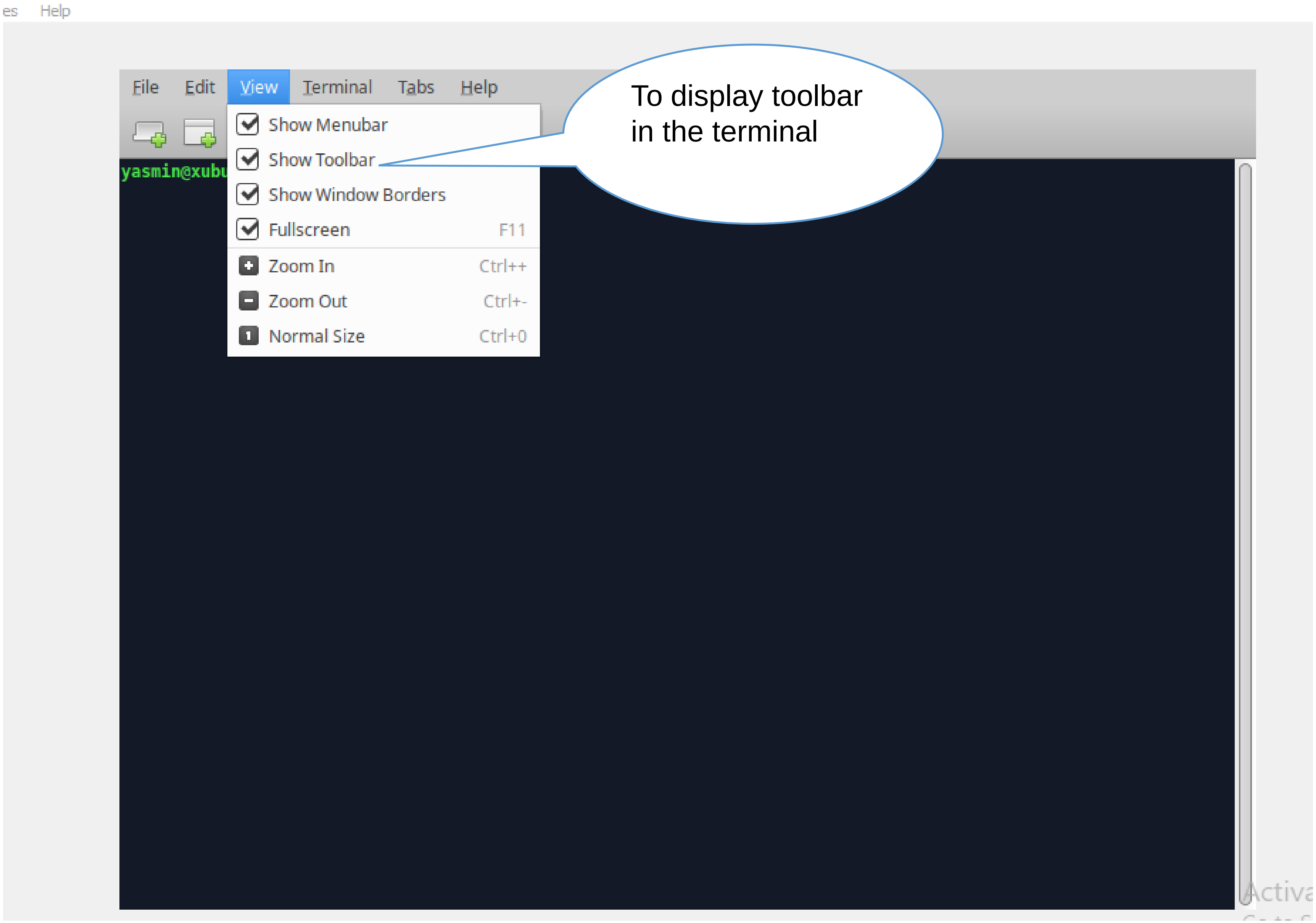
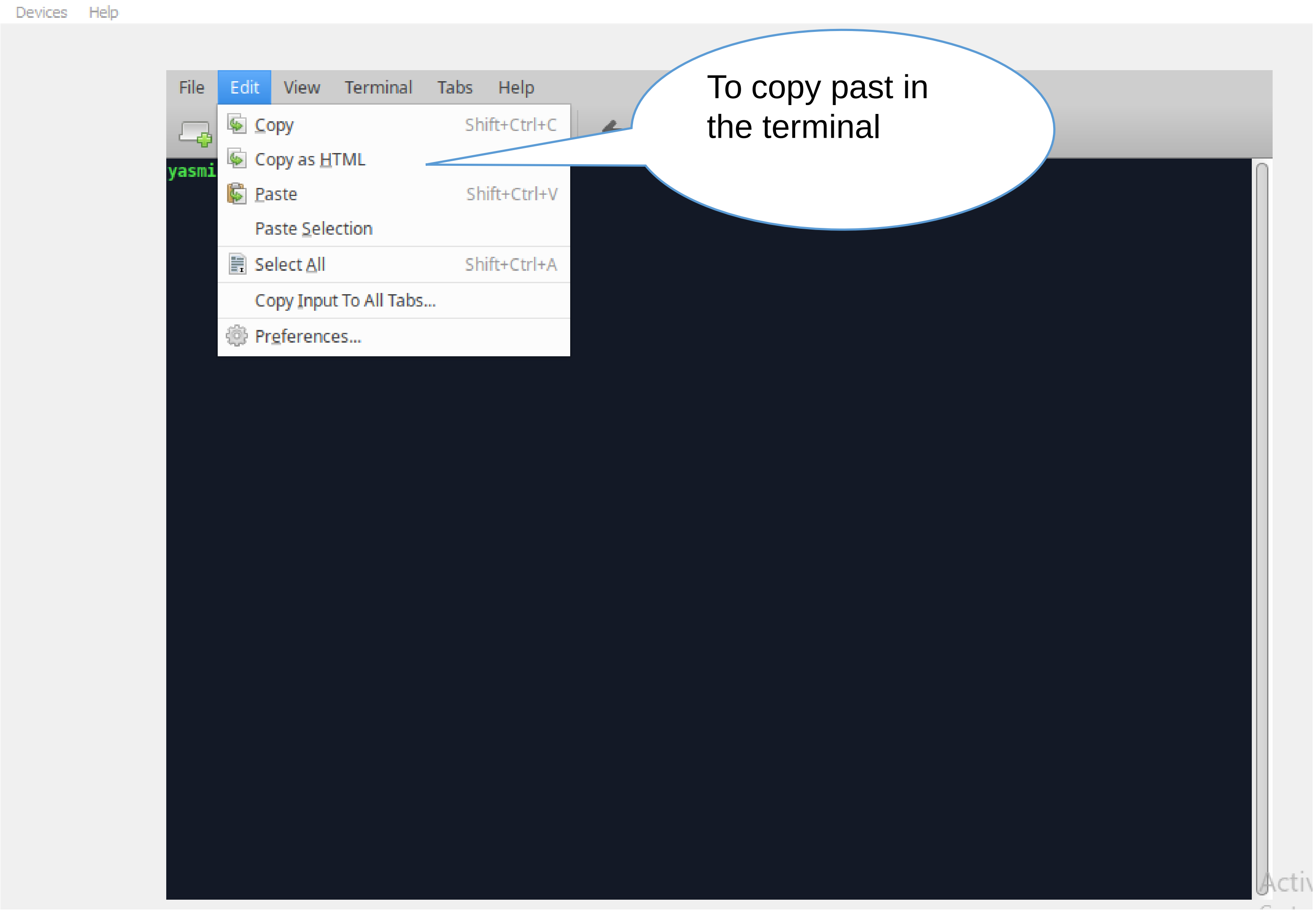
Open sources SW lab lecture 3

Third year CS

Prepared by assistant professor Yasmin Makki Mohialden



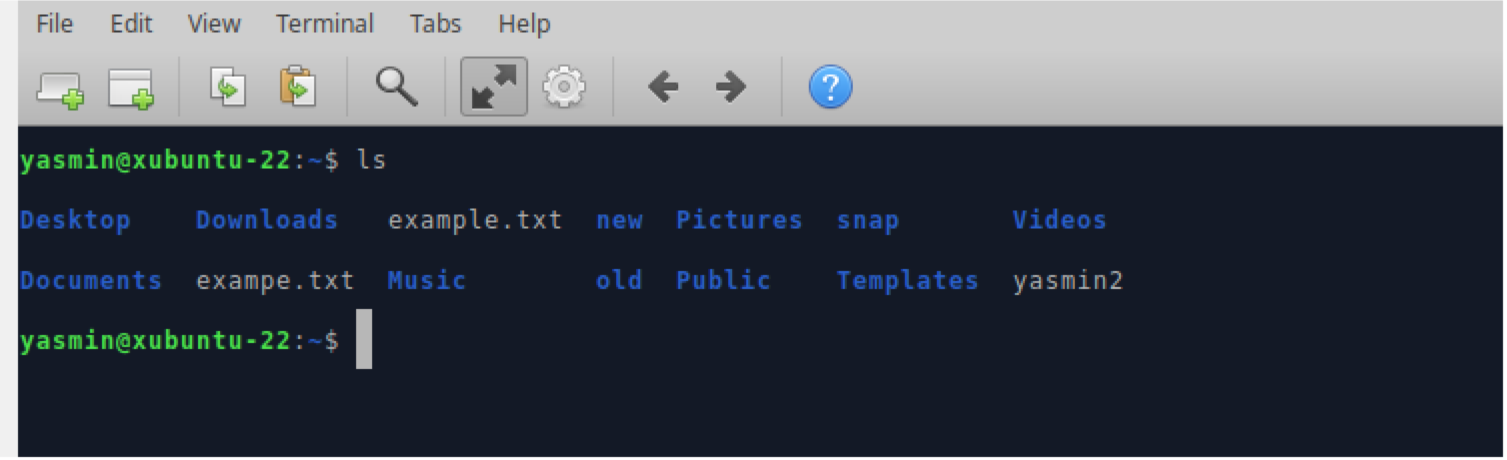
To display toolbar in the terminal



To copy past in
the terminal

1. ls command

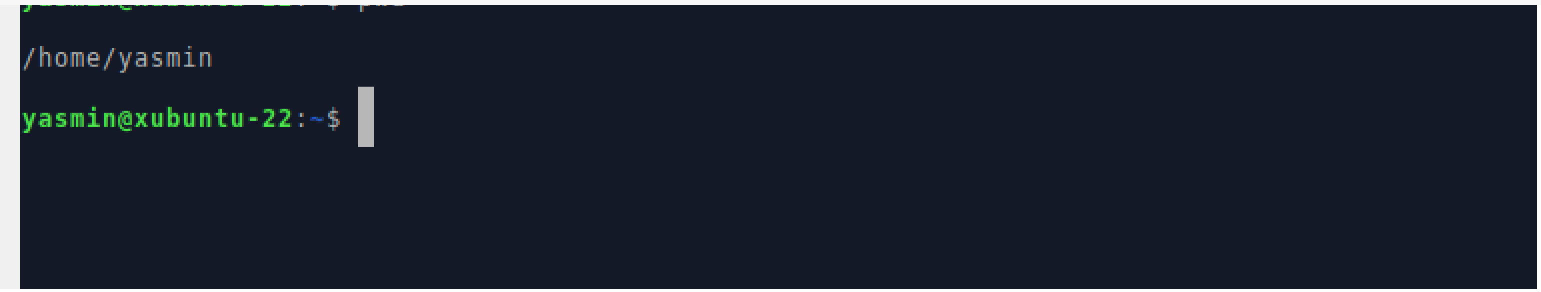
- ls is commonly used to identify files and directories in the working directory.
- It's one of the many frequently-used Linux commands to know.
- See the below image for an example of the output.



```
yasmin@xubuntu-22:~$ ls
Desktop    Downloads   example.txt  new  Pictures  snap       Videos
Documents  exampe.txt  Music        old  Public    Templates  yasmin2
yasmin@xubuntu-22:~$
```

2. pwd command

- pwd prints the current working directory on the terminal.
- It's one of the most commonly used commands.
- It's useful for quickly checking which directory you're in,
- especially if your prompt doesn't display the entire directory.
- See the below pictures for an example of the output.

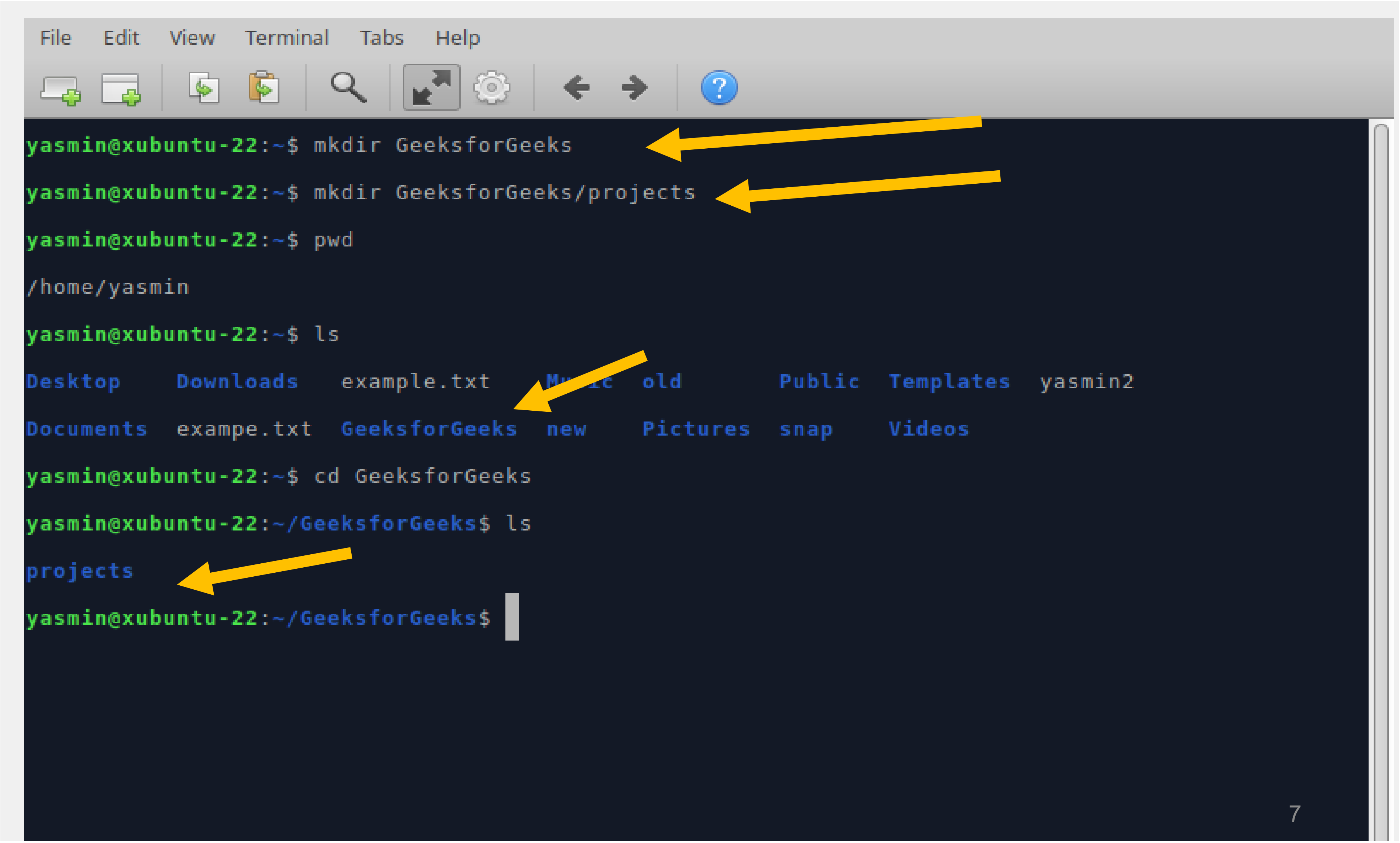


```
/home/yasmin
yasmin@xubuntu-22:~$ 
```

3. mkdir command

- **mkdir** creates directories in the terminal.
- The syntax is **mkdir** <**directory name**> to create a new directory.
- Example: **mkdir GeeksforGeeks** creates a directory named "GeeksforGeeks."
- To create a directory inside an existing directory, use a forward slash: **mkdir GeeksforGeeks/projects**.

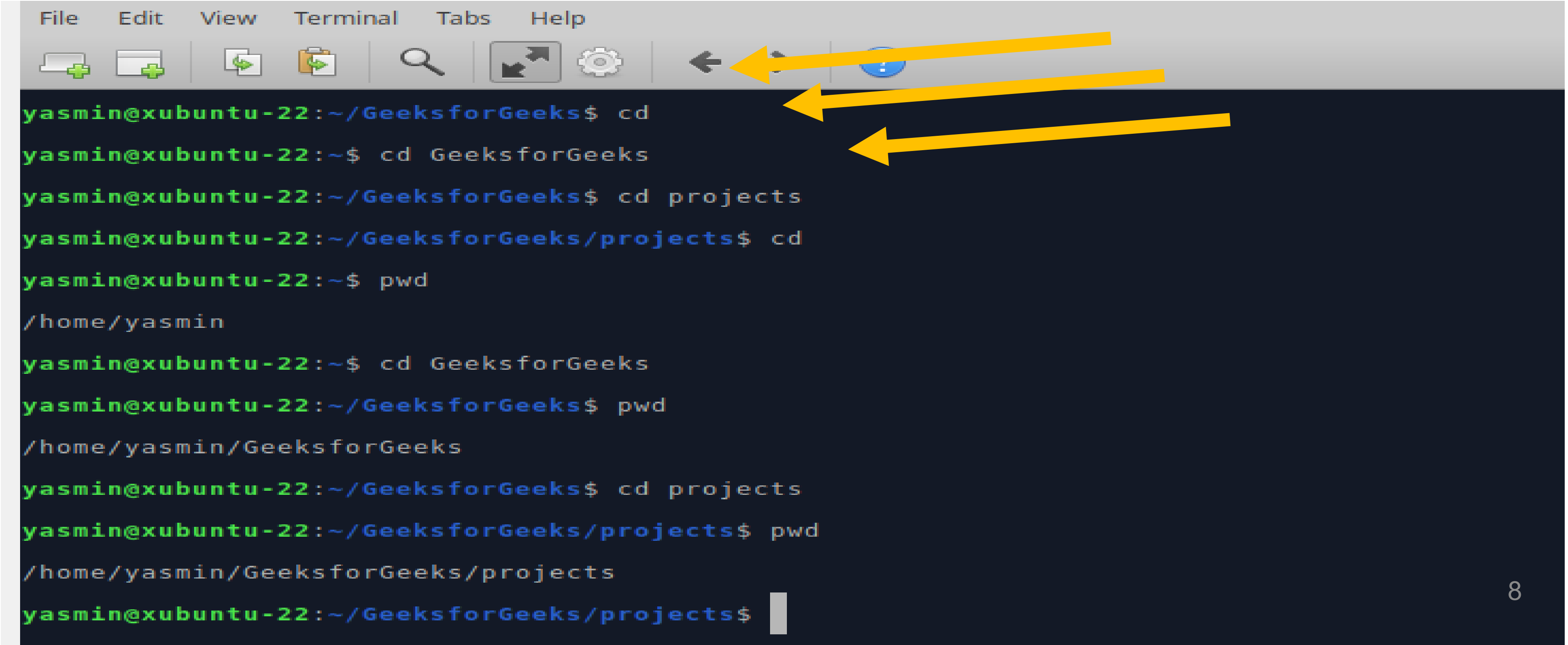
mkdir command



```
yasmin@xubuntu-22:~$ mkdir GeeksforGeeks
yasmin@xubuntu-22:~$ mkdir GeeksforGeeks/projects
yasmin@xubuntu-22:~$ pwd
/home/yasmin
yasmin@xubuntu-22:~$ ls
Desktop    Downloads  example.txt    Music  old       Public  Templates  yasmin2
Documents  exampe.txt  GeeksforGeeks  new    Pictures  snap    Videos
yasmin@xubuntu-22:~$ cd GeeksforGeeks
yasmin@xubuntu-22:~/GeeksforGeeks$ ls
projects
yasmin@xubuntu-22:~/GeeksforGeeks$ 
```

4. cd command

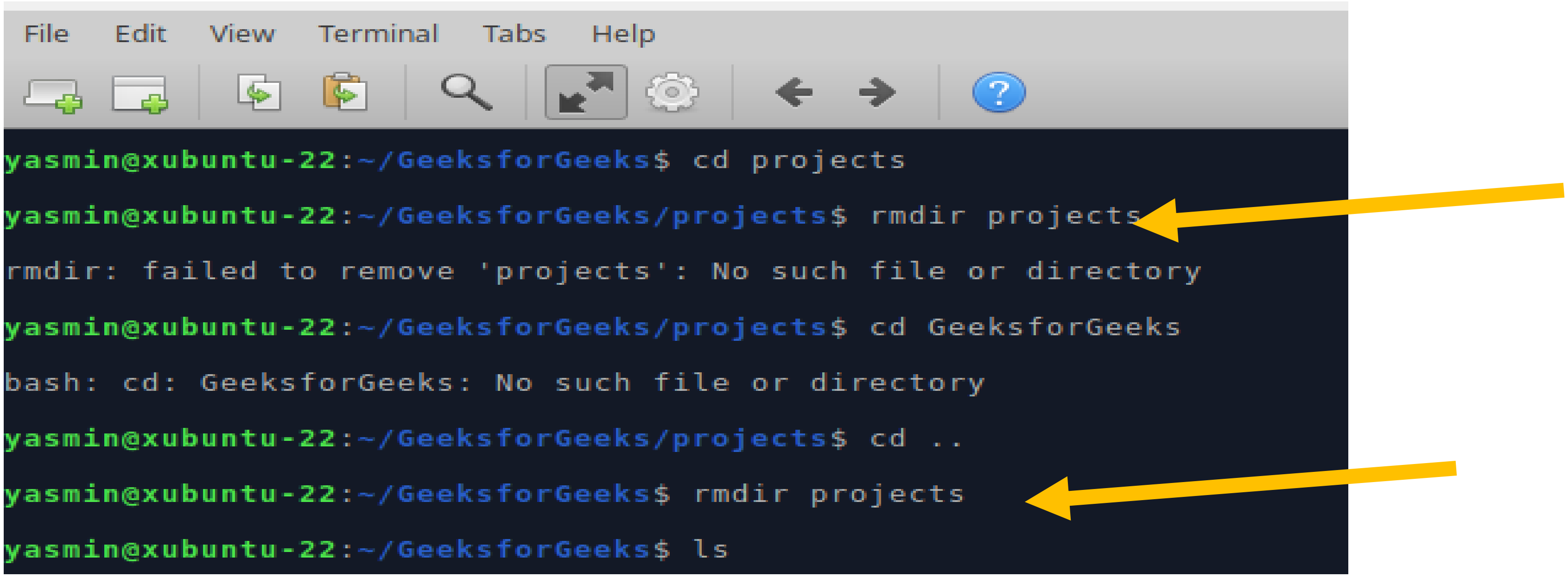
- cd navigates between directories.
- It requires the full path or directory name depending on your current working directory.
- Running cd without options takes you to your home folder.
- Only users with sudo privileges can execute cd.



```
yasmin@xubuntu-22:~/GeeksforGeeks$ cd
yasmin@xubuntu-22:~$ cd GeeksforGeeks
yasmin@xubuntu-22:~/GeeksforGeeks$ cd projects
yasmin@xubuntu-22:~/GeeksforGeeks/projects$ cd
yasmin@xubuntu-22:~$ pwd
/home/yasmin
yasmin@xubuntu-22:~$ cd GeeksforGeeks
yasmin@xubuntu-22:~/GeeksforGeeks$ pwd
/home/yasmin/GeeksforGeeks
yasmin@xubuntu-22:~/GeeksforGeeks$ cd projects
yasmin@xubuntu-22:~/GeeksforGeeks/projects$ pwd
/home/yasmin/GeeksforGeeks/projects
yasmin@xubuntu-22:~/GeeksforGeeks/projects$
```

5. rmdir command

- rmdir command requires sudo privileges in the parent directory to permanently delete an empty directory.

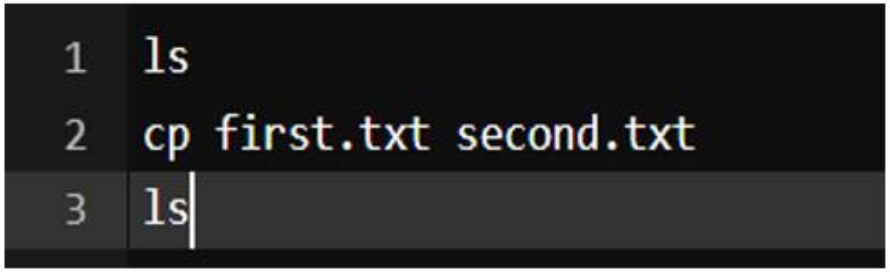


```
yasmin@xubuntu-22:~/GeeksforGeeks$ cd projects
yasmin@xubuntu-22:~/GeeksforGeeks/projects$ rmdir projects
rmdir: failed to remove 'projects': No such file or directory
yasmin@xubuntu-22:~/GeeksforGeeks/projects$ cd GeeksforGeeks
bash: cd: GeeksforGeeks: No such file or directory
yasmin@xubuntu-22:~/GeeksforGeeks/projects$ cd ..
yasmin@xubuntu-22:~/GeeksforGeeks$ rmdir projects
yasmin@xubuntu-22:~/GeeksforGeeks$ ls
```

6. cp command

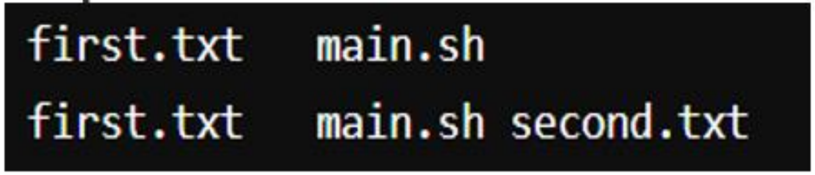
The cp command of Linux is equivalent to copy-paste and cut-paste in Windows.

Command:



```
ls
cp first.txt second.txt
ls
```

Output:



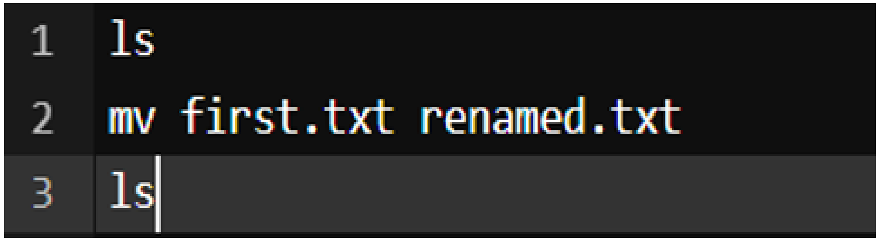
```
first.txt   main.sh
first.txt   main.sh second.txt
```

Here we used **ls** to view the files and then used **cp** to copy the files of ***first.txt*** to ***second.txt*** and again used **ls** command to view the updated files.

7. mv command

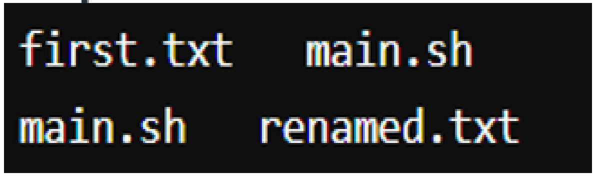
The **mv command** is generally used for renaming the files in Linux.

Command:



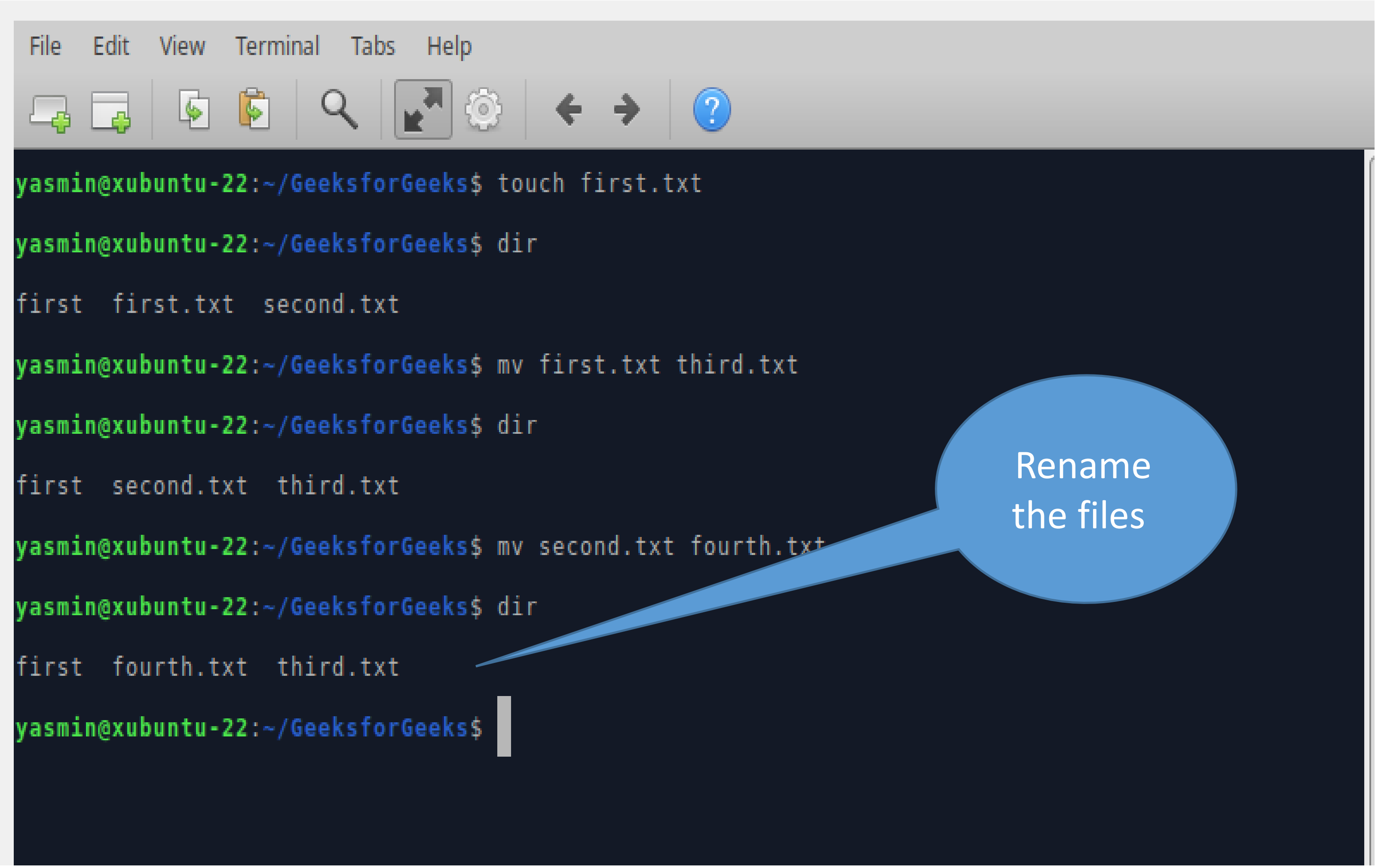
```
ls
mv first.txt renamed.txt
ls
```

Output:



```
first.txt   main.sh
main.sh   renamed.txt
```

Here we used the **ls** command to check the directories and then used **mv <file name> <Renamed file name>** to rename the files, and then again we used the **ls** command to view the renamed file as you can see in the output screenshot.



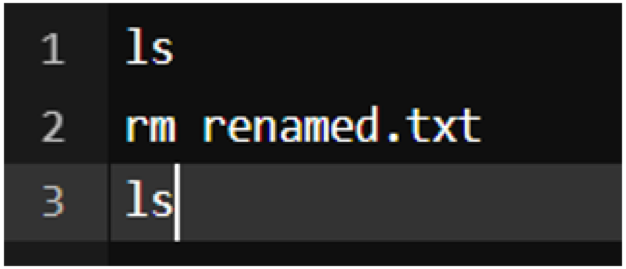
```
yasmin@xubuntu-22:~/GeeksforGeeks$ touch first.txt
yasmin@xubuntu-22:~/GeeksforGeeks$ dir
first  first.txt  second.txt
yasmin@xubuntu-22:~/GeeksforGeeks$ mv first.txt third.txt
yasmin@xubuntu-22:~/GeeksforGeeks$ dir
first  second.txt  third.txt
yasmin@xubuntu-22:~/GeeksforGeeks$ mv second.txt fourth.txt
yasmin@xubuntu-22:~/GeeksforGeeks$ dir
first  fourth.txt  third.txt
yasmin@xubuntu-22:~/GeeksforGeeks$
```

Rename the files

8. rm command

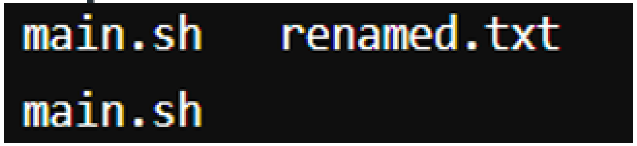
rm command in Linux is generally used to delete the files created in the directory.

Command:



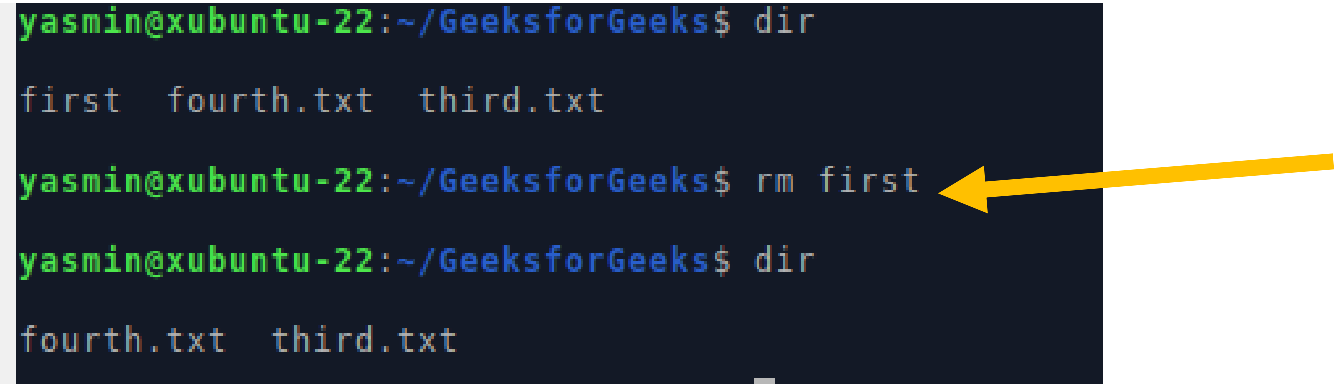
```
ls
rm renamed.txt
ls
```

Output:



```
main.sh    renamed.txt
main.sh
```

You can see as we wrote the **ls** command to view the files in the terminal and then **rm <file name>** to delete the files and again we had the **ls** command to check the update.

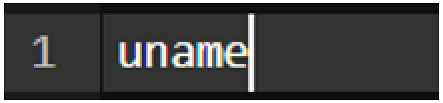


```
yasmin@xubuntu-22:~/GeeksforGeeks$ dir
first  fourth.txt  third.txt
yasmin@xubuntu-22:~/GeeksforGeeks$ rm first
yasmin@xubuntu-22:~/GeeksforGeeks$ dir
fourth.txt  third.txt
```

9. uname command

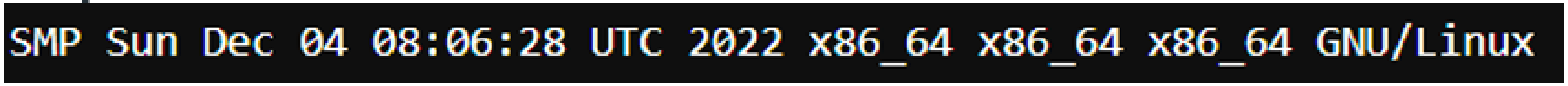
The **uname command** is used to check the complete OS information of the system. Check out the command and the output below

Command:

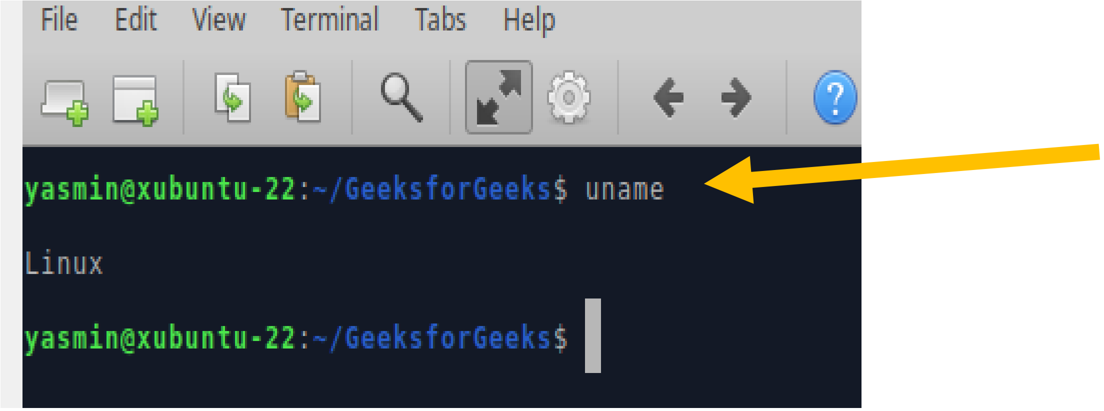


```
uname
```

Output:



```
SMP Sun Dec 04 08:06:28 UTC 2022 x86_64 x86_64 x86_64 GNU/Linux
```

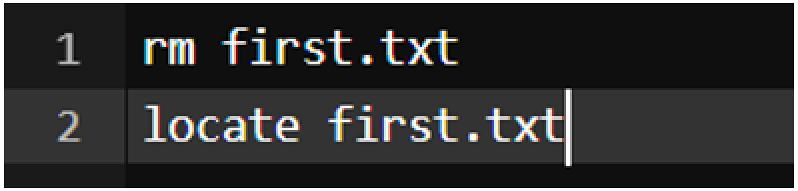


```
yasmin@xubuntu-22:~/GeeksforGeeks$ uname
Linux
yasmin@xubuntu-22:~/GeeksforGeeks$ 
```

10. locate command

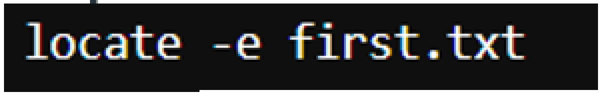
The **locate command** is generally used to locate the files in the database. Use an asterisk (*) to search for content that contains two or more words. As an example: **locate first*file.** This command will search the database for the files that contain these two names **first** and **file.**

Command:



```
rm first.txt
locate first.txt
```

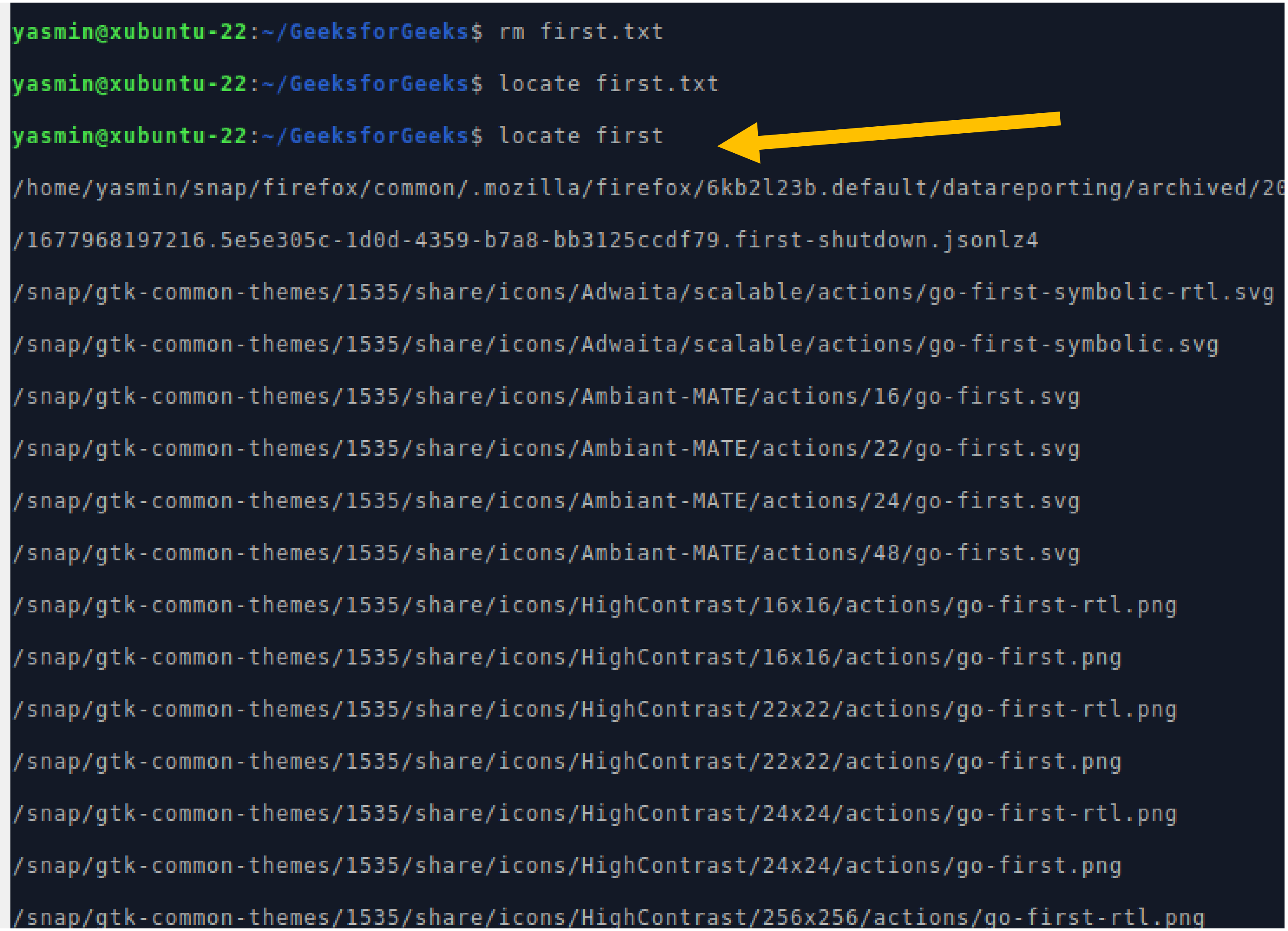
Output:



```
locate -e first.txt
```

We first used the **rm** command to delete the file and then used **locate** command to find the file in the database which in return has given the output with a **-e** as the file was removed.

- The text explains how to use the **locate** command to search for files in the database. The command is generally used to find files by name, and an asterisk (*) can be used to search for content that contains two or more words.
- For example, if you use the command **locate first*file** eht ni selif rof hcraes lliw ti , ."elif" dna "tsrif" :seman owt eseht niatnoc taht esabatad
- So, if there are files named "firstfile", "first_file", "first big file", or "my first file.txt" in the database, the **locate** command will return the paths of all these files that match the search criteria. The asterisk acts as a wildcard character and can represent any number of characters or words in the file name.

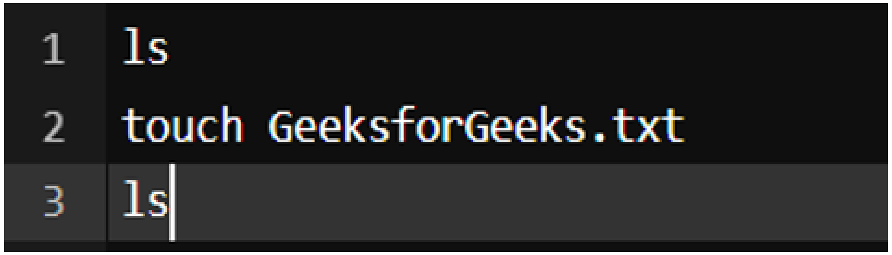


```
yasmin@xubuntu-22:~/GeeksforGeeks$ rm first.txt
yasmin@xubuntu-22:~/GeeksforGeeks$ locate first.txt
yasmin@xubuntu-22:~/GeeksforGeeks$ locate first
/home/yasmin/snap/firefox/common/.mozilla/firefox/6kb2l23b.default/datareporting/archived/20
/1677968197216.5e5e305c-1d0d-4359-b7a8-bb3125ccdf79.first-shutdown.jsonlz4
/snap/gtk-common-themes/1535/share/icons/Adwaita/scalable/actions/go-first-symbolic-rtl.svg
/snap/gtk-common-themes/1535/share/icons/Adwaita/scalable/actions/go-first-symbolic.svg
/snap/gtk-common-themes/1535/share/icons/Ambiant-MATE/actions/16/go-first.svg
/snap/gtk-common-themes/1535/share/icons/Ambiant-MATE/actions/22/go-first.svg
/snap/gtk-common-themes/1535/share/icons/Ambiant-MATE/actions/24/go-first.svg
/snap/gtk-common-themes/1535/share/icons/Ambiant-MATE/actions/48/go-first.svg
/snap/gtk-common-themes/1535/share/icons/HighContrast/16x16/actions/go-first-rtl.png
/snap/gtk-common-themes/1535/share/icons/HighContrast/16x16/actions/go-first.png
/snap/gtk-common-themes/1535/share/icons/HighContrast/22x22/actions/go-first-rtl.png
/snap/gtk-common-themes/1535/share/icons/HighContrast/22x22/actions/go-first.png
/snap/gtk-common-themes/1535/share/icons/HighContrast/24x24/actions/go-first-rtl.png
/snap/gtk-common-themes/1535/share/icons/HighContrast/24x24/actions/go-first.png
/snap/gtk-common-themes/1535/share/icons/HighContrast/256x256/actions/go-first-rtl.png
```

11. touch command

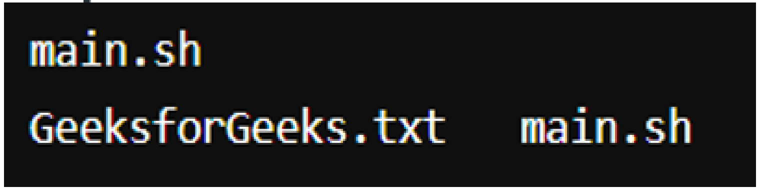
The **touch command** creates an empty file when put in the terminal in this format as touch **<file name>**

Command:



```
ls
touch GeeksforGeeks.txt
ls
```

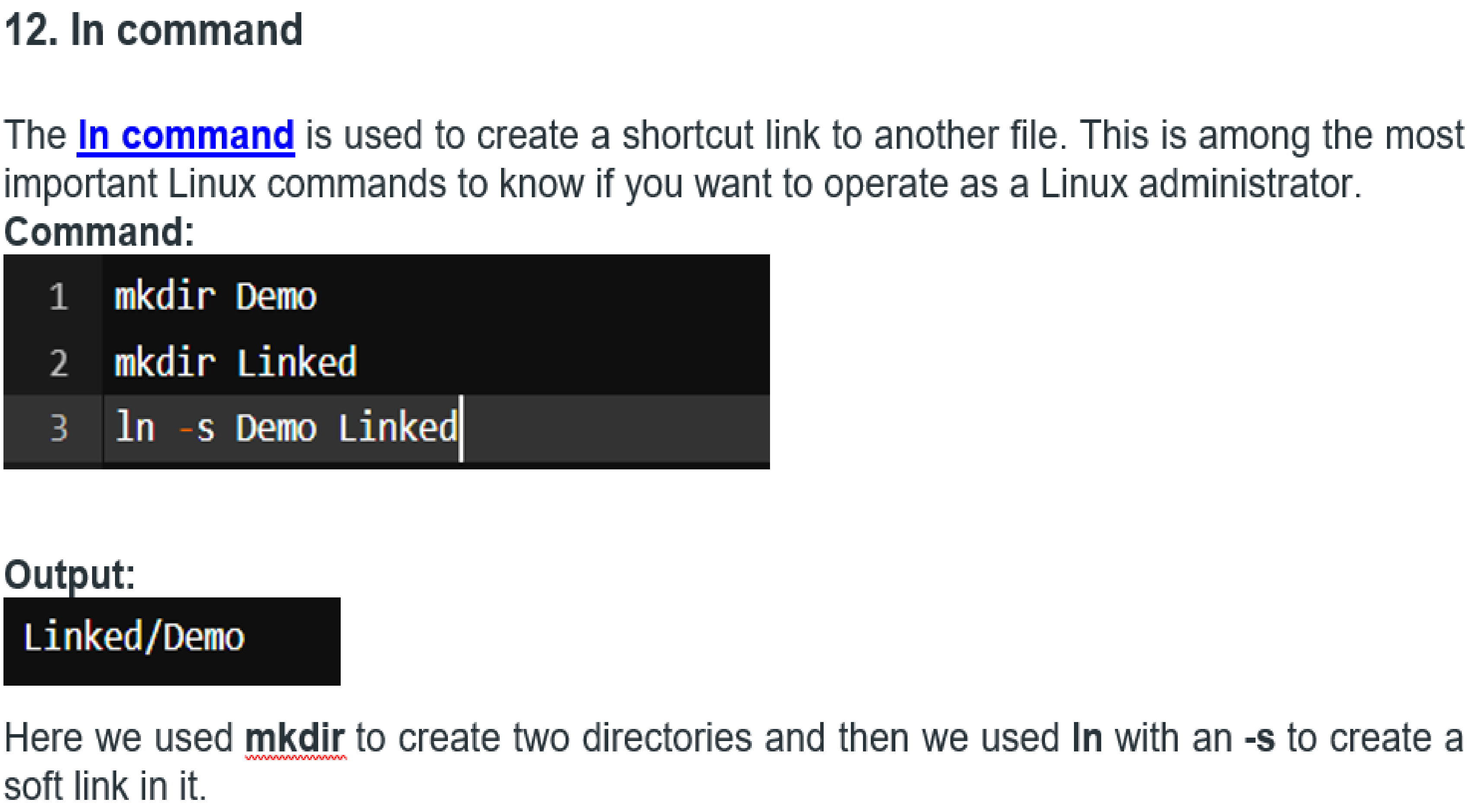
Output:



```
main.sh
GeeksforGeeks.txt   main.sh
```

We used the **ls** command to check the current directories in the terminal and then used the **touch** command to create an empty file and then again we used **ls** to find out the created file in the terminal.

12. ln command



12. ln command

The **ln command** is used to create a shortcut link to another file. This is among the most important Linux commands to know if you want to operate as a Linux administrator.

Command:

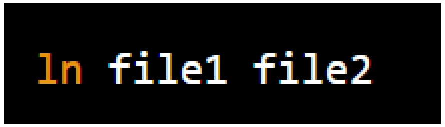
```
mkdir Demo
mkdir Linked
ln -s Demo Linked
```

Output:

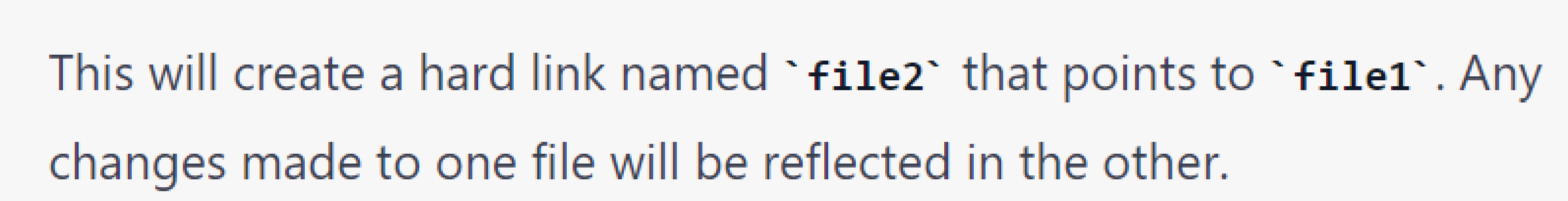
```
Linked/Demo
```

Here we used **mkdir** to create two directories and then we used **ln** with an **-s** to create a soft link in it.

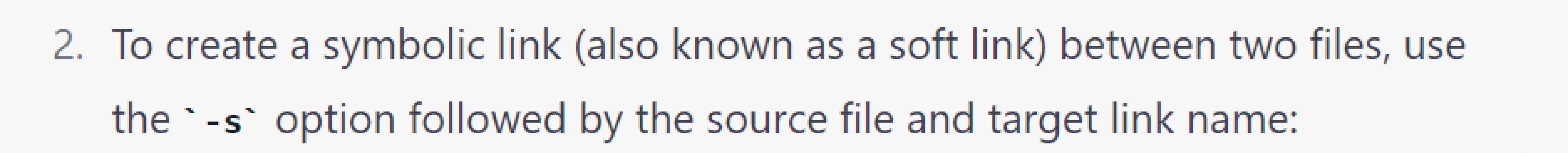
To create a hard link between two files, use the following syntax:



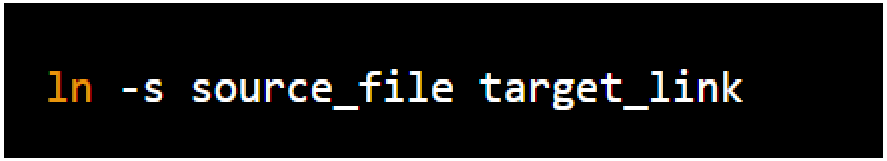
```
ln file1 file2
```



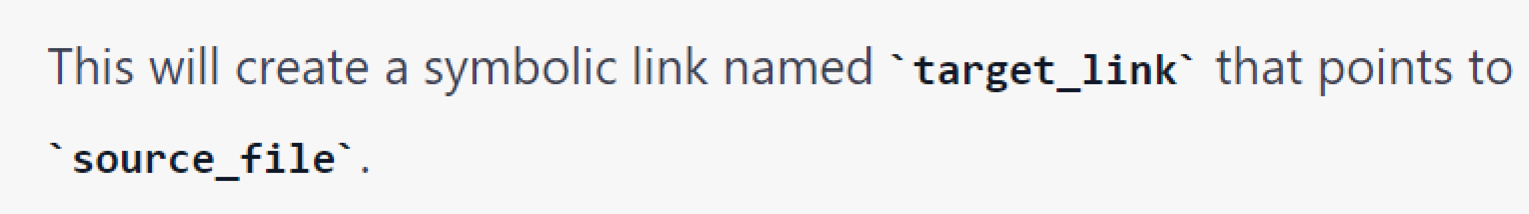
This will create a hard link named `file2` that points to `file1`. Any changes made to one file will be reflected in the other.



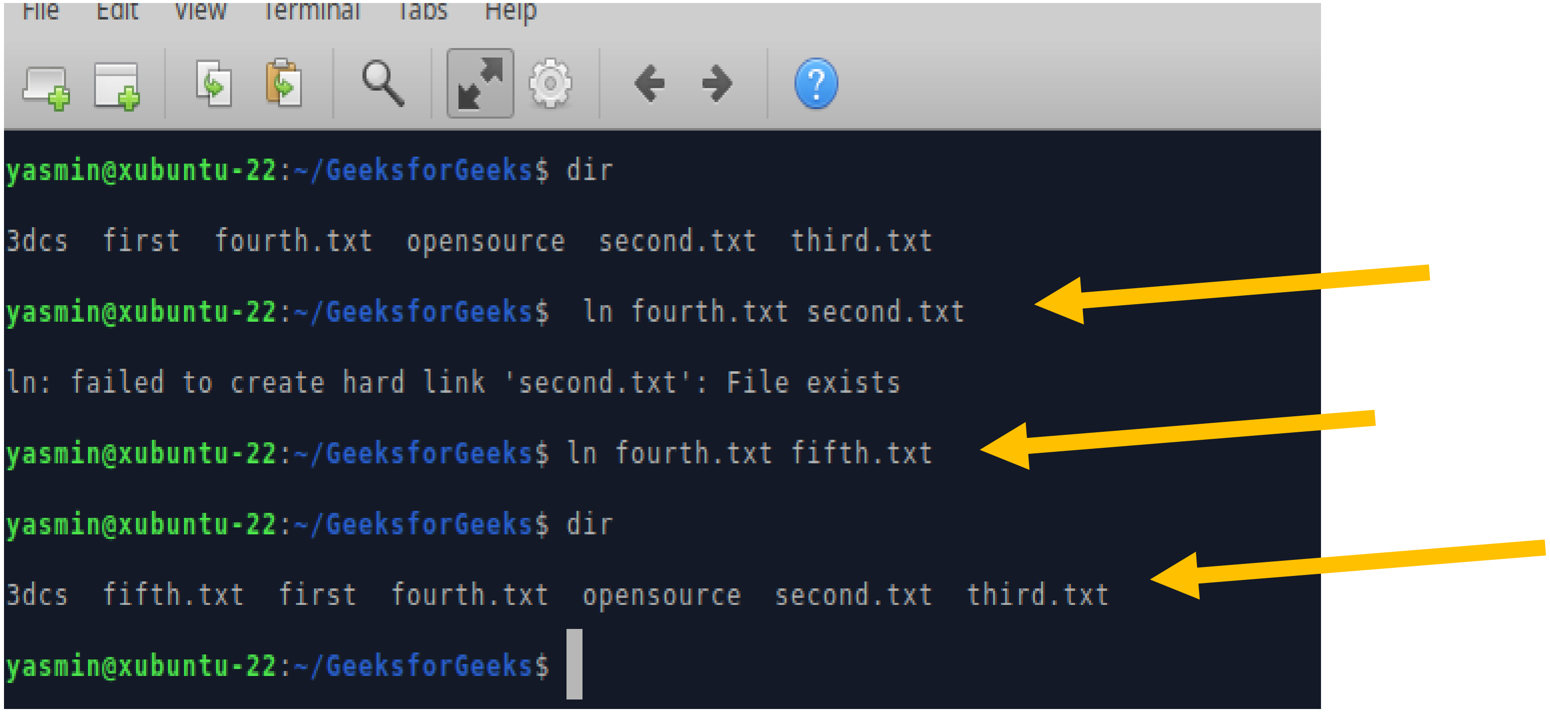
2. To create a symbolic link (also known as a soft link) between two files, use the `-s` option followed by the source file and target link name:



```
ln -s source_file target_link
```



This will create a symbolic link named `target_link` that points to `source_file`.

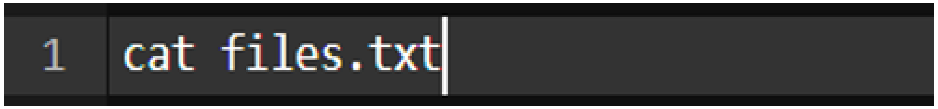


```
yasmin@xubuntu-22:~/GeeksforGeeks$ dir
3dcs  first  fourth.txt  opensource  second.txt  third.txt
yasmin@xubuntu-22:~/GeeksforGeeks$  ln fourth.txt second.txt
ln: failed to create hard link 'second.txt': File exists
yasmin@xubuntu-22:~/GeeksforGeeks$ ln fourth.txt fifth.txt
yasmin@xubuntu-22:~/GeeksforGeeks$ dir
3dcs  fifth.txt  first  fourth.txt  opensource  second.txt  third.txt
yasmin@xubuntu-22:~/GeeksforGeeks$ 
```

13. cat command

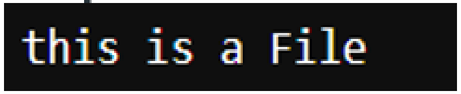
The **cat command** is the simplest command to use when you want to see the contents of a particular file. The only issue is that it simply unloads the entire file to your terminal. If you want to navigate around a huge file, should use **less** command alternatively.

Command:

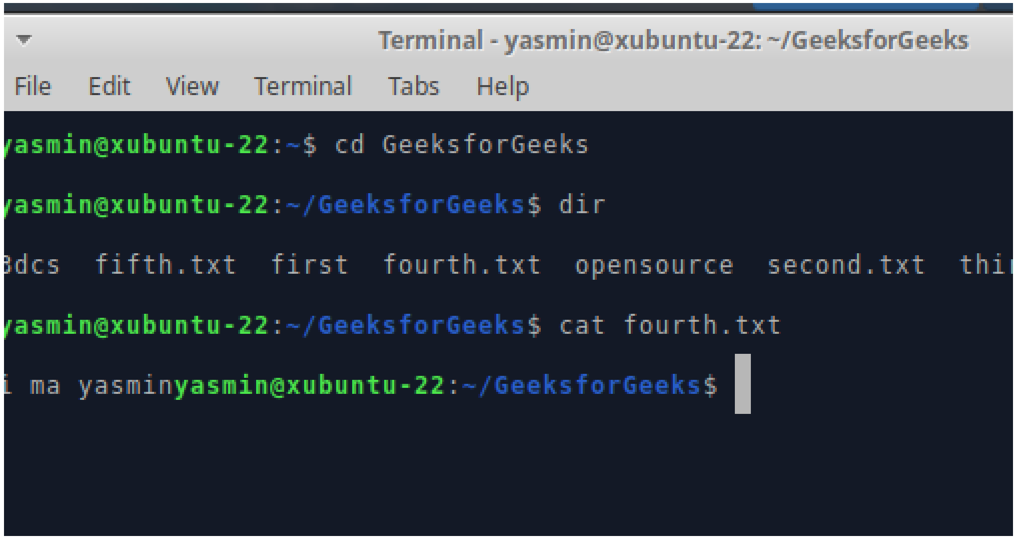


```
cat files.txt
```

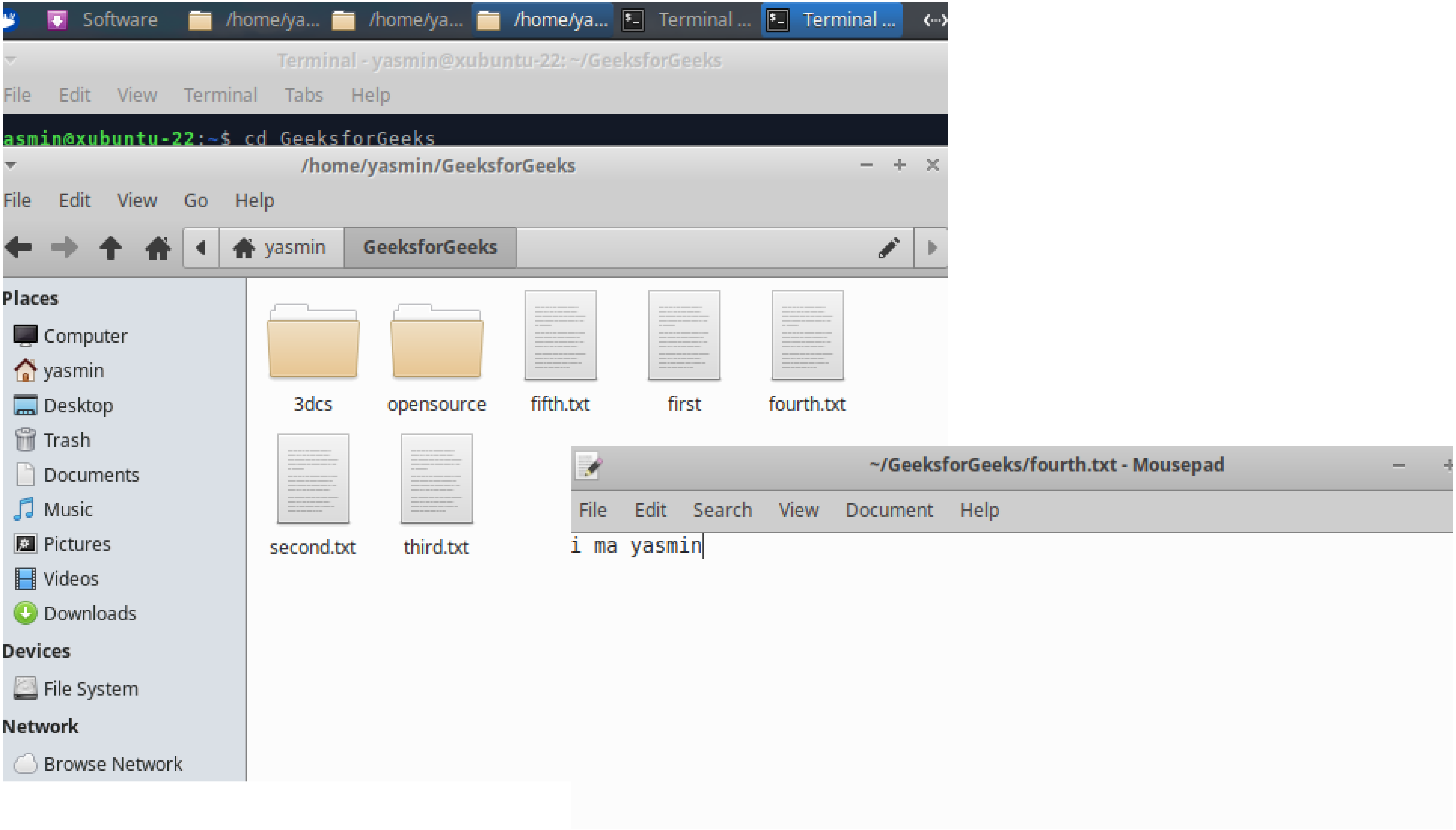
Output:



```
this is a File
```



```
yasmin@xubuntu-22:~$ cd GeeksforGeeks
yasmin@xubuntu-22:~/GeeksforGeeks$ dir
3dcs  fifth.txt  first  fourth.txt  opensource  second.txt  thi
yasmin@xubuntu-22:~/GeeksforGeeks$ cat fourth.txt
i ma yasminyasmin@xubuntu-22:~/GeeksforGeeks$
```

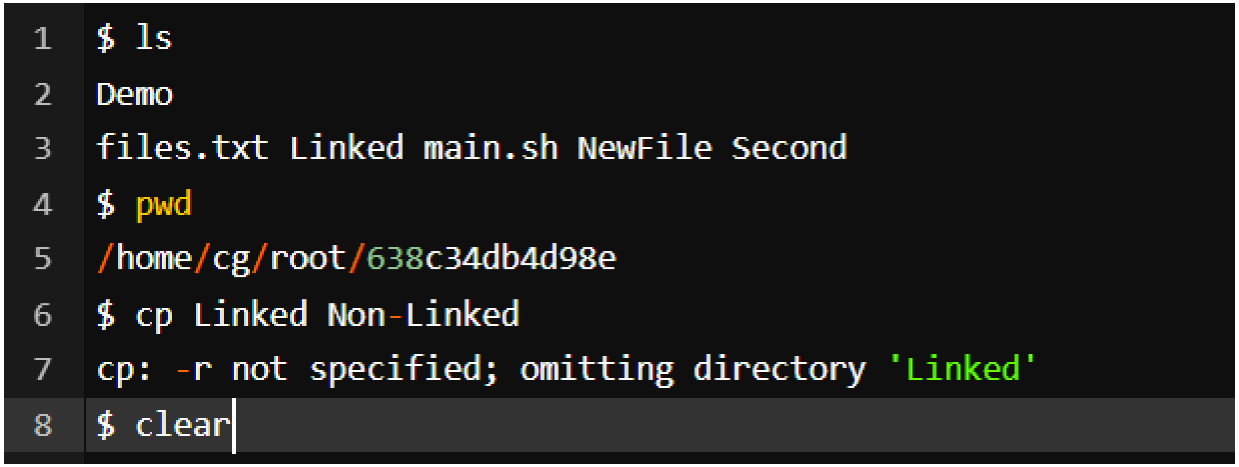


```
asmin@xubuntu-22:~$ cd GeeksforGeeks
```

```
i ma yasmin
```

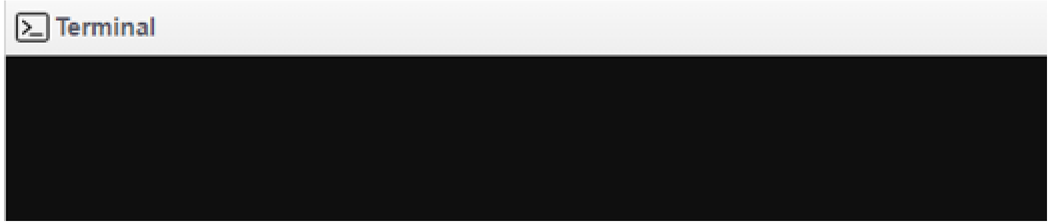
14. clear command

The **clear command** is a standard command to clear the terminal screen.
Command: *This was the terminal before the command.



```
$ ls
Demo
files.txt Linked main.sh NewFile Second
$ pwd
/home/cg/root/638c34db4d98e
$ cp Linked Non-Linked
cp: -r not specified; omitting directory 'Linked'
$ clear
```

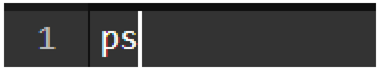
Output:



15. ps command

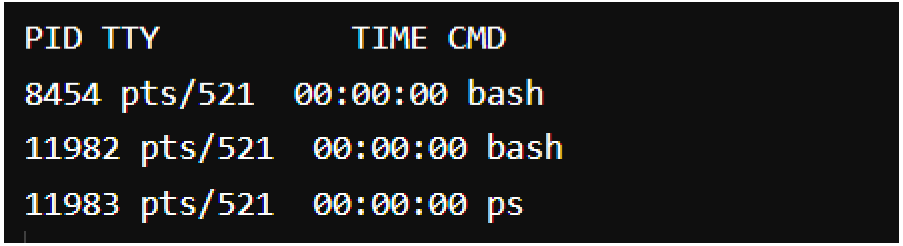
ps command in Linux is used to check the active processes in the terminal.

Command:

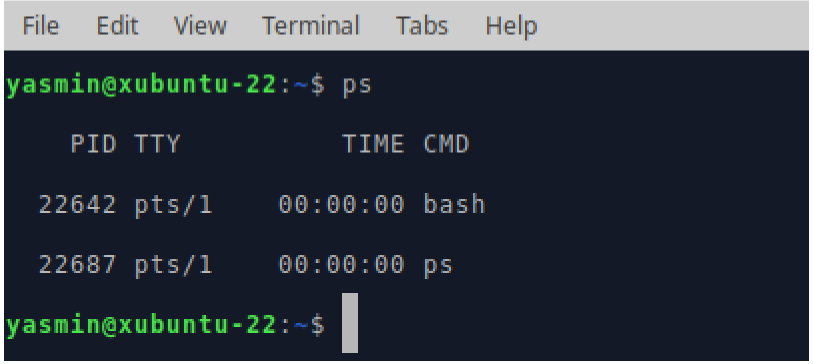


```
ps
```

Output:

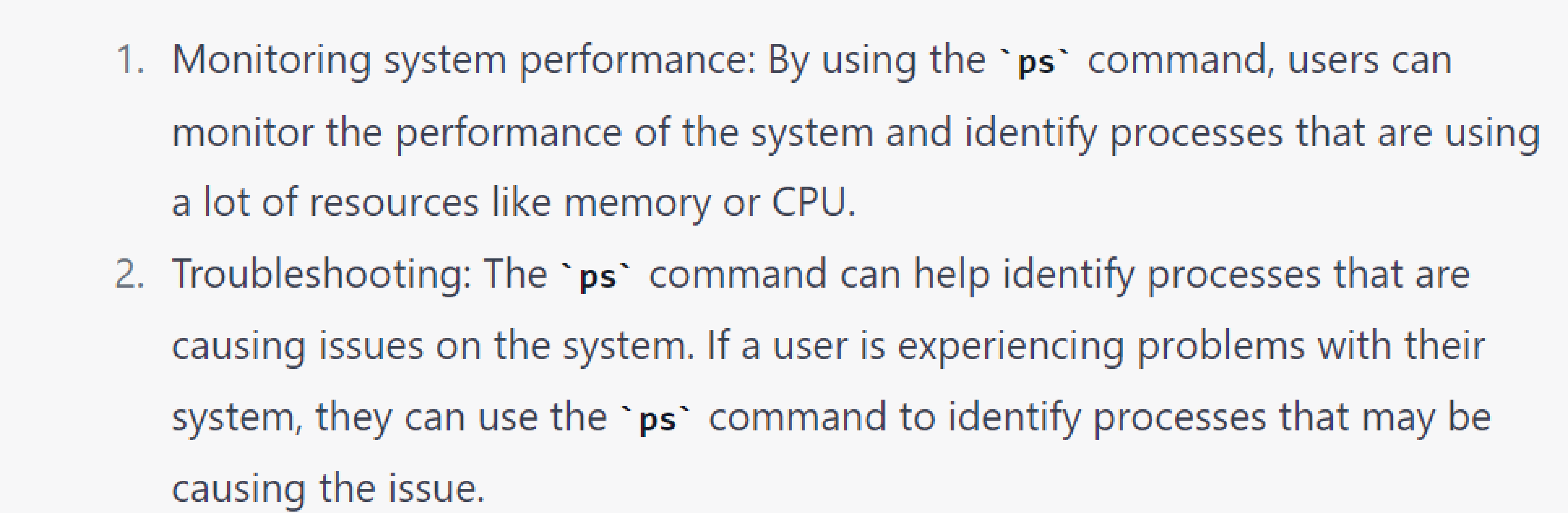


```
PID TTY          TIME CMD
8454 pts/521   00:00:00 bash
11982 pts/521  00:00:00 bash
11983 pts/521  00:00:00 ps
```

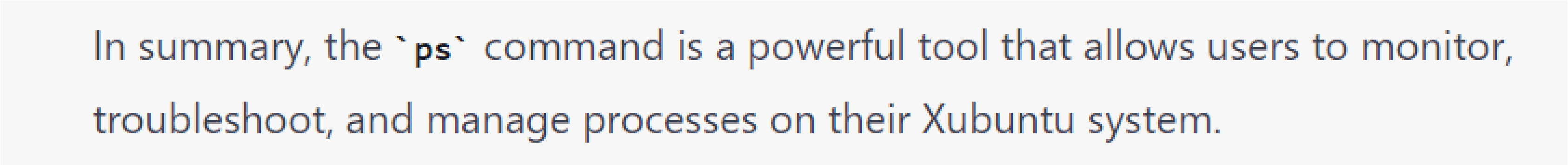


```
yasmin@xubuntu-22:~$ ps
    PID TTY          TIME CMD
  22642 pts/1    00:00:00 bash
  22687 pts/1    00:00:00 ps
yasmin@xubuntu-22:~$
```

The `` `ps` `` command can be useful for several reasons, including:



1. Monitoring system performance: By using the `` `ps` `` command, users can monitor the performance of the system and identify processes that are using a lot of resources like memory or CPU.
2. Troubleshooting: The `` `ps` `` command can help identify processes that are causing issues on the system. If a user is experiencing problems with their system, they can use the `` `ps` `` command to identify processes that may be causing the issue.



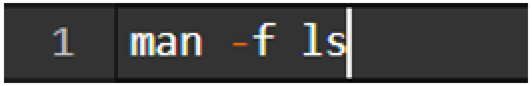
In summary, the `` `ps` `` command is a powerful tool that allows users to monitor, troubleshoot, and manage processes on their Xubuntu system.

16. man command

The man command displays a user manual for any commands or utilities available in the Terminal, including their name, description, and options.
Command to view the full manual:

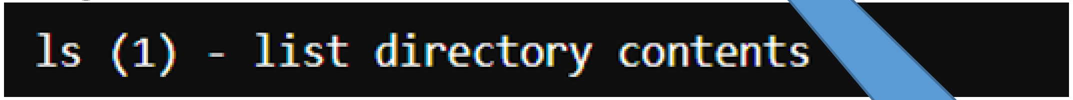
```
man <command name>
```

For example, suppose you want to look up the manual for the ls command: **man ls**
Command:

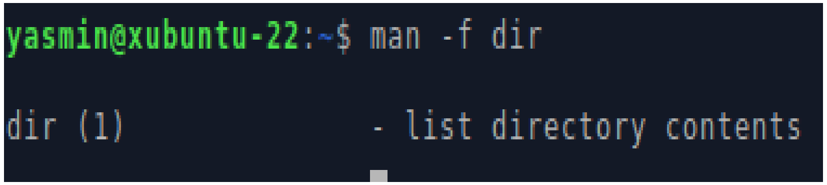


```
man -f ls
```

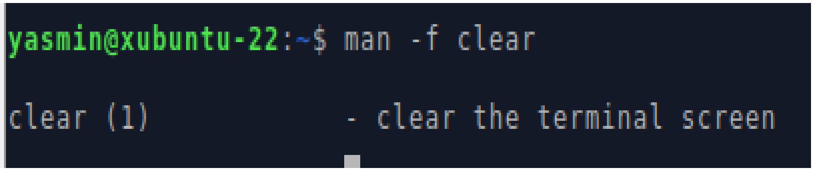
Output:



```
ls (1) - list directory contents
```

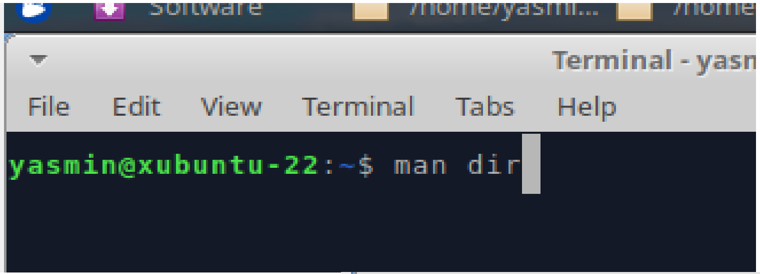


```
yasmin@xubuntu-22:~$ man -f dir

dir (1)              - list directory contents
```

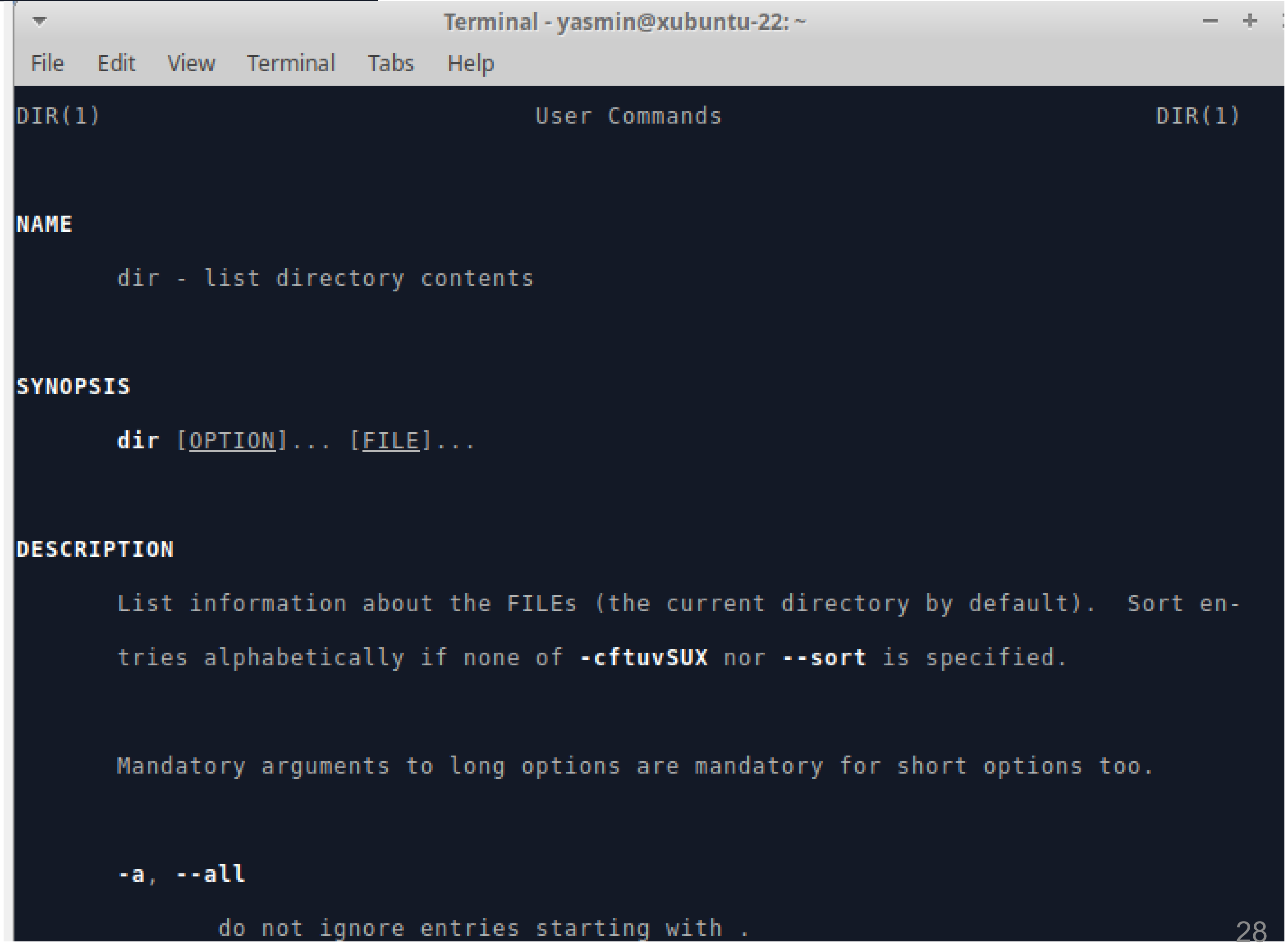


```
yasmin@xubuntu-22:~$ man -f clear

clear (1)            - clear the terminal screen
```

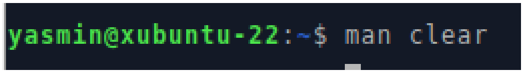
Display information about the command



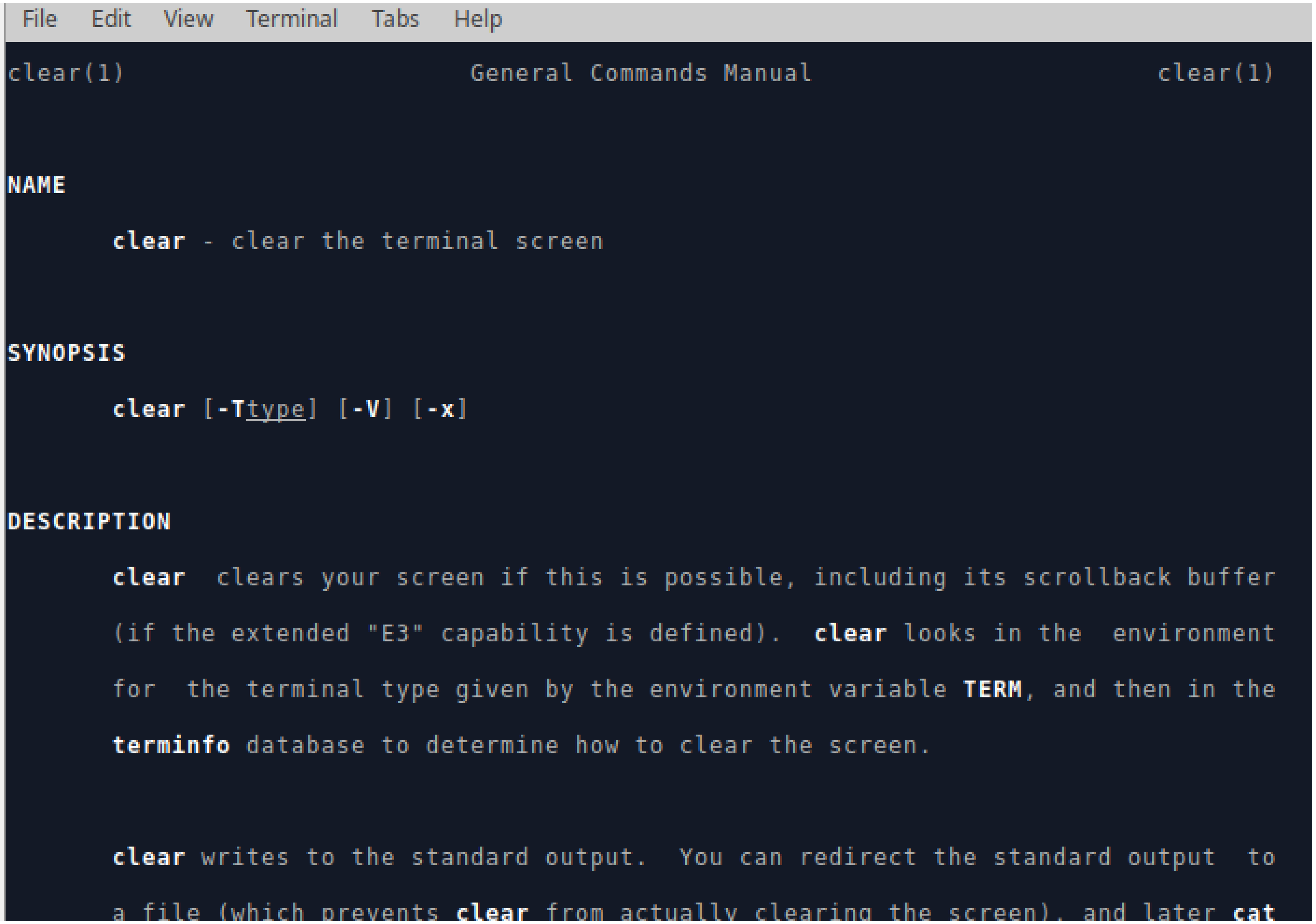
```
yasmin@xubuntu-22:~$ man dir
```



```
DIR(1)                       User Commands                      DIR(1)

NAME
       dir - list directory contents

SYNOPSIS
       dir [OPTION]... [FILE]...

DESCRIPTION
       List information about the FILEs (the current directory by default).  Sort en-
       tries alphabetically if none of -cftuvSUX nor --sort is specified.

       Mandatory arguments to long options are mandatory for short options too.

       -a, --all
              do not ignore entries starting with .
```



```
yasmin@xubuntu-22:~$ man clear
```



```
clear(1)                   General Commands Manual                   clear(1)

NAME
       clear - clear the terminal screen

SYNOPSIS
       clear [-Ttype] [-V] [-x]

DESCRIPTION
       clear  clears your screen if this is possible, including its scrollback buffer
       (if the extended "E3" capability is defined).  clear looks in the  environment
       for  the terminal type given by the environment variable TERM, and then in the
       terminfo database to determine how to clear the screen.

       clear writes to the standard output.  You can redirect the standard output  to
       a file (which prevents clear from actually clearing the screen), and later cat
```

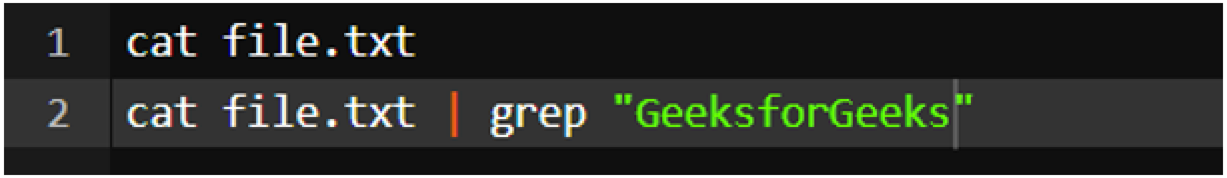
17. grep command

The grep command is used to find a specific string in a series of outputs. For example, if you want to find a string in a file, you can use the syntax: **<Any command with output> | grep “<string to find> “**

For Example:

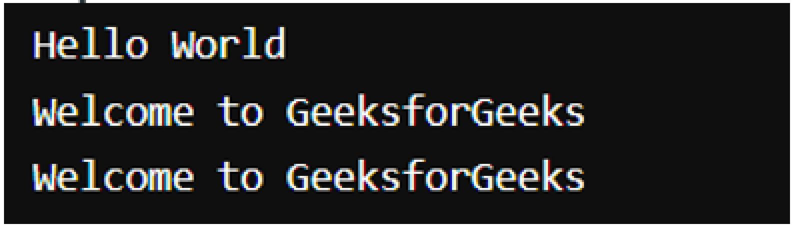
```
cat Files.txt | grep “new”
```

Command:



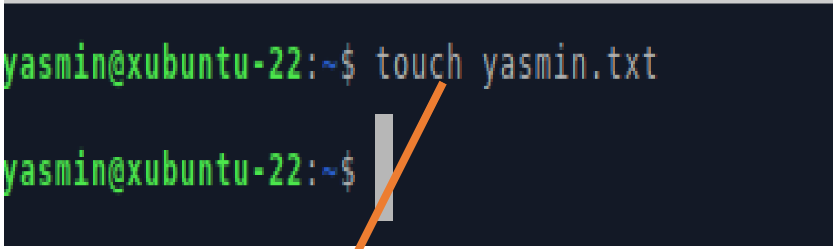
```
cat file.txt
cat file.txt | grep "GeeksforGeeks"
```

Output:

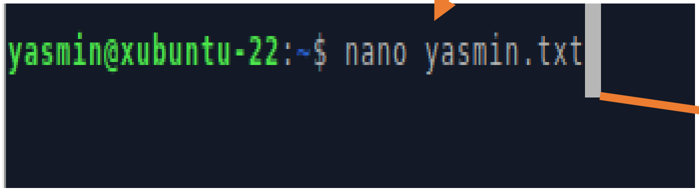


```
Hello World
Welcome to GeeksforGeeks
Welcome to GeeksforGeeks
```

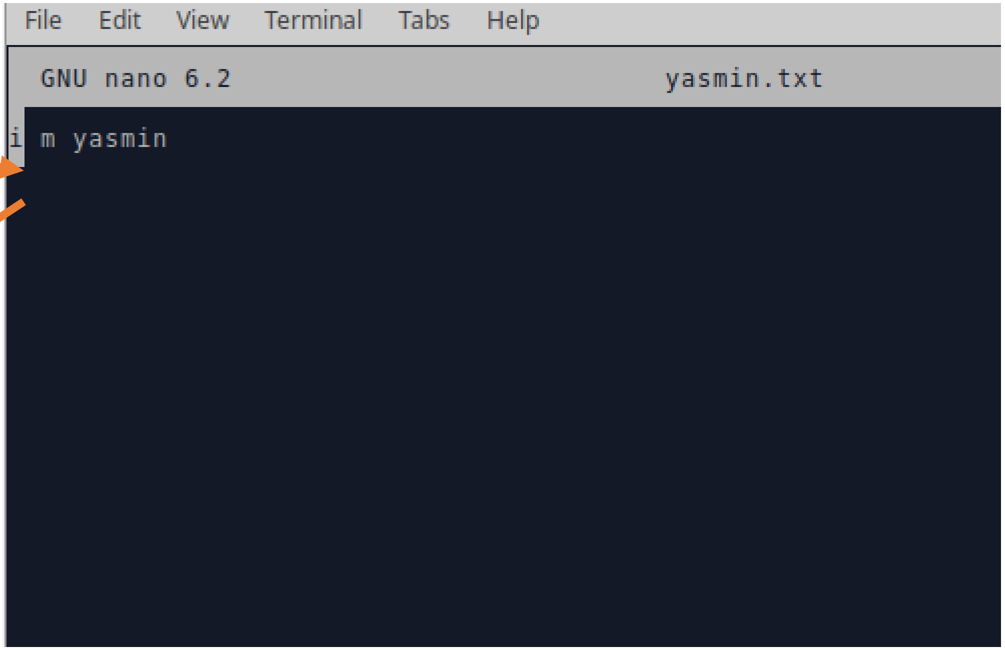
In this command, we first used **cat <file name>** to view the content of the file, and then we used **cat <file name> | grep “string”** to check the string in it.



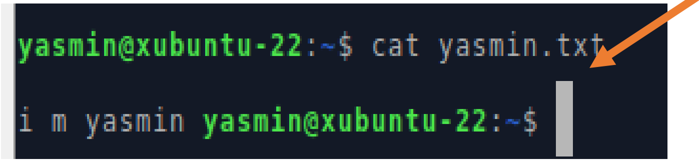
```
yasmin@xubuntu-22:~$ touch yasmin.txt
yasmin@xubuntu-22:~$
```



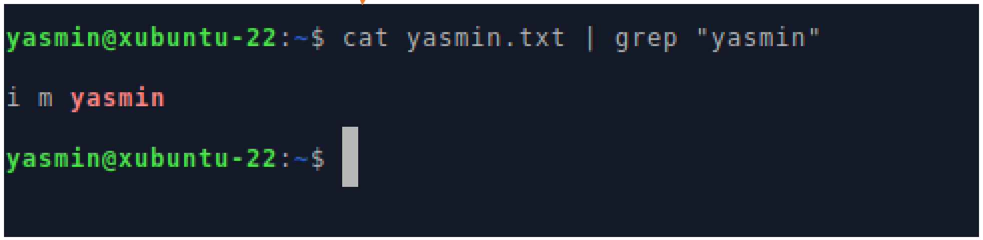
```
yasmin@xubuntu-22:~$ nano yasmin.txt
```



```
GNU nano 6.2                              yasmin.txt
i m yasmin
```



```
yasmin@xubuntu-22:~$ cat yasmin.txt
i m yasmin yasmin@xubuntu-22:~$
```

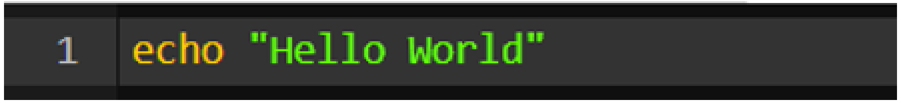


```
yasmin@xubuntu-22:~$ cat yasmin.txt | grep "yasmin"
i m yasmin
yasmin@xubuntu-22:~$
```

18. echo command

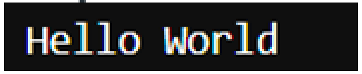
echo command in Linux is specially used to print something in the terminal

Command:

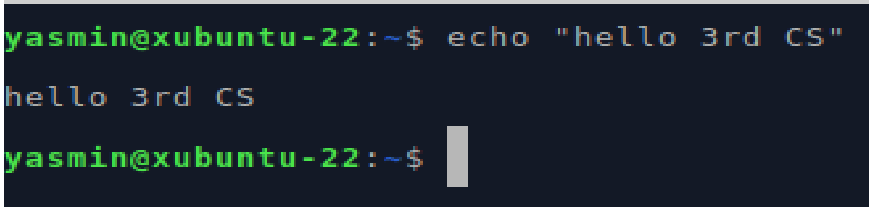


```
echo "Hello World"
```

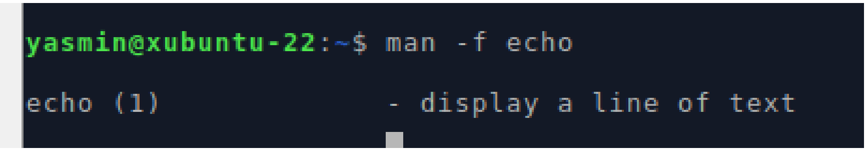
Output:



```
Hello World
```



```
yasmin@xubuntu-22:~$ echo "hello 3rd CS"
hello 3rd CS
yasmin@xubuntu-22:~$
```



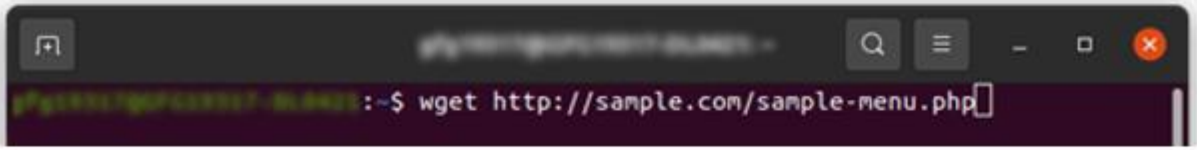
```
yasmin@xubuntu-22:~$ man -f echo
echo (1)             - display a line of text
```

19. wget command

The wget command in the Linux command line allows you to download files from the internet. It runs in the background and does not interfere with other processes.
Here is the basic syntax: **wget [option] [url]**

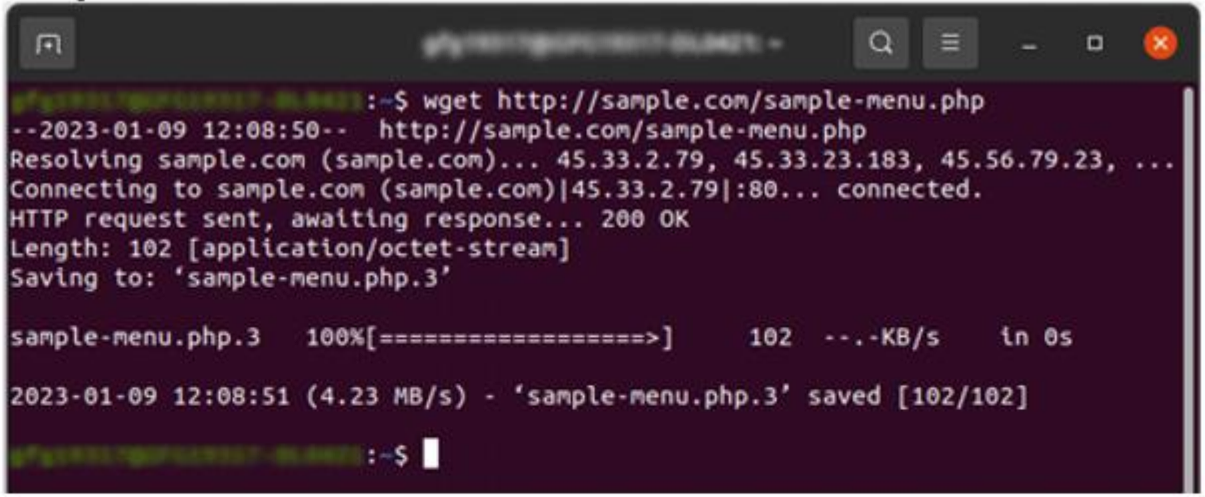
Command:

```
wget http://sample.com/sample-menu.php
```



```
[redacted]:~$ wget http://sample.com/sample-menu.php
```

Output:

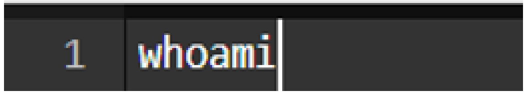


```
[redacted]:~$ wget http://sample.com/sample-menu.php
--2023-01-09 12:08:50--  http://sample.com/sample-menu.php
Resolving sample.com (sample.com)... 45.33.2.79, 45.33.23.183, 45.56.79.23, ...
Connecting to sample.com (sample.com)|45.33.2.79|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 102 [application/octet-stream]
Saving to: ‘sample-menu.php.3’

sample-menu.php.3   100%[==================>]     102  --.-KB/s    in 0s

2023-01-09 12:08:51 (4.23 MB/s) - ‘sample-menu.php.3’ saved [102/102]

[redacted]:~$ 
```

20. whoami command

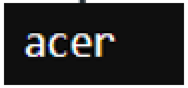
The **whoami command** provides basic information that is extremely useful when working on multiple systems. In general, if you are working with a single computer, you will not require it as frequently as a network administrator.

Command:

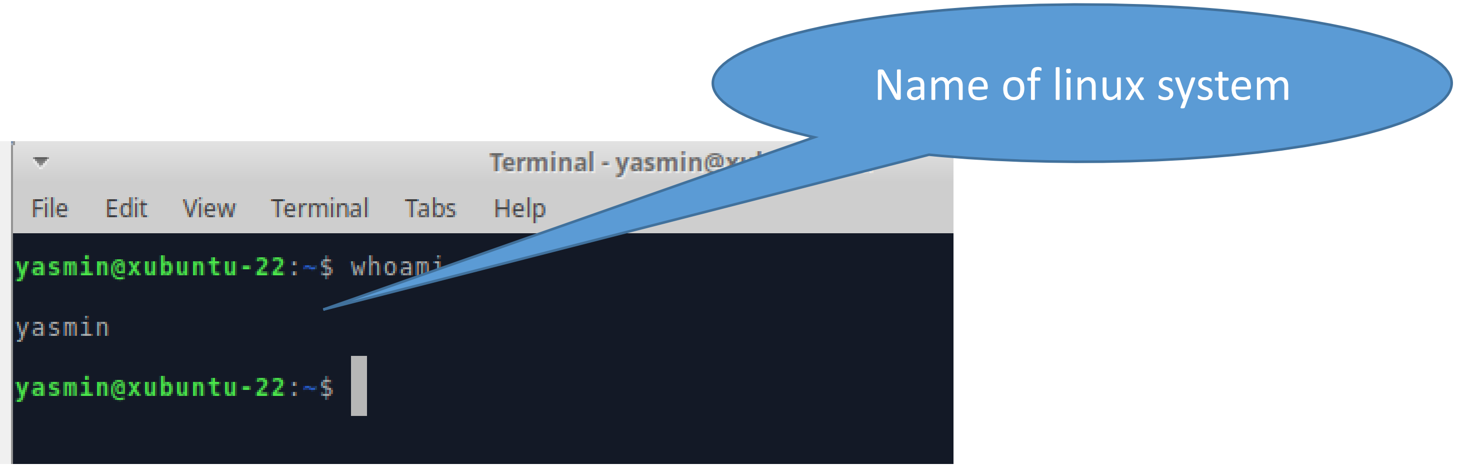


```
whoami
```

Output:



```
acer
```

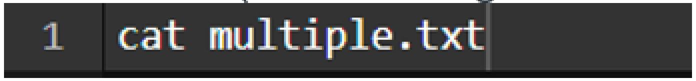


```
yasmin@xubuntu-22:~$ whoami
yasmin
yasmin@xubuntu-22:~$
```

21. sort command

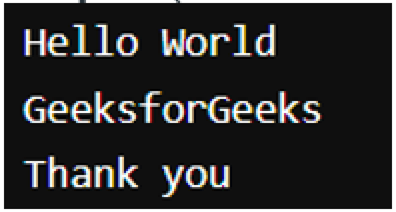
The **sort** command is used generally to sort the output of the file. Let’s use the command and see the output.

Command: (We are using the cat command to see the file content)



```
cat multiple.txt
```

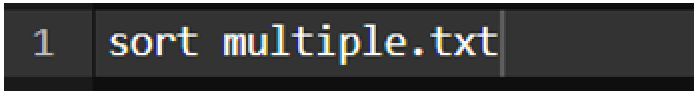
Output: (The content of multiple.txt file in the terminal)



```
Hello World
GeeksforGeeks
Thank you
```

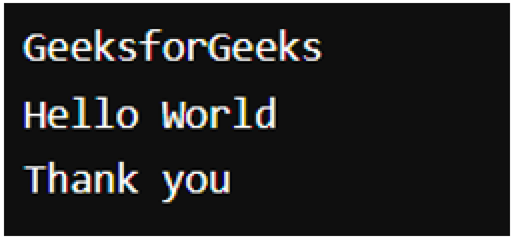
Now we will sort the outcome using the **sort command**

Command:



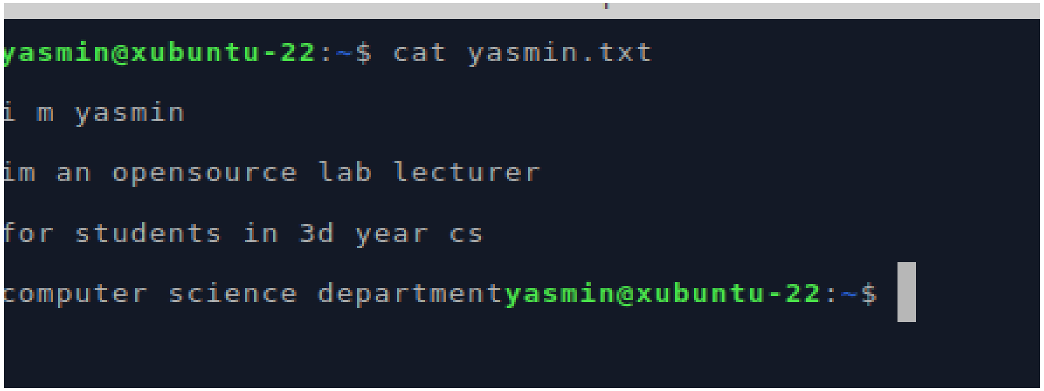
```
sort multiple.txt
```

Output:

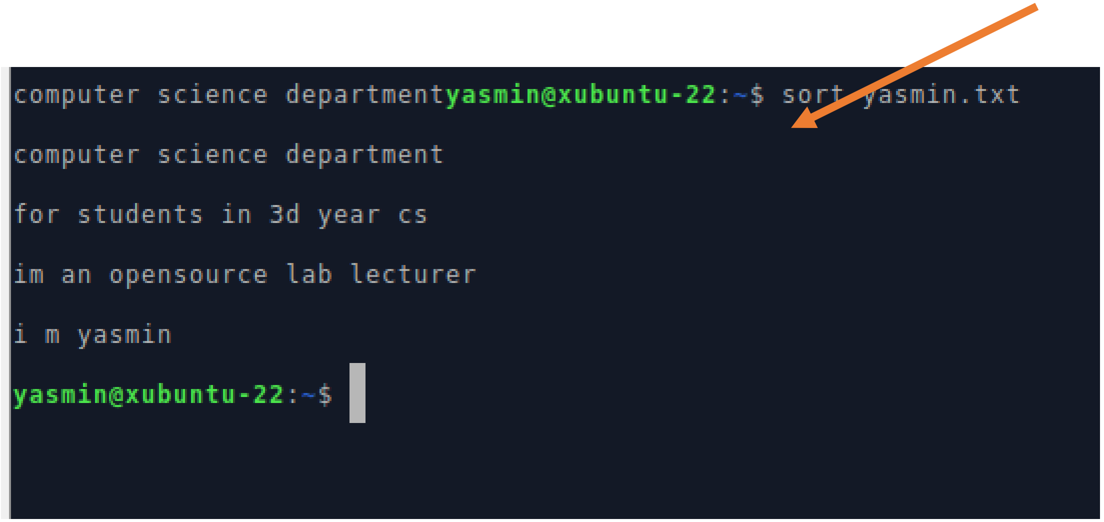


```
GeeksforGeeks
Hello World
Thank you
```

Here first we checked the file content using the **cat** command and then we sorted it alphabetically using the **sort** command.



```
yasmin@xubuntu-22:~$ cat yasmin.txt
i m yasmin
im an opensource lab lecturer
for students in 3d year cs
computer science departmentyasmin@xubuntu-22:~$
```

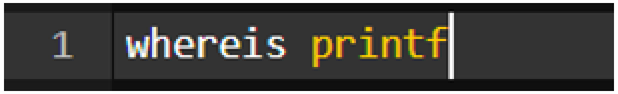


```
computer science departmentyasmin@xubuntu-22:~$ sort yasmin.txt
computer science department
for students in 3d year cs
im an opensource lab lecturer
i m yasmin
yasmin@xubuntu-22:~$
```

22. where is command

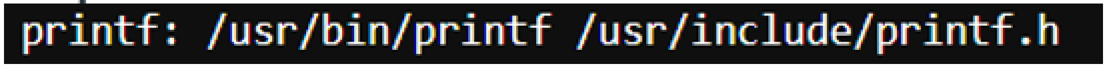
where is command in Linux is generally used to see the exact location of any command typed after this. Let's see how this performs.

Command:

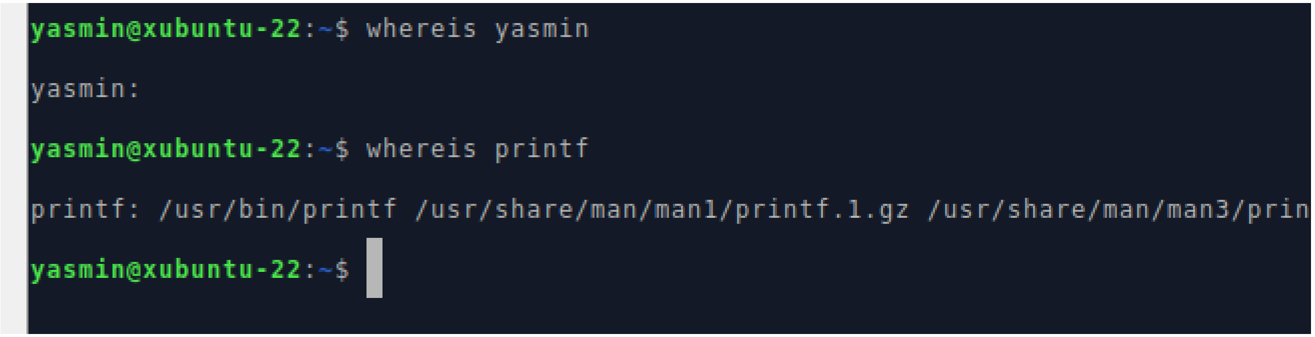


```
whereis printf
```

Output:



```
printf: /usr/bin/printf /usr/include/printf.h
```

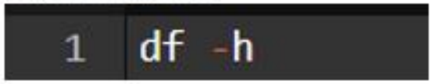


```
yasmin@xubuntu-22:~$ whereis yasmin
yasmin:
yasmin@xubuntu-22:~$ whereis printf
printf: /usr/bin/printf /usr/share/man/man1/printf.1.gz /usr/share/man/man3/prin
yasmin@xubuntu-22:~$
```

23. df is command

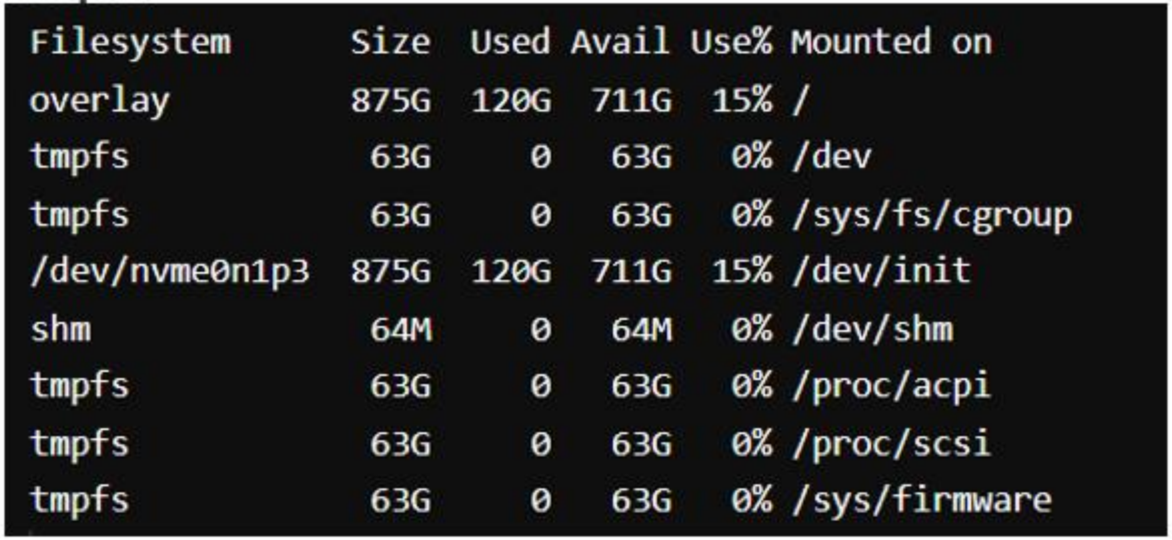
df command in Linux gets the details of the file system.

Command:

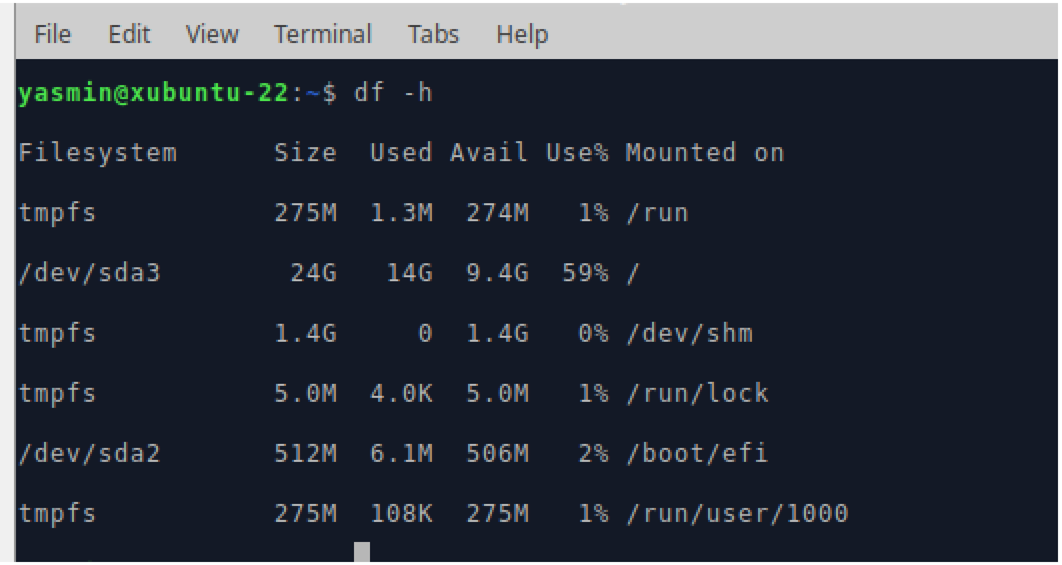


```
df -h
```

Output:



```
Filesystem      Size  Used Avail Use% Mounted on
overlay         875G  120G  711G  15% /
tmpfs            63G     0   63G   0% /dev
tmpfs            63G     0   63G   0% /sys/fs/cgroup
/dev/nvme0n1p3  875G  120G  711G  15% /dev/init
shm              64M     0   64M   0% /dev/shm
tmpfs            63G     0   63G   0% /proc/acpi
tmpfs            63G     0   63G   0% /proc/scsi
tmpfs            63G     0   63G   0% /sys/firmware
```



```
yasmin@xubuntu-22:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
tmpfs           275M  1.3M  274M   1% /run
/dev/sda3        24G   14G  9.4G  59% /
tmpfs           1.4G     0  1.4G   0% /dev/shm
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
/dev/sda2       512M  6.1M  506M   2% /boot/efi
tmpfs           275M  108K  275M   1% /run/user/1000
```

Here we have used **df -h** as simply typing **df** will return the output in bytes which is not readable, so we add **-h** to make the outputs more readable and understandable.

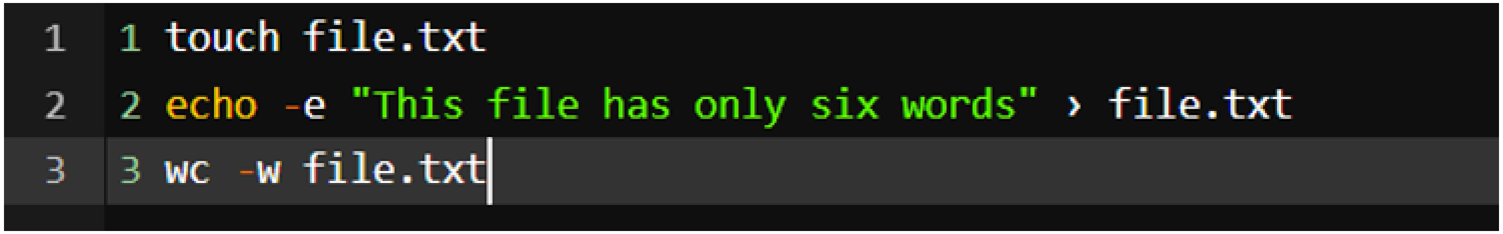
24. wc is command

wc command in Linux indicates the number of words, characters, lines, etc using a set of options.

- **wc -w** shows the number of words
- **wc -l** shows the number of lines
- **wc -m** shows the number of characters present in a file

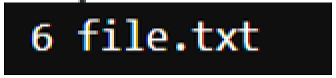
Let’s see one example of these options

Command:



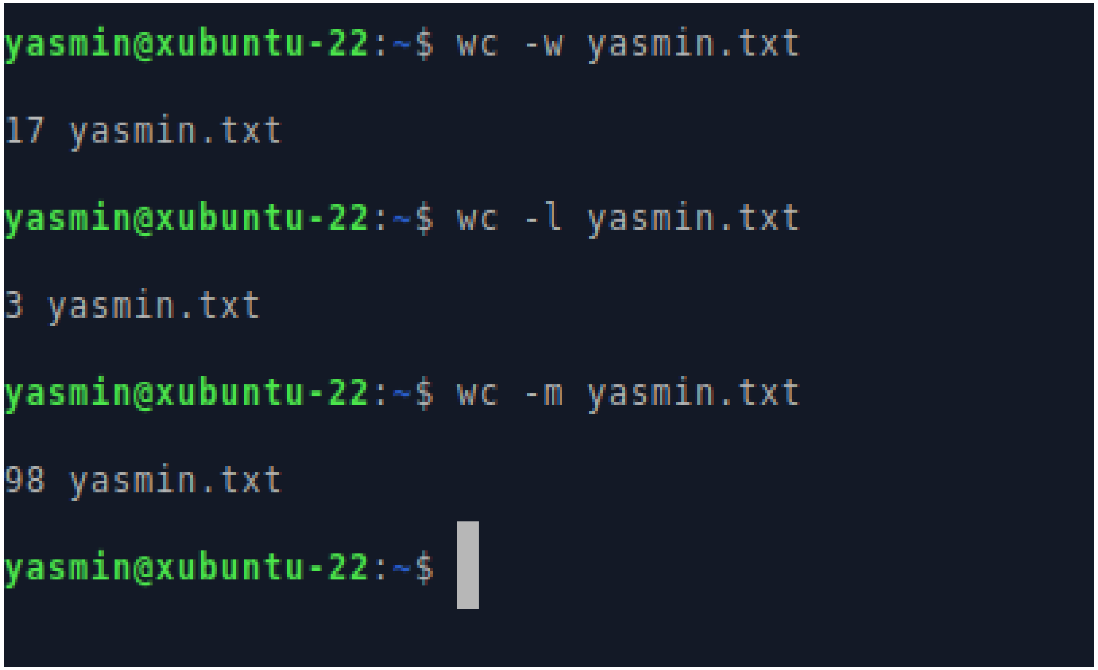
```
touch file.txt
echo -e "This file has only six words" > file.txt
wc -w file.txt
```

Output:



```
6 file.txt
```

Here we used the **touch** command to create a text file and then used the **echo** command to input a sentence that contains six words and we used the **wc -w** command to calculate the number of words in it.



```
yasmin@xubuntu-22:~$ wc -w yasmin.txt

17 yasmin.txt

yasmin@xubuntu-22:~$ wc -l yasmin.txt

3 yasmin.txt

yasmin@xubuntu-22:~$ wc -m yasmin.txt

98 yasmin.txt

yasmin@xubuntu-22:~$
```