



Bella Teaches the Linux Command Line

for version 25 August 2020

A. Bradford

Table of Contents

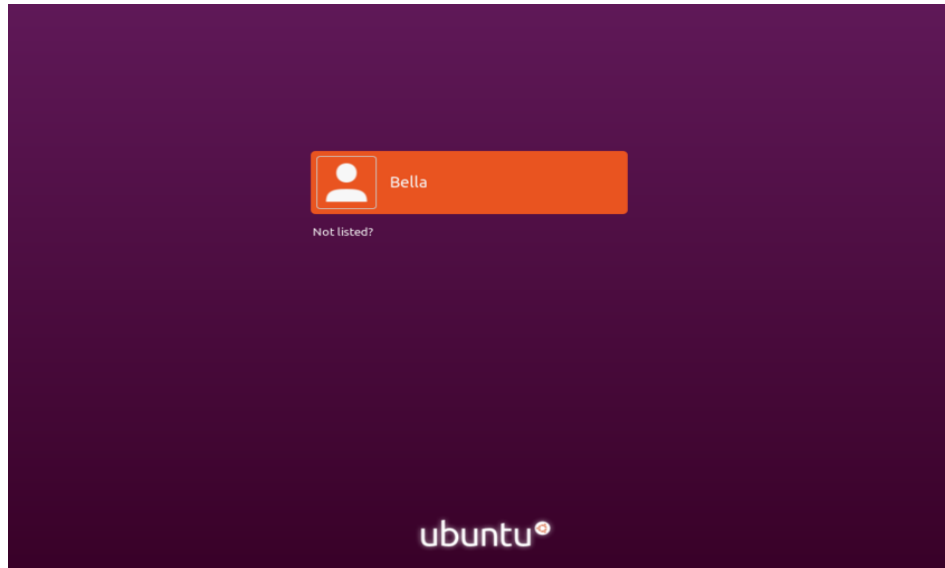
1	Login Screen, Desktop and Terminal.....	1
1.1	Login Screen.....	1
1.2	Desktop.....	2
1.3	Terminal.....	2
2	Linux Commands.....	5
2.1	whoami.....	5
2.2	pwd.....	5
2.3	ls.....	6
2.4	cd.....	7
2.5	mkdir and rmdir.....	9
2.6	echo and cat.....	10
2.7	cp and rm.....	12
2.8	gedit.....	16
2.9	more and less.....	17
2.10	date.....	18
2.11	cal.....	18
2.12	clear.....	20
2.13	man.....	21
2.14	top.....	22
3	Fun Commands.....	23
3.1	fortune.....	23
3.2	cowsay.....	23
3.3	figlet.....	23
3.4	banner.....	23
3.5	toilet.....	23
	Concept index.....	24

1 Login Screen, Desktop and Terminal

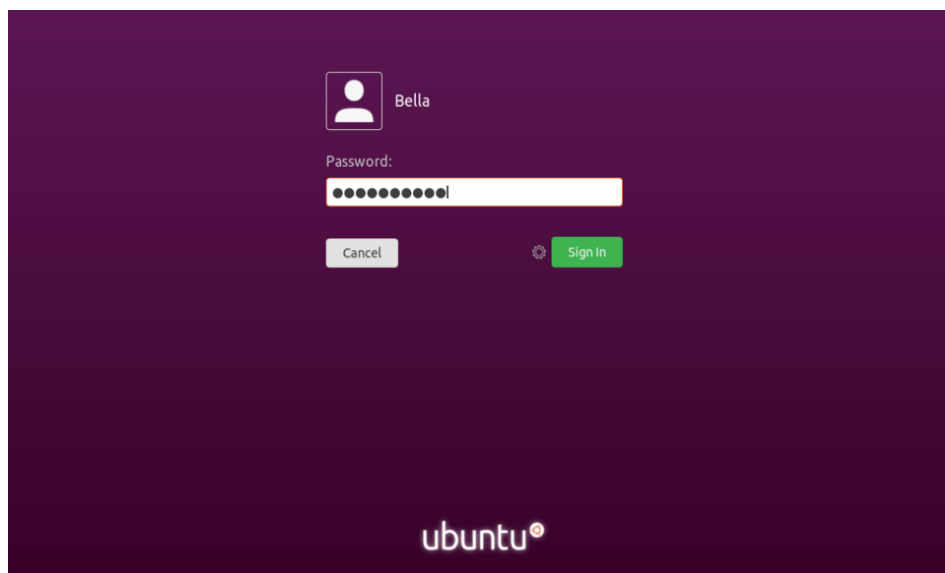
Examples are given with Ubuntu Linux. Other Linux distributions should be similar.

1.1 Login Screen

Start the Linux computer and it will proceed to the login screen.

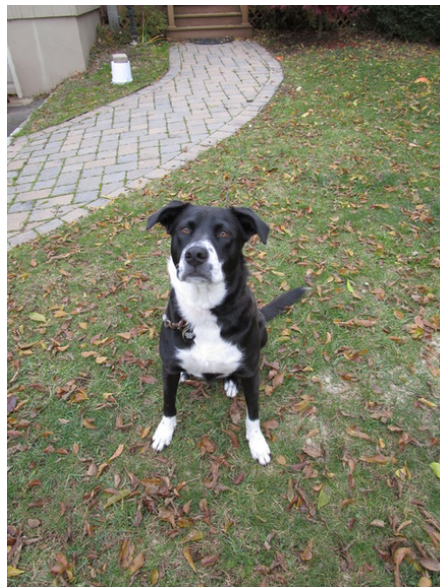
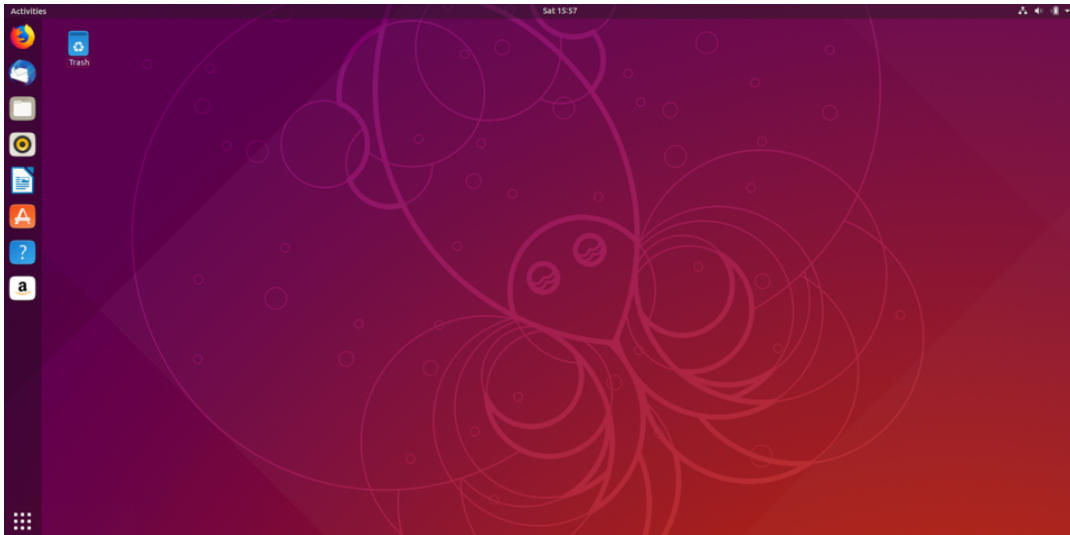


Select the user name and enter the password.



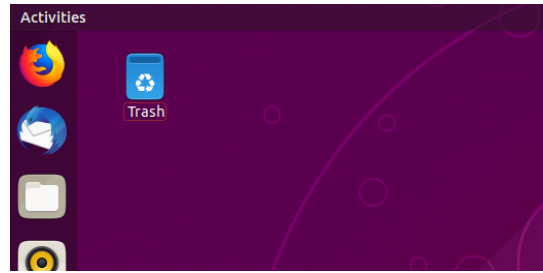
1.2 Desktop

The Linux desktop will display.

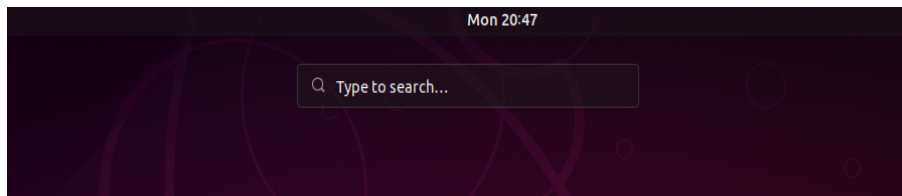


1.3 Terminal

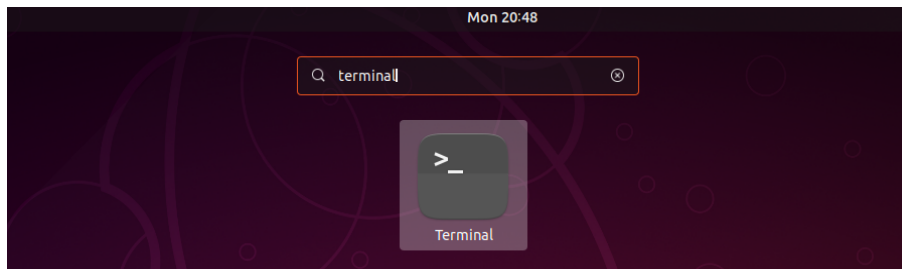
Open a Linux terminal. On the upper left hand side of the screen select Activities.



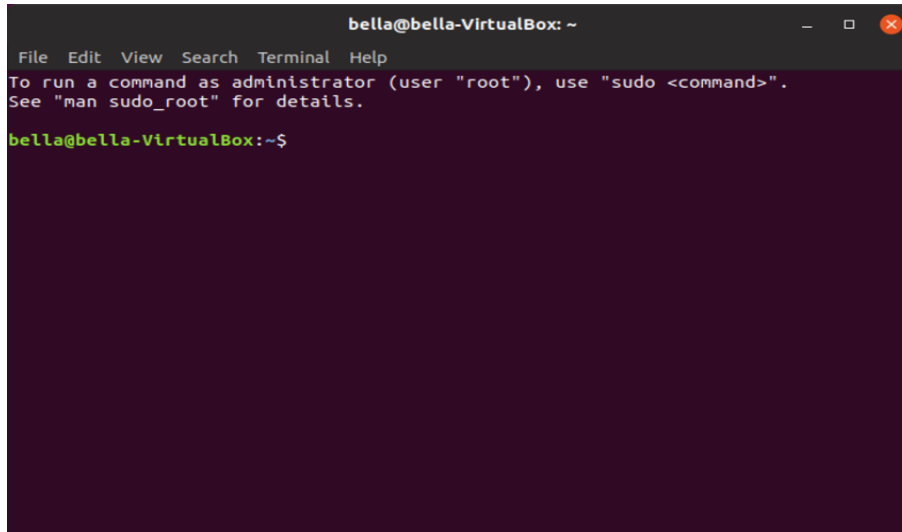
An input box saying “Type To Search...” box will appear.



Inside the “Type To Search...” input box type the word “terminal”.



The Linux terminal icon will appear. Select the terminal icon by clicking on it with the left mouse button. The terminal is also known as the Linux shell or the command line.

A screenshot of a Linux terminal window titled "bella@bella-VirtualBox: ~". The window has a dark purple background and a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". Below the menu bar, there is a message: "To run a command as administrator (user \"root\"), use \"sudo <command>\". See \"man sudo_root\" for details." The prompt "bella@bella-VirtualBox:~\$" is displayed in green text on a dark purple background.

```
bella@bella-VirtualBox: ~
File Edit View Search Terminal Help
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
bella@bella-VirtualBox:~$
```



Congratulations! You have logged into Linux and opened a terminal on the desktop! The Linux Terminal is a command language interpreter that executes commands read from your input. In the next chapter we will execute some commands.

2 Linux Commands



2.1 whoami

The command `whoami` prints your userid. At the Linux terminal type the command: `whoami`.

```
bella@bella-VirtualBox:~$ whoami
bella
bella@bella-VirtualBox:~$
```

Linux responds that “bella” is the userid. Your system should respond with your userid.

2.2 pwd

The command `pwd` is used to print the name of current working directory.

```
bella@bella-VirtualBox:~$ pwd
/home/bella
bella@bella-VirtualBox:~$
```

Bella’s current working directory is `/home/bella`. Therefore the `bella` directory is inside the `home` directory. The first slash character `/` in the `/home/bella` directory name is known as the root directory. The `home` directory is subdirectory of `root`. The `bella` directory is a subdirectory of `home`. All users of Linux have a directory in `/home`.

Linux has a hierarchical file system. The hierarchy starts with the root directory `/`. The structure of the file system looks like a tree. Inside the root directory `/` you will find files and subdirectories. The subdirectories can continue to have their own files or subdirectories. This structure can continue for many levels.

2.3 ls

The command `ls` lists directory contents. Directories can also be called folders. At the Linux terminal type the command: `ls`.

```
bella@bella-VirtualBox:~$ ls
Desktop Documents Downloads Music Pictures Public snap Templates Videos
bella@bella-VirtualBox:~$
```

We see Bella has some directories. Your system should display something similar.

We can look inside an individual directory by giving command `ls` a specific directory name to look in. Type the command: `ls Music`

```
bella@bella-VirtualBox:~$ ls
Desktop Documents Downloads Music Pictures Public snap Templates Videos
bella@bella-VirtualBox:~$ ls Music
bella@bella-VirtualBox:~$
```

Looks like Bella has no music since the `Music` directory is empty. When we type a command such as `ls Music` we say that `Music` is an argument to the command `ls`. When we type the command `ls` alone we can say `ls` has no arguments.

We can check if Bella has any documents. Type the command: `ls Documents`

```
bella@bella-VirtualBox:~$ ls
Desktop Documents Downloads Music Pictures Public snap Templates Videos
bella@bella-VirtualBox:~$ ls Music
bella@bella-VirtualBox:~$ ls Documents
bella@bella-VirtualBox:~$
```

The Document directory is empty. Bella has no documents.



The command `ls` has the ability to match filenames using a wildcard. Try typing the command: `ls D*`

The asterik `*` or star character tells the command `ls` to match any string after the letter `D`. The contents of all directories beginning with the letter `D` will be displayed.

```
bella@bella-VirtualBox:~$ ls D*
Desktop:

Documents:

Downloads:
bella@bella-VirtualBox:~$
```

The directories `Desktop`, `Documents` and `Downloads` are all empty.

Use the command `ls` to explore the Linux filesystem. Type the command: `ls /`

```
bella@bella-VirtualBox:~$ ls /
bin      etc      lib      media   root    srv      usr
boot    home    lib32    mnt     run     swapfile var
cdrom   initrd.img lib64    opt     sbin    sys     vmlinuz
dev     initrd.img.old lost+found proc    snap    tmp     vmlinuz.old
bella@bella-VirtualBox:~$
```

This lists the contents of the `/` which is known as root.

Type the command: `ls /home`

```
bella@bella-VirtualBox:~$ ls /home
abradford bella
bella@bella-VirtualBox:~$
```

Bella's home directory shows another user. Your system may display other users.

The command `ls` can also take options (also called switches). The listing of the contents of all subdirectories can be done with the command: `ls -R`

The `-R` part of command tells `ls` to recursively list all subdirectories. Try typing the command: `ls -R`

2.4 cd

The command `cd` changes your current working directory.

When the command `cd` is typed alone (with no arguments) the current working directory

is changed to your home directory.

```
bella@bella-VirtualBox:~$ cd
bella@bella-VirtualBox:~$
```

At the Linux terminal type the command: `cd Music`.

Now the current working directory has changed to the Music directory.

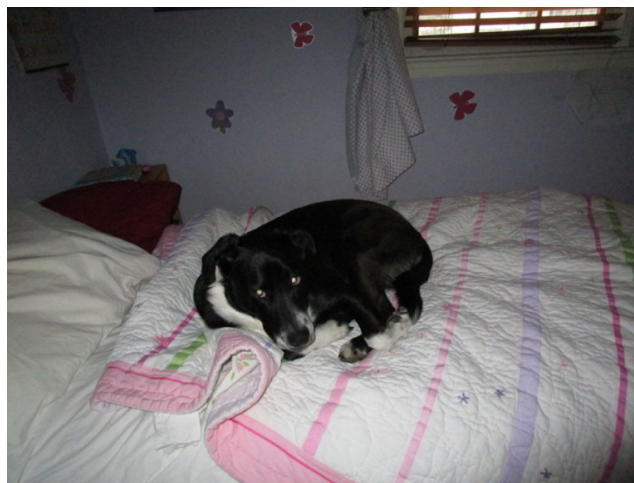
The command `ls` will display the contents of the Music directory.

```
bella@bella-VirtualBox:~$ cd Music
bella@bella-VirtualBox:~/Music$ ls
bella@bella-VirtualBox:~/Music$
```

The Music is empty.

The command `pwd` will display the full directory path.

```
bella@bella-VirtualBox:~$ cd Music
bella@bella-VirtualBox:~/Music$ ls
bella@bella-VirtualBox:~/Music$ pwd
/home/bella/Music
bella@bella-VirtualBox:~/Music$
```



To pop up from the Music directory type the command: `cd ..`

```
bella@bella-VirtualBox:~$ cd Music
bella@bella-VirtualBox:~/Music$ ls
bella@bella-VirtualBox:~/Music$ pwd
/home/bella/Music
bella@bella-VirtualBox:~/Music$ cd ..
bella@bella-VirtualBox:~$
```

Now the current working directory is one level above. To confirm this type the command: `ls`

```
bella@bella-VirtualBox:~$ cd Music
bella@bella-VirtualBox:~/Music$ ls
bella@bella-VirtualBox:~/Music$ pwd
/home/bella/Music
bella@bella-VirtualBox:~/Music$ cd ..
bella@bella-VirtualBox:~$ ls
Desktop Documents Downloads Music Pictures Public snap Templates Videos
bella@bella-VirtualBox:~$
```

Typing the command `pwd` will confirm that the current working directory is back to the user's home directory.

```
bella@bella-VirtualBox:~$ pwd
/home/bella
bella@bella-VirtualBox:~$
```

2.5 mkdir and rmdir

When using the command `ls` we can see that Bella has the following folders in her home directory.

```
bella@bella-VirtualBox:~$ ls
Desktop Documents Downloads Music Pictures Public snap Templates Videos
bella@bella-VirtualBox:~$
```

To create a directory named `Test` type the command: `mkdir Test`

```
bella@bella-VirtualBox:~$ ls
Desktop Documents Downloads Music Pictures Public snap Templates Videos
bella@bella-VirtualBox:~$ mkdir Test
bella@bella-VirtualBox:~$
```

Try typing the command: `ls` to confirm this directory was created. You should see the directory `Test` listed with the other directories.

```
bella@bella-VirtualBox:~$ ls
Desktop Documents Downloads Music Pictures Public snap Templates Videos
bella@bella-VirtualBox:~$ mkdir Test
bella@bella-VirtualBox:~$ ls
Desktop Downloads Pictures snap Test
Documents Music Public Templates Videos
bella@bella-VirtualBox:~$
```

To remove the directory named Test type the command: `rmdir Test`

```
bella@bella-VirtualBox:~$ ls
Desktop Documents Downloads Music Pictures Public snap Templates Videos
bella@bella-VirtualBox:~$ mkdir Test
bella@bella-VirtualBox:~$ ls
Desktop Downloads Pictures snap Test
Documents Music Public Templates Videos
bella@bella-VirtualBox:~$ rmdir Test
bella@bella-VirtualBox:~$
```

Type the command `ls` to confirm that the directory Test has been removed.

```
bella@bella-VirtualBox:~$ ls
Desktop Documents Downloads Music Pictures Public snap Templates Videos
bella@bella-VirtualBox:~$ mkdir Test
bella@bella-VirtualBox:~$ ls
Desktop Downloads Pictures snap Test
Documents Music Public Templates Videos
bella@bella-VirtualBox:~$ rmdir Test
bella@bella-VirtualBox:~$ ls
Desktop Documents Downloads Music Pictures Public snap Templates Videos
bella@bella-VirtualBox:~$
```



2.6 echo and cat

The command `echo` is used to write text to the screen. The command `echo` can also write text to a file. At the Linux terminal type the command: `echo`.

```
bella@bella-VirtualBox:~$ echo
bella@bella-VirtualBox:~$ █
```

The command `echo` with no arguments will just display a blank line. Now try typing the command: `echo Hello`

```
bella@bella-VirtualBox:~$ echo
bella@bella-VirtualBox:~$ echo Hello
Hello
bella@bella-VirtualBox:~$ █
```

The text `Hello` will be displayed to the terminal. The command `echo` can also push text to a file.

Try the command: `echo "This is a test" > file.txt`

The greater-than sign `>` tells the command `echo` to redirect the output to the file. The output will *not* go to the display.

```
bella@bella-VirtualBox:~$ echo
bella@bella-VirtualBox:~$ echo Hello
Hello
bella@bella-VirtualBox:~$ echo "This is a test" > file.txt
bella@bella-VirtualBox:~$ █
```

The command `echo` has placed the text “This is a test” inside the `file.txt`.

We can confirm the file exists with the command `ls`

```
bella@bella-VirtualBox:~$ echo
bella@bella-VirtualBox:~$ echo Hello
Hello
bella@bella-VirtualBox:~$ echo "This is a test" > file.txt
bella@bella-VirtualBox:~$ ls
Desktop  Downloads  Music      Public  Templates
Documents file.txt   Pictures  snap    Videos
bella@bella-VirtualBox:~$ █
```

Notice the new file `file.txt`. The contents of the file `file.txt` can be displayed with the command: `cat file.txt`. The command `cat` can concatenate files and print files on the display. The contents of the file is the text “This is a test”.

```
bella@bella-VirtualBox:~$ echo
bella@bella-VirtualBox:~$ echo Hello
Hello
bella@bella-VirtualBox:~$ echo "This is a test" > file.txt
bella@bella-VirtualBox:~$ ls
Desktop  Downloads  Music      Public  Templates
Documents file.txt   Pictures  snap    Videos
bella@bella-VirtualBox:~$ cat file.txt
This is a test
bella@bella-VirtualBox:~$
```



2.7 cp and rm

The command `cp` copies files and directories. The command `rm` removes files or directories. Create a file with the command: `echo "This is another test file" > TestFile.txt`

```
bella@bella-VirtualBox:~$ echo "This is another test file" > TestFile.txt
bella@bella-VirtualBox:~$
```

Confirm the file exists with the command `ls` and confirm the file contents with the command `cat`.

```
bella@bella-VirtualBox:~$ echo "This is another test file" > TestFile.txt
bella@bella-VirtualBox:~$ ls
Desktop  Downloads  Pictures  snap    TestFile.txt
Documents Music      Public   Templates Videos
bella@bella-VirtualBox:~$ cat TestFile.txt
This is another test file
bella@bella-VirtualBox:~$
```

Using the command `rm` delete the file with: `rm TestFile.txt`

```
bella@bella-VirtualBox:~$ echo "This is another test file" > TestFile.txt
bella@bella-VirtualBox:~$ ls
Desktop  Downloads  Pictures  snap      TestFile.txt
Documents Music      Public   Templates Videos
bella@bella-VirtualBox:~$ cat TestFile.txt
This is another test file
bella@bella-VirtualBox:~$ rm TestFile.txt
bella@bella-VirtualBox:~$
```

Now confirm the file was deleted with the command `ls`

```
bella@bella-VirtualBox:~$ echo "This is another test file" > TestFile.txt
bella@bella-VirtualBox:~$ ls
Desktop  Downloads  Pictures  snap      TestFile.txt
Documents Music      Public   Templates Videos
bella@bella-VirtualBox:~$ cat TestFile.txt
This is another test file
bella@bella-VirtualBox:~$ rm TestFile.txt
bella@bella-VirtualBox:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  snap  Templates  Videos
bella@bella-VirtualBox:~$
```



Create another file `NewTestFile.txt` with the command `echo`.

Type the command: `echo "Test file to be copied" > NewTestFile.txt`

Confirm the file exists with the command `ls`. Confirm the contents with the command `cat`.

```
bella@bella-VirtualBox:~$ echo "Test file to be copied" > NewTestFile.txt
bella@bella-VirtualBox:~$ ls
Desktop  Downloads  NewTestFile.txt  Public  Templates
Documents Music      Pictures         snap    Videos
bella@bella-VirtualBox:~$ cat NewTestFile.txt
Test file to be copied
```


Copy the file `NewTestFile.txt` to the file `1.txt` with the command: `cp NewTestFile.txt 1.txt`

```
bella@bella-VirtualBox:~$ echo "Test file to be copied" > NewTestFile.txt
bella@bella-VirtualBox:~$ ls
Desktop  Downloads  NewTestFile.txt  Public  Templates
Documents Music      Pictures          snap    Videos
bella@bella-VirtualBox:~$ cat NewTestFile.txt
Test file to be copied
bella@bella-VirtualBox:~$ cp NewTestFile.txt 1.txt
```

Copy the file `NewTestFile.txt` to the file `2.txt` with the command: `cp NewTestFile.txt 2.txt`

```
bella@bella-VirtualBox:~$ echo "Test file to be copied" > NewTestFile.txt
bella@bella-VirtualBox:~$ ls
Desktop  Downloads  NewTestFile.txt  Public  Templates
Documents Music      Pictures          snap    Videos
bella@bella-VirtualBox:~$ cat NewTestFile.txt
Test file to be copied
bella@bella-VirtualBox:~$ cp NewTestFile.txt 1.txt
bella@bella-VirtualBox:~$ cp NewTestFile.txt 2.txt
```

Copy the file `NewTestFile.txt` to the file `3.txt` with the command: `cp NewTestFile.txt 3.txt`

```
bella@bella-VirtualBox:~$ echo "Test file to be copied" > NewTestFile.txt
bella@bella-VirtualBox:~$ ls
Desktop  Downloads  NewTestFile.txt  Public  Templates
Documents Music      Pictures          snap    Videos
bella@bella-VirtualBox:~$ cat NewTestFile.txt
Test file to be copied
bella@bella-VirtualBox:~$ cp NewTestFile.txt 1.txt
bella@bella-VirtualBox:~$ cp NewTestFile.txt 2.txt
bella@bella-VirtualBox:~$ cp NewTestFile.txt 3.txt
```

Confirm the existence of the new files `1.txt`, `2.txt` and `3.txt` with the command: `ls`

```
bella@bella-VirtualBox:~$ echo "Test file to be copied" > NewTestFile.txt
bella@bella-VirtualBox:~$ ls
Desktop  Downloads  NewTestFile.txt  Public  Templates
Documents Music      Pictures          snap    Videos
bella@bella-VirtualBox:~$ cat NewTestFile.txt
Test file to be copied
bella@bella-VirtualBox:~$ cp NewTestFile.txt 1.txt
bella@bella-VirtualBox:~$ cp NewTestFile.txt 2.txt
bella@bella-VirtualBox:~$ cp NewTestFile.txt 3.txt
bella@bella-VirtualBox:~$ ls
1.txt 3.txt Documents Music Pictures snap Videos
2.txt Desktop Downloads NewTestFile.txt Public Templates
bella@bella-VirtualBox:~$
```

Concatenate the files `1.txt`, `2.txt` and `3.txt` to the screen display with the command:

```
cat 1.txt 2.txt 3.txt
```

```

bella@bella-VirtualBox:~$ echo "Test file to be copied" > NewTestFile.txt
bella@bella-VirtualBox:~$ ls
Desktop  Downloads  NewTestFile.txt  Public  Templates
Documents Music  Pictures  snap  Videos
bella@bella-VirtualBox:~$ cat NewTestFile.txt
Test file to be copied
bella@bella-VirtualBox:~$ cp NewTestFile.txt 1.txt
bella@bella-VirtualBox:~$ cp NewTestFile.txt 2.txt
bella@bella-VirtualBox:~$ cp NewTestFile.txt 3.txt
bella@bella-VirtualBox:~$ ls
1.txt 3.txt  Documents  Music  Pictures  snap  Videos
2.txt Desktop Downloads  NewTestFile.txt  Public  Templates
bella@bella-VirtualBox:~$ cat 1.txt 2.txt 3.txt
Test file to be copied
Test file to be copied
Test file to be copied
bella@bella-VirtualBox:~$ █

```

All three files are displayed on the screen back to back. All three files have the same contents. Remove one of the files with the command: `rm 1.txt`

```

bella@bella-VirtualBox:~$ echo "Test file to be copied" > NewTestFile.txt
bella@bella-VirtualBox:~$ ls
Desktop  Downloads  NewTestFile.txt  Public  Templates
Documents Music  Pictures  snap  Videos
bella@bella-VirtualBox:~$ cat NewTestFile.txt
Test file to be copied
bella@bella-VirtualBox:~$ cp NewTestFile.txt 1.txt
bella@bella-VirtualBox:~$ cp NewTestFile.txt 2.txt
bella@bella-VirtualBox:~$ cp NewTestFile.txt 3.txt
bella@bella-VirtualBox:~$ ls
1.txt 3.txt  Documents  Music  Pictures  snap  Videos
2.txt Desktop Downloads  NewTestFile.txt  Public  Templates
bella@bella-VirtualBox:~$ cat 1.txt 2.txt 3.txt
Test file to be copied
Test file to be copied
Test file to be copied
bella@bella-VirtualBox:~$ rm 1.txt
bella@bella-VirtualBox:~$ █

```

Using what is known as a wildcard have command `ls` check for files only ending with `.txt` file extension. Try the command: `ls *.txt`

```

bella@bella-VirtualBox:~$ ls *.txt
2.txt 3.txt NewTestFile.txt
bella@bella-VirtualBox:~$ █

```

Only files with the `.txt` file extension have been displayed. The file `1.txt` has been removed. Remove the files `2.txt` and `3.txt` with the command `rm 2.txt 3.txt`

```

bella@bella-VirtualBox:~$ rm 2.txt 3.txt
bella@bella-VirtualBox:~$ █

```

Confirm the removal of `2.txt` and `3.txt` by wildcard listing `.txt` file extension with command: `ls *.txt`

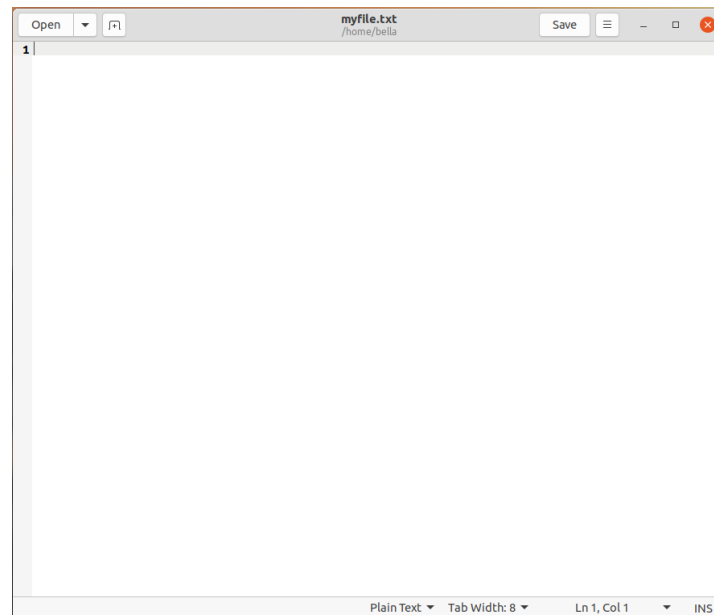
```
bella@bella-VirtualBox:~$ ls *.txt
NewTestFile.txt
bella@bella-VirtualBox:~$
```

2.8 gedit

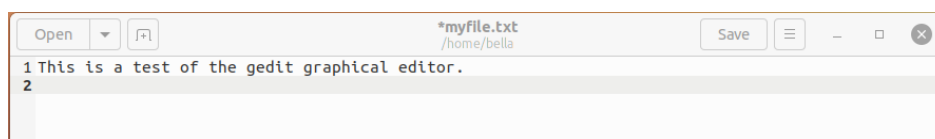
The command `gedit` is a graphical text editor. `gedit` can be used to create or edit files. `gedit` can be started from the command line. Type the command: `gedit myfile.txt` to create and edit the file `myfile.txt`

```
bella@bella-VirtualBox:~$ gedit
```

The `gedit` graphical text editor will display. Feel free to enter any text you want.



In this example I have entered the text “This is a test of the gedit graphical editor” and pressed newline. Press the “Save” button to save the file `myfile.txt`



Type the command: `ls` to see that there is a new file named `myfile.txt`. Also try the command: `ls *.txt` to list all files ending with `.txt`. Both commands will show that the new file `myfile.txt` exists.

```
bella@bella-VirtualBox:~$ ls
Desktop  Downloads  myfile.txt  Public  Templates
Documents Music      Pictures    snap    Videos
bella@bella-VirtualBox:~$ ls *.txt
myfile.txt
bella@bella-VirtualBox:~$
```

Feel free to use the command `gedit myfile.txt` to add more text or change the file `myfile.txt`.



2.9 more and less



2.10 date

The command `date` will print the system date and time.

```
bella@bella-VirtualBox:~$ date
Tue 25 Aug 2020 07:40:57 AM EDT
bella@bella-VirtualBox:~$
```



2.11 cal

The command `cal` will display a calendar. With no arguments to the command, the current month is displayed. Today's date will be highlighted.

```

bella@bella-VirtualBox:~$ cal
  August 2020
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
bella@bella-VirtualBox:~$ █

```

The command `cal` can display all months for a year. The year to display can be given as an argument to the command `cal`. Try typing the command: `cal 2020`

```

bella@bella-VirtualBox:~$ cal 2020
 2020
  January          February          March
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
                1  2  3  4                1  1  2  3  4  5  6  7
 5  6  7  8  9 10 11  2  3  4  5  6  7  8  8  9 10 11 12 13 14
12 13 14 15 16 17 18  9 10 11 12 13 14 15 15 16 17 18 19 20 21
19 20 21 22 23 24 25 16 17 18 19 20 21 22 22 23 24 25 26 27 28
26 27 28 29 30 31  23 24 25 26 27 28 29 29 30 31

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

  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  3  4                1  1  2  3  4  5
 5  6  7  8  9 10 11  2  3  4  5  6  7  8  6  7  8  9 10 11 12
12 13 14 15 16 17 18  9 10 11 12 13 14 15 13 14 15 16 17 18 19
19 20 21 22 23 24 25 16 17 18 19 20 21 22 20 21 22 23 24 25 26
26 27 28 29 30 31  23 24 25 26 27 28 29 27 28 29 30
                30 31

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

bella@bella-VirtualBox:~$ █

```

The command `cal` can display calendars in the past. Try typing the command `cal` and giving the your birthday year as the first argument. If your birthday year was 2009 you would type: `cal 2009`



2.12 clear

To clear the terminal screen type the command: `clear`

```
bella@bella-VirtualBox:~$
```

A terminal window with a dark purple background. The prompt 'bella@bella-VirtualBox:~\$' is visible at the top left. The rest of the screen is empty, indicating that the 'clear' command has been executed successfully.



2.13 man

The command `man` provides a reference manual to commands. All commands such as `whoami`, `pwd`, `ls`, `cd`, `mkdir`, `rmdir`, `echo`, `rm`, `rm`, `more`, `less`, `date`, `cal` and `clear` have reference manuals.

The command reference manuals give information on command usage and options. Command switches (options) and arguments are described. Try command `man ls`

```
LS(1)                                User Commands                                LS(1)
NAME
  ls - list directory contents

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

DESCRIPTION
  List information about the FILES (the current directory by default).
  Sort entries 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 .

  -A, --almost-all
        do not list implied . and ..

  --author
  Manual page ls(1) line 1 (press h for help or q to quit)
```

Explore more `man` pages. The command `man` even has a reference manual page. Try the command: `man man`.



2.14 top



3 Fun Commands

3.1 fortune

3.2 cowsay

3.3 figlet

3.4 banner

3.5 toilet

Concept index

B

banner 23

C

cal 18

cat 10

cd 7

clear 20

cowsay 23

cp 12

D

date 18

Desktop 2

E

echo 10

F

figlet 23

fortune 23

G

gedit 16

L

less 17

Linux Commands 5

Login Screen 1

ls 6

M

man 21

mkdir 9

more 17

P

pwd 5

R

rm 12

rmdir 9

T

Terminal 2

toilet 23

top 22

W

whoami 5