)

Example

Using only **df** command shows default information.

```
softeko@UBUNTU:~/Desktop$ df
Filesystem     1K-blocks     Used Available Use% Mounted on
tmpfs             810584     2136    808448   1% /run
/dev/sda3       50770432 23813584  24345444  50% /
tmpfs            4052900        0   4052900   0% /dev/shm
tmpfs               5120        4      5116   1% /run/lock
/dev/sda2         524252     5364    518888   2% /boot/efi
tmpfs             810580     4732    805848   1% /run/user/1002
```

Using **df -h** shows us the file size in a human-readable format.

```
softeko@UBUNTU:~/Desktop$ df -h
Filesystem      Size  Used Avail Use% Mounted on
tmpfs           792M  2.1M  790M   1% /run
/dev/sda3        49G   23G   24G  50% /
tmpfs           3.9G     0  3.9G   0% /dev/shm
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
/dev/sda2       512M  5.3M  507M   2% /boot/efi
tmpfs           792M  4.7M  787M   1% /run/user/1002
```

Using **df file_name** shows information about only the **file_name.**

```
softeko@UBUNTU:~/Desktop$ df COMPRESSED.zip
Filesystem     1K-blocks     Used Available Use% Mounted on
/dev/sda3       50770432 23813584  24345444  50% /
```

30. ping command

ping (**P**acket **In**ternet **G**roper) command shows the information about the network information about the host and the server. It can check the internet connection and show the latency between the host and server.

Synopsis

```
ping [-aAbBdDfhLnOqrRUvV46] [-c count] [-F flowlabel] [-i interval]
     [-I interface] [-l preload] [-m mark] [-M pmtudisc_option]
     [-N nodeinfo_option] [-w deadline] [-W timeout] [-p pattern]
     [-Q tos] [-s packetsize] [-S sndbuf] [-t ttl]
     [-T timestamp option] [hop...] {destination}
```

Useful Options

-c COUNT (sends COUNT number of ECHO_REQUEST)
-i COUNT (sends a packet in every COUNT second interval)
-f (flood ping with rapid display, **sudo** permission is needed)

Example

Finds information about the **localhost**, exiting the command needed, using ctrl+c.

```
softeko@UBUNTU:~/Desktop$ ping localhost
PING localhost (127.0.0.1) 56(84) bytes of data.
64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0.018 ms
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.029 ms
64 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64 time=0.036 ms
64 bytes from localhost (127.0.0.1): icmp_seq=4 ttl=64 time=0.029 ms
64 bytes from localhost (127.0.0.1): icmp_seq=5 ttl=64 time=0.026 ms
64 bytes from localhost (127.0.0.1): icmp_seq=6 ttl=64 time=0.027 ms
64 bytes from localhost (127.0.0.1): icmp_seq=7 ttl=64 time=0.026 ms
64 bytes from localhost (127.0.0.1): icmp_seq=8 ttl=64 time=0.041 ms
64 bytes from localhost (127.0.0.1): icmp_seq=9 ttl=64 time=0.026 ms
^C
--- localhost ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8171ms
rtt min/avg/max/mdev = 0.018/0.028/0.041/0.006 ms
```

Finds **ping** information about a website exiting the command needed, using **ctrl+c**.

```
softeko@UBUNTU:~$ ping www.linuxsimply.com
PING www.linuxsimply.com (172.67.219.37) 56(84) bytes of data.
64 bytes from 172.67.219.37 (172.67.219.37): icmp_seq=1 ttl=128 time=48.2 ms
64 bytes from 172.67.219.37 (172.67.219.37): icmp_seq=2 ttl=128 time=48.9 ms
64 bytes from 172.67.219.37 (172.67.219.37): icmp_seq=3 ttl=128 time=47.4 ms
64 bytes from 172.67.219.37 (172.67.219.37): icmp_seq=4 ttl=128 time=47.3 ms
64 bytes from 172.67.219.37 (172.67.219.37): icmp_seq=5 ttl=128 time=47.7 ms
64 bytes from 172.67.219.37 (172.67.219.37): icmp_seq=6 ttl=128 time=50.7 ms
64 bytes from 172.67.219.37 (172.67.219.37): icmp_seq=7 ttl=128 time=47.4 ms
64 bytes from 172.67.219.37 (172.67.219.37): icmp_seq=8 ttl=128 time=47.8 ms
64 bytes from 172.67.219.37 (172.67.219.37): icmp_seq=9 ttl=128 time=47.1 ms
^C
--- www.linuxsimply.com ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8015ms
rtt min/avg/max/mdev = 47.096/48.052/50.727/1.077 ms
```

We can add **-c NUM** option to limit the number of packets using ping.

```
softeko@UBUNTU:~/Desktop$ ping -c 3 localhost
PING localhost (127.0.0.1) 56(84) bytes of data.
64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0.025 ms
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.031 ms
64 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64 time=0.027 ms

--- localhost ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2025ms
rtt min/avg/max/mdev = 0.025/0.027/0.031/0.002 ms
```

31. diff command

diff command finds checks 2 files and shows the difference between 2 files. Normally, it does not change the content of files, but it can generate a script.

Synopsis

```
diff [OPTION]... FILES
```

Useful Options

-c (shows the differences between the 2 files in context mode)
-u, -U, --unified (shows the unified context of the 2 files)
-r, --recursive (compares files inside any subdirectories)
-i, --ignore-case (ignore cases)

Example

Using the **diff** command with 2 files shows the difference between those 2 files.

```
softeko@UBUNTU:~/Desktop$ diff file_1.txt file_2.txt
2,5c2,4
< Linux
< Ubuntu
< 22.04
< Jammy Jellyfish
---
> MacOS
> 13.0
> Ventura
```

We can use **-u** option to see the difference in the unified context of 2 files.

```
softeko@UBUNTU:~/Desktop$ diff -u file_1.txt file_2.txt
--- file_1.txt  2022-11-23 12:42:27.909161219 +0600
+++ file_2.txt  2022-11-23 12:44:17.260027429 +0600
@@ -1,6 +1,5 @@
 OS
-Linux
-Ubuntu
-22.04
-Jammy Jellyfish
+MacOS
+13.0
+Ventura
 Bye
```

Option **-c** shows both files in context mode.

```
softeko@UBUNTU:~/Desktop$ diff -c file_1.txt file_2.txt
*** file_1.txt  2022-11-23 12:42:27.909161219 +0600
--- file_2.txt  2022-11-23 12:44:17.260027429 +0600
***************
*** 1,6 ****
  OS
! Linux
! Ubuntu
! 22.04
! Jammy Jellyfish
  Bye
--- 1,5 ----
  OS
! MacOS
! 13.0
! Ventura
  Bye
```

32. ps command

ps (**P**rocess **S**tatus) command shows the process status and information about that process.

Synopsis

```
ps [options]
```

Useful Options

-a, --all (shows all the processes including hidden processes)
-r (shows all the running processes)
--pid PID (shows the specific process according to PID)

Example

Simply typing **ps** shows the process status of the Linux system.

```
softeko@UBUNTU:~/Desktop$ ps -r
    PID TTY      STAT   TIME COMMAND
   9147 pts/0    R+     0:00 ps -r
```

Option **-a** shows all the processes including hidden processes.

```
softeko@UBUNTU:~/Desktop$ ps -a
    PID TTY          TIME CMD
   5660 tty2     00:00:00 gnome-session-b
   9148 pts/0    00:00:00 ps
```

To look into a specific process, **--pid** command is used.

```
softeko@UBUNTU:~/Desktop$ ps
    PID TTY          TIME CMD
   8794 pts/0    00:00:00 bash
   9149 pts/0    00:00:00 ps
softeko@UBUNTU:~/Desktop$ ps --pid 8794
    PID TTY          TIME CMD
   8794 pts/0    00:00:00 bash
```

33. apt command

apt (**A**dvanced **P**ackage **T**ool) command manages different packages including install, remove, update, etc.

Synopsis

```
apt [-h] [-o=config_string] [-c=config_file] [-t=target_release]
    [-a=architecture] {list | search | show | update |
    install pkg [{=pkg_version_number | /target_release}]...  |
    remove pkg...  | upgrade | full-upgrade | edit-sources |
    {-v | --version} | {-h | --help}}
```

Useful Options

apt [install, update, upgrade] (downloads package information and install, update or upgrade the package, **sudo** permission is required)
apt full-upgrade (upgrade all installed packages, even remove some if required)
apt remove Package_Name (removes a specific package)
apt autoremove (removes unneeded dependencies)
apt --only-upgrade install Package_Name (installs or upgrades a specific package)

Example

sudo apt update command updates all packages.

```
softeko@UBUNTU:~/Desktop$ sudo apt update
[sudo] password for softeko:
Hit:1 https://dl.google.com/linux/chrome/deb stable InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu jammy InRelease
Get:3 http://us.archive.ubuntu.com/ubuntu jammy-updates InRelease [114 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu jammy-backports InRelease [99.8 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:6 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [726
 kB]
Get:7 http://us.archive.ubuntu.com/ubuntu jammy-updates/main i386 Packages [380
kB]
Get:8 http://us.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [164
 kB]
Get:9 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 DEP-11 Metada
ta [95.0 kB]
Get:10 http://us.archive.ubuntu.com/ubuntu jammy-updates/restricted i386 Package
s [23.4 kB]
Get:11 http://us.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packag
es [445 kB]
```

We can use **sudo apt upgrade** command to upgrade all available packages.

```
softeko@UBUNTU:~/Desktop$ sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
#
# News about significant security updates, features and services will
# appear here to raise awareness and perhaps tease /r/Linux ;)
# Use 'pro config set apt_news=false' to hide this and future APT news.
#
The following packages have been kept back:
  apport apport-gtk python3-apport python3-problem-report
The following packages will be upgraded:
  firmware-sof-signed libflac8 libfreerdp-client2-2 libfreerdp-server2-2
  libfreerdp2-2 libwinpr2-2 rsync
7 upgraded, 0 newly installed, 0 to remove and 4 not upgraded.
5 standard LTS security updates
Need to get 1,223 kB/2,624 kB of archives.
After this operation, 1,172 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

The **autoremove** option removes all unneeded dependencies.

```
softeko@UBUNTU:~/Desktop$ sudo apt autoremove
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
0 upgraded, 0 newly installed, 0 to remove and 4 not upgraded.
```

The **--only-upgrade Packae_name** option updates a specific package.

```
softeko@UBUNTU:~/Desktop$ sudo apt --only-upgrade install google-chrome-stable
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
google-chrome-stable is already the newest version (107.0.5304.110-1).
0 upgraded, 0 newly installed, 0 to remove and 4 not upgraded.
```

34. dd command

dd command converts and copies a file to another directory. This command can be used to create a backup inside the hard drive or an external hard drive.

Synopsis

```
dd [OPERAND]...
dd OPTION
```

Useful Options

if (read the file instead of standard input)
of (write the file instead of standard output)

Example

dd command is used to back up a folder into another directory as **backup.img** file.

```
softeko@Ubuntu:~$ sudo dd if=/dev/sda of=~/backup.img
[sudo] password for softeko:
dd: writing to '/home/softeko/backup.img': No space left on device
46645353+0 records in
46645352+0 records out
23882420224 bytes (24 GB, 22 GiB) copied, 173.489 s, 138 MB/s
```

35. top command

top (**T**able **O**f **P**rocesses) command shows the currently running process inside linux. It gives a dynamic but not interactive view of the process.

Synopsis

```
top -hv|-bcEeHiOSs1 -d secs -n max -u|U user -p pids -o field -w [cols]

The traditional switches `-' and whitespace are optional.
```

Useful Options

-n NUM (shows the top NUM number of processes)

-u PARAS (shows the processes according to PID or User as given as paras)
-d TIME (shows a dynamic view which updates in TIME tenth of seconds)

Example

Simply typing **top** shows the processes dynamically.

```
top - 15:05:50 up  4:24,  1 user,  load average: 0.00, 0.00, 0.00
Tasks: 289 total,   1 running, 288 sleeping,   0 stopped,   0 zombie
%Cpu(s):  1.0 us,  1.0 sy,  0.0 ni, 98.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :   7915.8 total,   4834.7 free,   1217.7 used,   1863.4 buff/cache
MiB Swap:   2048.0 total,   2048.0 free,      0.0 used.   6370.4 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   5820 softeko   20   0 4436744 267592 121020 S   4.7   3.3   1:43.39 gnome-s+
   8764 softeko   20   0  639328  56072  42456 S   1.0   0.7   0:08.06 gnome-t+
   6030 softeko   20   0  320932   7604   6820 S   0.3   0.1   0:00.53 gsd-hou+
   9225 root      20   0       0      0      0 I   0.3   0.0   0:00.97 kworker+
      1 root      20   0  168240  13504   8168 S   0.0   0.2   0:04.76 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.05 kthreadd
      3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par+
      5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 netns
      7 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker+
      9 root       0 -20       0      0      0 I   0.0   0.0   0:02.17 kworker+
     10 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_perc+
     11 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tas+
     12 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tas+
     13 root      20   0       0      0      0 S   0.0   0.0   0:00.37 ksoftir+
```

We can use option **-n 10** to find the top 10 running processes according to cpu usage.

```
top - 15:07:10 up  4:25,  1 user,  load average: 0.00, 0.00, 0.00
Tasks: 289 total,   1 running, 288 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.2 us,  0.2 sy,  0.0 ni, 99.7 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :   7915.8 total,   4834.8 free,   1217.6 used,   1863.4 buff/cache
MiB Swap:   2048.0 total,   2048.0 free,      0.0 used.   6370.5 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
    681 systemd+  20   0   14824   6068   5272 S   0.3   0.1   0:16.04 systemd+
   5942 softeko   20   0  632848  14344  11804 S   0.3   0.2   0:00.52 xdg-des+
  10084 softeko   20   0   22224   4184   3316 R   0.3   0.1   0:00.04 top
      1 root      20   0  168240  13504   8168 S   0.0   0.2   0:04.76 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.05 kthreadd
      3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par+
      5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 netns
      7 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker+
      9 root       0 -20       0      0      0 I   0.0   0.0   0:02.18 kworker+
     10 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_perc+
     11 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tas+
     12 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tas+
     13 root      20   0       0      0      0 S   0.0   0.0   0:00.37 ksoftir+
     14 root      20   0       0      0      0 I   0.0   0.0   0:04.44 rcu_sch+
```

36. htop command

htop process is quite similar to the top process but in **htop**, the user can have interactive control over the top running processes in Linux. **htop** may need to be installed in the terminal first for usage.

```
softeko@UBUNTU:~/Desktop$ sudo snap install htop
htop 3.2.1 from Maximiliano Bertacchini (maxiberta) installed
```

Synopsis

There is no manual page for **htop**, but help, there are different commands and options shown.

```
softeko@UBUNTU:~/Desktop$ htop -help
locales-launch: Data of en_US locale not found, generating, please wait...
htop 3.2.1
(C) 2004-2019 Hisham Muhammad. (C) 2020-2022 htop dev team.
Released under the GNU GPLv2+.

-C --no-color                   Use a monochrome color scheme
-d --delay=DELAY                Set the delay between updates, in tenths of seco
nds
-F --filter=FILTER              Show only the commands matching the given filter
-h --help                       Print this help screen
-H --highlight-changes[=DELAY]  Highlight new and old processes
-M --no-mouse                   Disable the mouse
-p --pid=PID[,PID,PID...]       Show only the given PIDs
   --readonly                   Disable all system and process changing features
-s --sort-key=COLUMN            Sort by COLUMN in list view (try --sort-key=help
 for a list)
-t --tree                       Show the tree view (can be combined with -s)
-u --user[=USERNAME]            Show only processes for a given user (or $USER)
-U --no-unicode                 Do not use unicode but plain ASCII
-V --version                    Print version info
```

Example

Simply typing **htop** shows the prompt where each process can be analyzed interactively.

```
  0[||                                1.3%] Tasks: 117, 256 thr, 174 kthr; 1 runni
  1[||                                3.3%] Load average: 0.06 0.06 0.02
Mem[||||||||||||||          1.17G/7.73G] Uptime: 04:35:27
Swp[                           0K/2.00G]

  Main   I/O
    PID USER       PRI  NI  VIRT   RES   SHR S  CPU%▽MEM%   TIME+  Command
   5561 root        20   0  391M 11836  9936 S   0.0  0.1  0:00.00 gdm-session-w
   5563 root        20   0  391M 11836  9936 S   0.0  0.1  0:00.01 gdm-session-w
   5570 softeko     20   0 18108 10856  8072 S   0.0  0.1  0:01.07 /lib/systemd/
   5571 softeko     20   0  166M  4364    12 S   0.0  0.1  0:00.00 (sd-pam)
   5578 softeko      9 -11 48580  6728  5596 S   0.0  0.1  0:00.03 /usr/bin/pipe
   5579 softeko     20   0 32468  6668  5568 S   0.0  0.1  0:00.03 /usr/bin/pipe
   5580 softeko      9 -11 1471M 28312 21212 S   0.0  0.3  0:12.13 /usr/bin/puls
   5583 softeko     20   0  244M  7536  6552 S   0.0  0.1  0:00.13 /usr/bin/gnom
   5584 softeko     20   0  244M  7536  6552 S   0.0  0.1  0:00.00 /usr/bin/gnom
   5590 softeko     20   0  244M  7536  6552 S   0.0  0.1  0:00.07 /usr/bin/gnom
   5597 softeko     20   0 10460  6660  4172 S   0.0  0.1  0:01.59 /usr/bin/dbus
   5599 softeko     20   0  533M  7616  6916 S   0.0  0.1  0:00.09 /usr/libexec/
   5600 softeko     20   0  315M  8452  7428 S   0.0  0.1  0:00.10 /usr/libexec/
   5601 softeko     20   0  533M  7616  6916 S   0.0  0.1  0:00.00 /usr/libexec/
   5602 softeko     20   0  533M  7616  6916 S   0.0  0.1  0:00.06 /usr/libexec/
F1Help  F2Setup F3SearchF4FilterF5Tree  F6SortByF7Nice -F8Nice +F9Kill  F10Quit
```

37. useradd command

useradd command is used to add a user to the Linux system. sudo command is needed to create a new user. The superuser **sudo** command is required to use this command.

Synopsis

```
useradd [options] LOGIN

useradd -D

useradd -D [options]
```

Useful Options

-d DIRECTORY USER_NAME (creates a user in DIRECTORY)
-u, --uid ID (creates a user with the user id of ID)
-g --gid ID (creates a user with a group id of ID)
-M USER (creates a user without any home directory)
-e DATE USER (creates a user with an expiry date)

Example

Creating a simple user is possible with **useradd** command.

```
softeko@UBUNTU:~/Desktop$ sudo useradd linuxsimply
softeko@UBUNTU:~/Desktop$
```

Option **-d** creates a user inside a specific directory.

```
softeko@UBUNTU:~$ sudo useradd -d ~/home/softeko LS
[sudo] password for softeko:
softeko@UBUNTU:~$
```

We can also use **-u NUM** to create a user with a specific user id.

```
softeko@UBUNTU:~$ sudo useradd -u 999 LinuxSimply
softeko@UBUNTU:~$
```

38. unalias command

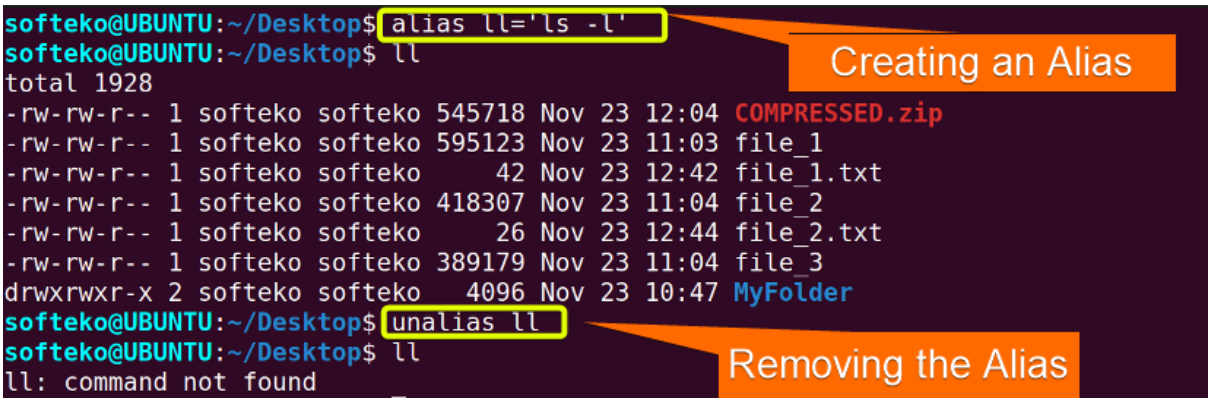
unalias command removes **alias** created before.

Useful Options

-a (removes all alias)

Example

Removes an alias can be done using **unalias ALS** command.



Removing all alias requires using **-a** option.

39. uname command

uname command shows information about the system. Without any option it prints Linux.

Synopsis

```
ssh [-46AaCfGgKkMNnqsTtVvXxYy] [-B bind_interface] [-b bind_address]
    [-c cipher_spec] [-D [bind_address:]port] [-E log_file]
    [-e escape_char] [-F configfile] [-I pkcs11] [-i identity_file]
    [-J destination] [-L address] [-l login_name] [-m mac_spec]
    [-O ctl_cmd] [-o option] [-p port] [-Q query_option] [-R address]
    [-S ctl_path] [-W host:port] [-w local_tun[:remote_tun]] destination
    [command [argument ...]]
```

Useful Options

-a (displays all information about the system)
-o (displays the operating system name)
-s (displays the kernel name)
-p (displays the processor type)

Example

Option **-s** displays kernel name.

Option **-o** displays operating system name.

We can use **-a** to display all information about the system.

```
softeko@UBUNTU:~/Desktop$ uname -a
Linux UBUNTU 5.15.0-53-generic #59-Ubuntu SMP Mon Oct 17 18:53:30 UTC 2022 x86_6
4 x86_64 x86_64 GNU/Linux
softeko@UBUNTU:~/Desktop$
```

40. ssh command

ssh (**S**ecured **Sh**ell) command is used for securely connecting to a remote server.

Synopsis

```
ssh [-46AaCfGgKkMNnqsTtVvXxYy] [-B bind_interface] [-b bind_address]
    [-c cipher_spec] [-D [bind_address:]port] [-E log_file]
    [-e escape_char] [-F configfile] [-I pkcs11] [-i identity_file]
    [-J destination] [-L address] [-l login_name] [-m mac_spec]
    [-O ctl_cmd] [-o option] [-p port] [-Q query_option] [-R address]
    [-S ctl_path] [-W host:port] [-w local_tun[:remote_tun]] destination
    [command [argument ...]]
```

41. traceroute command

traceroute command displays the packet route to reach the host. This command can be used to see the different hops or routes it takes to connect with a particular hostname.

Synopsis

```
traceroute [-46dFITUnreAV] [-f first_ttl] [-g gate,...]
        [-i device] [-m max_ttl] [-p port] [-s src_addr]
        [-q nqueries] [-N squeries] [-t tos]
        [-l flow_label] [-w waittimes] [-z sendwait] [-UL] [-D]
        [-P proto] [--sport=port] [-M method] [-O mod_options]
        [--mtu] [--back]
        host [packet_len]
traceroute6  [options]
tcptraceroute  [options]
lft  [options]
```

Useful Options

-g Host_name (Routing the packet through a gate or IP address)
-m Host_name (Fixing the maximum number of packets)

Example

Simply typing traceroute shows the maximum number of hops and the routes.

```
softeko@Ubuntu:~$ traceroute www.linuxsimply.com
traceroute to www.linuxsimply.com (104.21.67.91), 30 hops max, 60 byte packets
 1  _gateway (192.168.0.1)  4.026 ms  3.799 ms  3.593 ms
 2  10.100.101.161 (10.100.101.161)  3.270 ms  3.025 ms  2.828 ms
 3  10.100.105.13 (10.100.105.13)  2.623 ms  2.444 ms  2.222 ms
 4  10.100.105.1 (10.100.105.1)  1.934 ms  1.729 ms  1.537 ms
 5  10.10.102.37 (10.10.102.37)  15.671 ms  15.431 ms  15.104 ms
 6  43.245.194.98 (43.245.194.98)  2.403 ms  1.756 ms  2.175 ms
 7  104.21.67.91 (104.21.67.91)  1.625 ms  1.584 ms  1.540 ms
```

To fix a gate for routing **-g** option is used.

```
softeko@Ubuntu:~$ traceroute -g 192.168.43.45 google.com
traceroute to google.com (142.250.196.46), 30 hops max, 72 byte packets
 1  * * *
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  * * *
 9  * * *
10  * * *
11  * * *
12  * * *
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
```

To fix the maximum number of hops, **-m** option is used.

```
softeko@Ubuntu:~$ traceroute  -m 5 google.com
traceroute to google.com (142.250.196.46), 5 hops max, 60 byte packets
 1  _gateway (192.168.0.1)  0.763 ms  0.673 ms  0.628 ms
 2  10.100.101.161 (10.100.101.161)  1.978 ms  1.935 ms  2.178 ms
 3  10.100.105.13 (10.100.105.13)  1.855 ms  1.818 ms  1.776 ms
 4  10.100.105.1 (10.100.105.1)  1.706 ms  1.602 ms  1.568 ms
 5  103.12.172.221 (103.12.172.221)  1.302 ms  1.261 ms  1.436 ms
```

42. groups command

Groups are used to manage multiple users and permissions. Several users can be members of a group and then the group can have permission for some files or folders to manage permissions.

Synopsis

```
groups [OPTION]... [USERNAME]...
```

Example

Simply typing **groups** will display all the available groups in the machine.

```
softeko@Ubuntu:~$ groups
softeko adm cdrom sudo dip plugdev lpadmin lxd sambashare
softeko@Ubuntu:~$
```

We can use **id** command to find the group's id.

```
softeko@Ubuntu:~$ id softeko
uid=1001(softeko) gid=1001(softeko) groups=1001(softeko),4(adm),24(cdrom),27(sud
o),30(dip),46(plugdev),122(lpadmin),134(lxd),135(sambashare)
softeko@Ubuntu:~$
```

43. shutdown command

shutdown command is used to shut down the system with some conditions. This command can schedule a shutdown at a specific time or cancel an already scheduled shutdown. The superuser **sudo** permission is required for this command.

Synopsis

```
shutdown [OPTIONS...] [TIME] [WALL...]
```

Useful Options

--halt Time/Delay (Schedule a shutdown at a specific time or after a certain delay)
-c (Cancel any shutdown command scheduled beforehand)
-r Time/Delay (Schedule a restart at a specific time or after a certain delay)

Example

Option **--halt** can schedule a shutdown at 2:00 PM. The display will also show the date and the region.

```
softeko@Ubuntu:~$ sudo shutdown --halt 14:00
[sudo] password for softeko:
Shutdown scheduled for Thu 2022-11-24 14:00:00 +06, use 'shutdown -c' to cancel.
```

We can also schedule a shutdown after a 5 minute delay **--halt** option.

```
softeko@Ubuntu:~$ sudo shutdown --halt +5
Shutdown scheduled for Thu 2022-11-24 12:43:34 +06, use 'shutdown -c' to cancel.
softeko@Ubuntu:~$
```

Option **-c** cancels a shutdown.

```
softeko@Ubuntu:~$ sudo shutdown -c
softeko@Ubuntu:~$
```

44. exit command

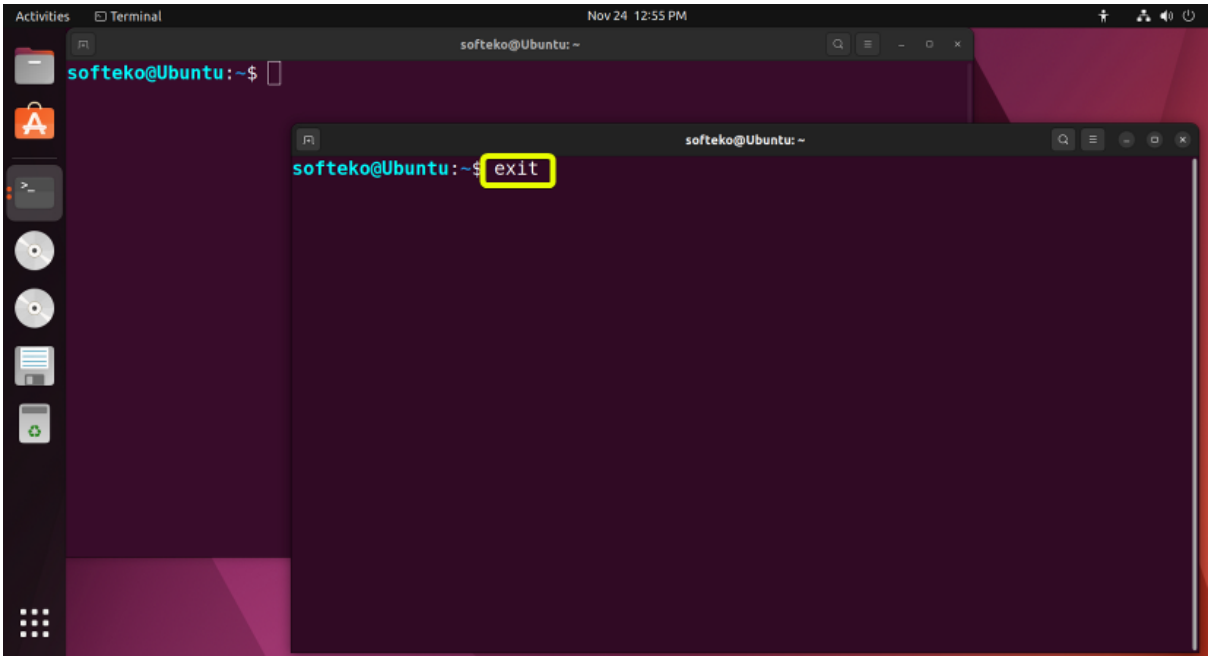
exit command exits the terminal. It is a shell command, which means it doesn't have any man page.

Synopsis

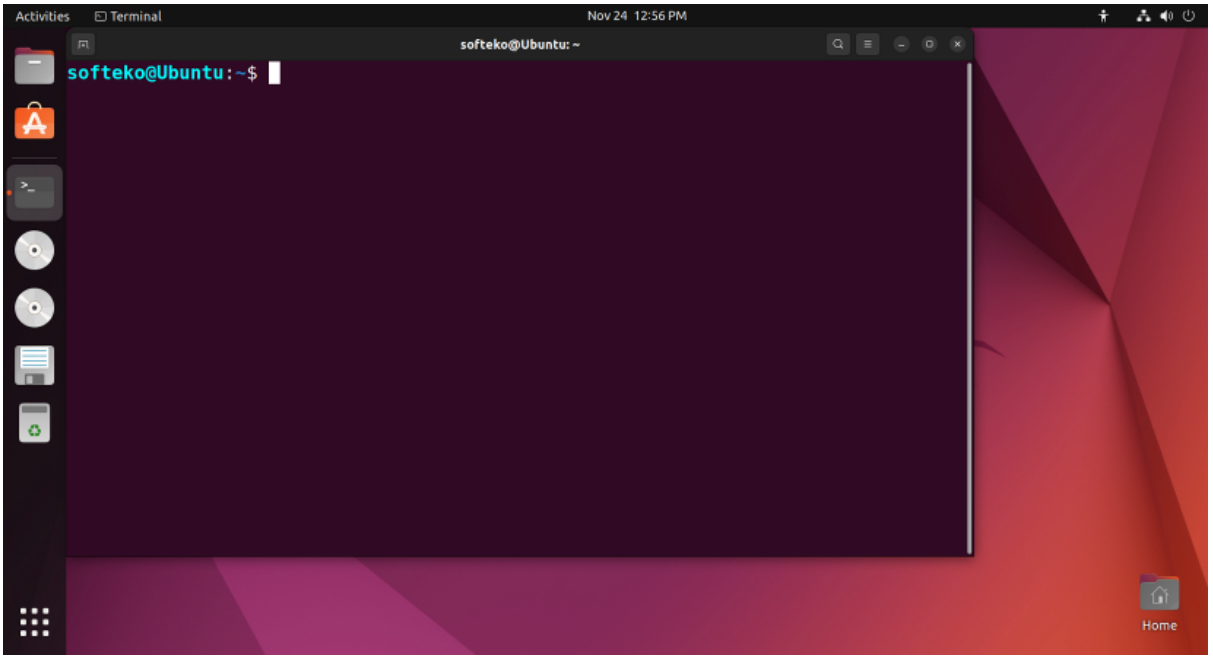
```
softeko@Ubuntu:~$ exit --help
exit: exit [n]
    Exit the shell.

    Exits the shell with a status of N.  If N is omitted, the exit status
    is that of the last command executed.
```

Example

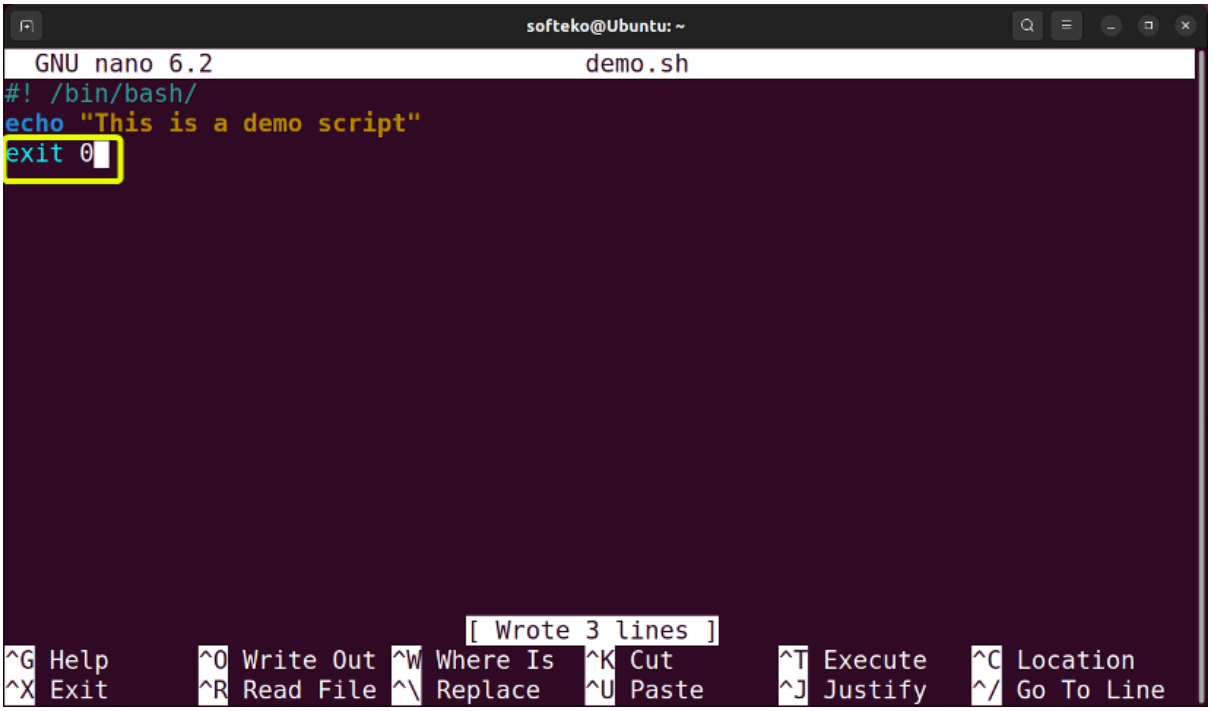
exit command terminates or exits a terminal.



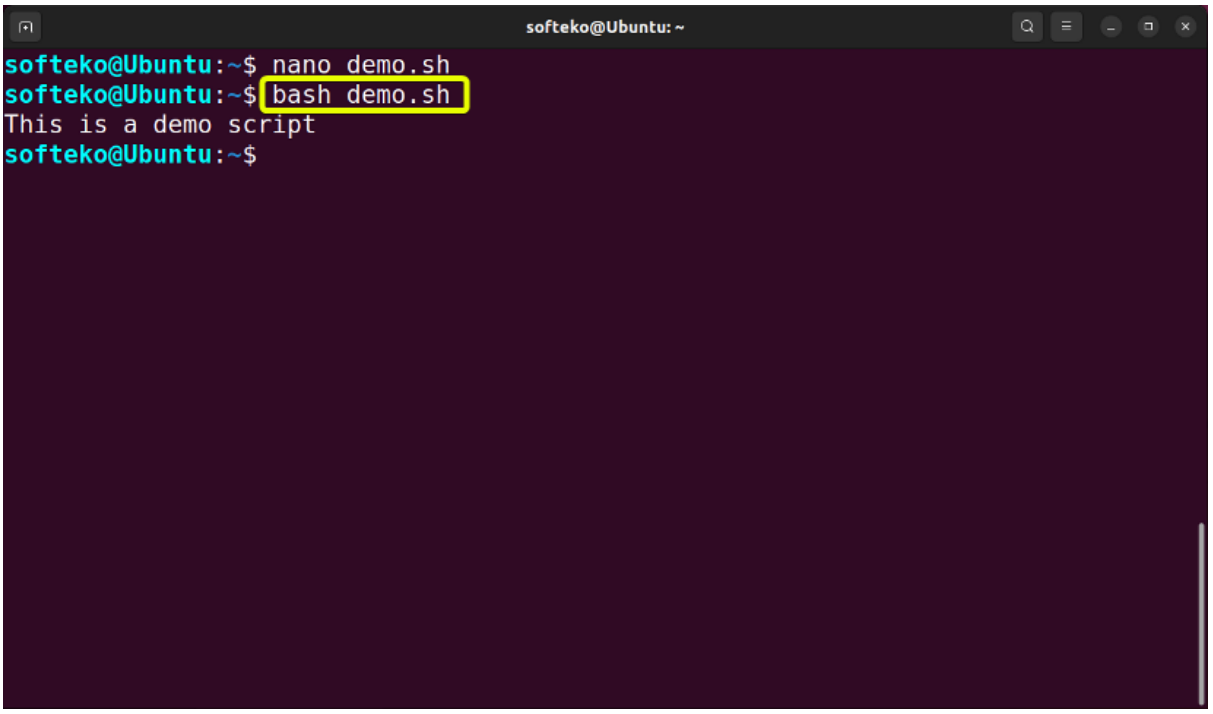
Now that terminal is not available anymore.



The command **exit** is also used to exit a script. First, a demo script is created and **exit 0** is present at the end.



Now running the script using **bash** command, the script will exit at the end.



45. wc command

wc (**W**ord **c**ount) command is used to count the number of characters or the number of lines in a file.

Synopsis

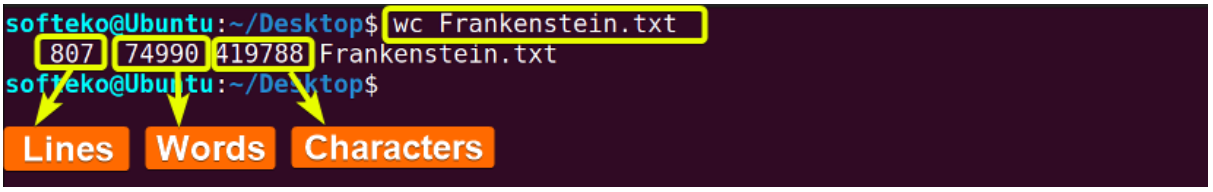
```
wc [OPTION]... [FILE]...
wc [OPTION]... --files0-from=F
```

Useful Options

-l File.txt (shows the number of lines)
-w File.txt (shows the number of words)
-c File.txt (shows the number of characters)

Example

Simply typing **wc Filename.txt** shows the line, word, and character count of the file.



Using **-l** option shows the number of lines of that file.

```
softeko@Ubuntu:~/Desktop$ wc -l Frankenstein.txt
807 Frankenstein.txt
```

Using **-w** option shows the number of words in that file.

```
softeko@Ubuntu:~/Desktop$ wc -w Frankenstein.txt
74990 Frankenstein.txt
```

Using **-c** options shows the number of characters in the file.

```
softeko@Ubuntu:~/Desktop$ wc -c Frankenstein.txt
419788 Frankenstein.txt
```

46. sort command

sort command sorts the content inside a file and shows it in the display. By default **sort** command sort alphabetically. But different conditions can be applied for flexibility.

Synopsis

```
sort [OPTION]... [FILE]...
sort [OPTION]... --files0-from=F
```

Useful Options

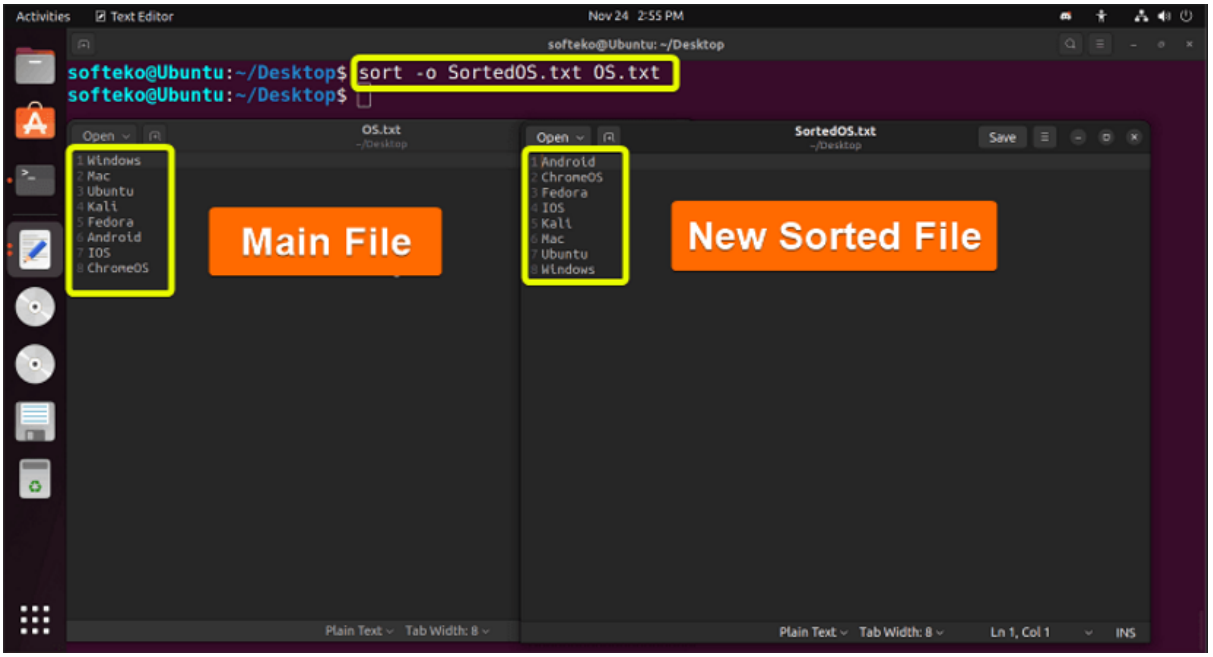
-o Main_file Sorted_File (sort and create a new sorted file)
-r, --reverse File (sort in reverse order for a specific line, sort in horizontal order)
-n, --numeric-sort File (sort in numeric order)
-kNUM (sort by using a key **NUM**, sorting by **NUM** column is possible using this option)

Example

Simply typing this command sorts the contents inside a file alphabetically.

```
softeko@Ubuntu:~/Desktop$ sort OS.txt
Android
ChromeOS
Fedora
IOS
Kali
Mac
Ubuntu
Windows
```

Sorts the content inside a file and creates a new sorted file.



-nk3 option sorts the using the third column numerically.

```
softeko@Ubuntu:~/Desktop$ sort -nk3 purchases.txt
robert91    credit    12.00
steve1      paypal    19.99
steve1      paypal    34.99
robert91    credit    49.50
sherif12    paypal    99.99
amelie54    paypal    195.50
sherif12    credit    456.75
```

Sorted Column

47. cal command

cal command stands for the **cal**endar. It shows calendars in many different formats according to the condition.

Synopsis

```
cal [-3hjy] [-A number] [-B number] [[month] year]
cal [-3hj] [-A number] [-B number] -m month [year]
ncal [-3bhjJpwySM] [-A number] [-B number] [-W number] [-s country_code]
     [[month] year]
ncal [-Jeo] [-A number] [-B number] [year]
ncal [-CN] [-H yyyy-mm-dd] [-d yyyy-mm]
```

Useful Options

-y (shows the whole calendar of the current year)
cal MM YYYY (shows the calendar of the **MM** month of **YYYY** year)
cal YYYY (shows the calendar of the **YYYY** year)
-j (shows the Julian calendar)

Example

Simply typing **cal** shows the current month calendar.

```
softeko@Ubuntu:~/Desktop$ cal
   November 2022
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
```

Using **-y** option shows the whole year calendar.

```
softeko@Ubuntu:~/Desktop$ cal -y
                                2022
      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          1  2  3  4  5          1  2  3  4  5
 2  3  4  5  6  7  8   6  7  8  9 10 11 12   6  7  8  9 10 11 12
 9 10 11 12 13 14 15  13 14 15 16 17 18 19  13 14 15 16 17 18 19
16 17 18 19 20 21 22  20 21 22 23 24 25 26  20 21 22 23 24 25 26
23 24 25 26 27 28 29  27 28                 27 28 29 30 31
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  2   1  2  3  4  5  6  7            1  2  3  4
 3  4  5  6  7  8  9   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


        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  2      1  2  3  4  5  6                1  2  3
```

Typing the month MM and year YYYY shows the calender of that specific month.

```
softeko@Ubuntu:~/Desktop$ cal 12 2022
   December 2022
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
```

Using -**3** command shows the calendar of the current, previous and next month.

```
softeko@Ubuntu:~/Desktop$ cal -3
                              2022
      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
 2  3  4  5  6  7  8   6  7  8  9 10 11 12   4  5  6  7  8  9 10
 9 10 11 12 13 14 15  13 14 15 16 17 18 19  11 12 13 14 15 16 17
16 17 18 19 20 21 22  20 21 22 23 24 25 26  18 19 20 21 22 23 24
23 24 25 26 27 28 29  27 28 29 30           25 26 27 28 29 30 31
30 31
```

48. nano command

Nano is **ano**ther text editor. It is a simple and intuitive text editor that has many different shortcuts and is very light. It comes with the basic Ubuntu install.

Synopsis

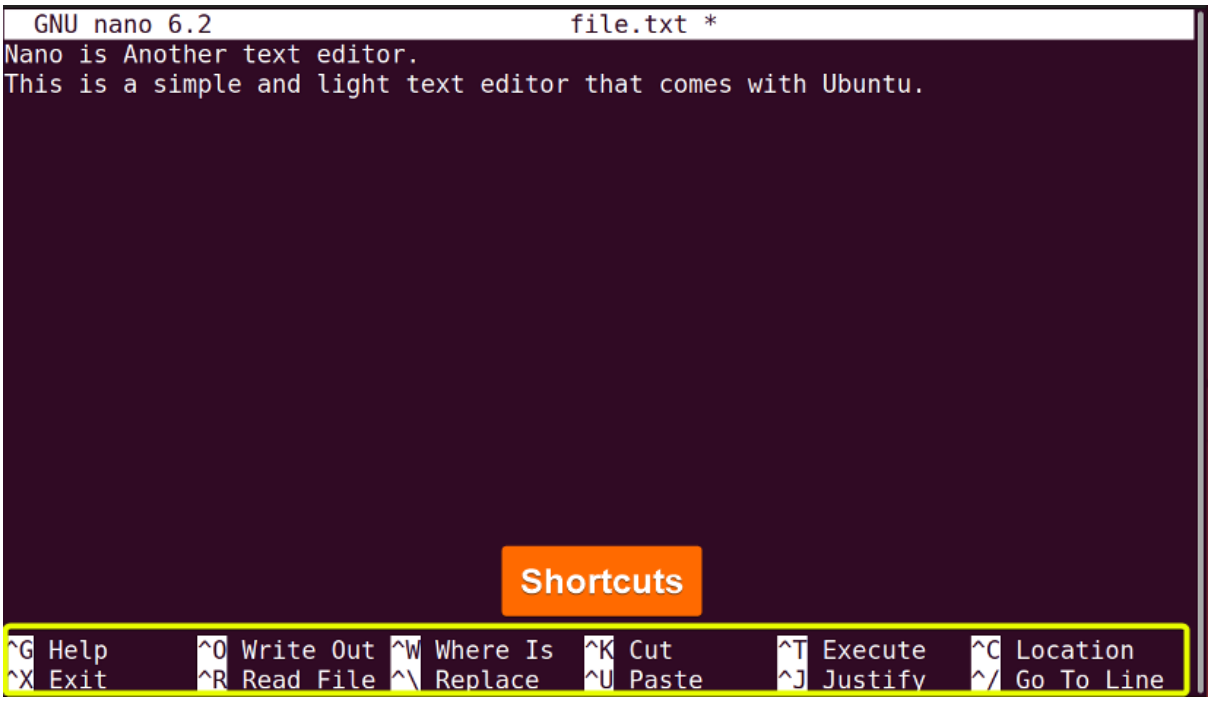
```
nano [options] [[+line[,column]] file]...

nano [options] [[+[crCR](/|?)string] file]...
```

Example

Just typing **nano Filename.txt** starts the nano text editor.

```
softeko@Ubuntu:~/Desktop$ nano file.txt
```

Many shortcuts of nano are available at the bottom of the terminal.



49. du command

du command means **d**isk **u**sage. Using this command the total usage of the disk and the disk usage of the different files are shown in the terminal.

Synopsis

```
du [OPTION]... [FILE]...
du [OPTION]... --files0-from=F
```

Useful Options

-a (shows the total disk usge)
-ah (shows all files disk usage)
--time (shows the last modification time)

Example

Just typing **du** displays the disk usage of the current directory.

```
softeko@Ubuntu:~/Desktop$ du
440     .
softeko@Ubuntu:~/Desktop$
```

-a option can be used to find disk usage of individual files inside the current directory.

```
softeko@Ubuntu:~/Desktop$ du -a
412     ./Frankenstein.txt
4       ./OS.txt
4       ./file.txt
8       ./poem.txt
4       ./purchases.txt
4       ./SortedOS.txt
440     .
```

-**ah** command is used to show all individual files' disk usage in a human-readable format.

```
softeko@Ubuntu:~/Desktop$ du -ah
412K    ./Frankenstein.txt
4.0K    ./OS.txt
4.0K    ./file.txt
8.0K    ./poem.txt
4.0K    ./purchases.txt
4.0K    ./SortedOS.txt
440K    .
```

We can use **--time** option to show the last modification of the directory.

```
softeko@Ubuntu:~/Desktop$ du --time
440     2022-11-24 15:57        .
softeko@Ubuntu:~/Desktop$
```

50. apt-get command

apt-get command is quite similar to the **apt** command. But in **apt-get** common is generally used in the back-end and has backward compatibility. The apt-get (**A**dvanced **P**ackaging **T**ool) command is used for retrieving package and managing (install, upgrade, update, remove) the package.

Synopsis

```
apt-get [-asqdyfmubV] [-o=config_string] [-c=config_file]
        [-t=target_release] [-a=architecture] {update | upgrade |
        dselect-upgrade | dist-upgrade |
        install pkg [{=pkg_version_number | /target_release}]...  |
        remove pkg...  | purge pkg...  |
        source pkg [{=pkg_version_number | /target_release}]...  |
        build-dep pkg [{=pkg_version_number | /target_release}]...  |
        download pkg [{=pkg_version_number | /target_release}]...  |
        check | clean | autoclean | autoremove | {-v | --version} |
        {-h | --help}}
```

Useful Options

apt-get [install, update, upgrade, remove] (retrieve the package and install, update, upgrade or remove the package from the system)
--download-only (downloads a package without installing or updating)
--autoremove, **--auto-remove** (removes unnecessary packages from the system)
--version (shows the version)

Example

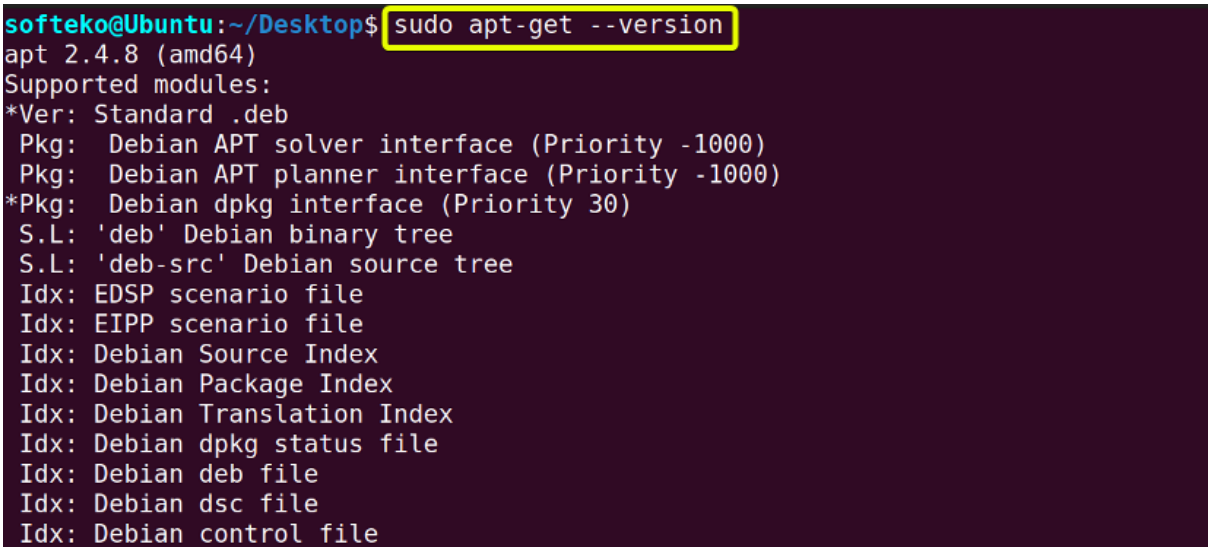
update command is used to update packages in the system.

```
softeko@Ubuntu:~/Desktop$ sudo apt-get update
Hit:1 https://dl.google.com/linux/chrome/deb stable InRelease
Get:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Hit:3 http://bd.archive.ubuntu.com/ubuntu jammy InRelease
Get:4 http://bd.archive.ubuntu.com/ubuntu jammy-updates InRelease [114 kB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security/main amd64 DEP-11 Metadat
a [20.1 kB]
Get:6 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 DEP-11 Met
adata [13.3 kB]
Get:7 http://bd.archive.ubuntu.com/ubuntu jammy-backports InRelease [99.8 kB]
Get:8 http://bd.archive.ubuntu.com/ubuntu jammy-updates/main amd64 DEP-11 Metada
ta [95.2 kB]
Get:9 http://bd.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 DEP-11 Me
tadata [257 kB]
Get:10 http://bd.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 DEP-11
 Metadata [940 B]
Get:11 http://bd.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 DEP-11
 Metadata [11.7 kB]
Fetched 723 kB in 3s (213 kB/s)
Reading package lists... Done
```

We can use **autoremove** option to remove unnecessary packages that are not being used.

```
softeko@Ubuntu:~/Desktop$ sudo apt-get autoremove
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages will be REMOVED:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi i965-va-driver
  intel-media-va-driver libaacs0 libaom3 libass9 libavcodec58 libavformat58
  libavutil56 libbdplus0 libblas3 libbluray2 libbs2b0 libchromaprint1
  libcodec2-1.0 libdav1d5 libflashrom1 libflite1 libftdi1-2 libgme0 libgsm1
  libgstreamer-plugins-bad1.0-0 libigdgmm12 liblilv-0-0 libmfx1 libmysofa1
  libnorm1 libopenmpt0 libpgm-5.3-0 libpostproc55 librabbitmq4 librubberband2
  libserd-0-0 libshine3 libsnappy1v5 libsord-0-0 libsratom-0-0
  libsrt1.4-gnutls libssh-gcrypt-4 libswresample3 libswscale5 libudfread0
  libva-drm2 libva-wayland2 libva-x11-2 libva2 libvdpau1 libvidstab1.1
  libx265-199 libxvidcore4 libzimg2 libzmq5 libzvbi-common libzvbi0
  mesa-va-drivers mesa-vdpau-drivers pocketsphinx-en-us va-driver-all
  vdpau-driver-all
0 upgraded, 0 newly installed, 60 to remove and 4 not upgraded.
After this operation, 198 MB disk space will be freed.
Do you want to continue? [Y/n] y
(Reading database ... 196653 files and directories currently installed.)
Removing chromium-codecs-ffmpeg-extra (1:85.0.4183.83-0ubuntu2) ...
Removing gstreamer1.0-vaapi:amd64 (1.20.1-1ubuntu1) ...
```

Option **--version** shows the version of the apt-get. It also shows the other possible commands on which **--version** command can be used



Conclusion

In this article, we covered the 50 most used Linux commands that a regular user will encounter most of the time. Of course, there are many more of these commands and each of these commands has many more options and functionality available for the user.
You don’t need to memorize every one of them and we already have a free pdf document available that covers all these commands.
We will shortly cover every one of these commands individually and go into the details. In the meantime, you can also bookmark this page in case you need to have a look at any of these commands.