Introduction to Programming

Week – *1*, Lecture – *2*

Introduction to Linux - II

SAURABH SRIVASTAVA

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

IIT KANPUR

Revisiting Linux Kernel

Remember that the Linux Kernel is the core set of programs that talk to your hardware

Revisiting Linux Kernel

Remember that the Linux Kernel is the core set of programs that talk to your hardware

Applications and Programs do not have access to the hardware directly

- But, if Applications are not allowed to interact with the hardware, how can they work?
- Imagine an addition program – that cannot take its operands from the user !!

Revisiting Linux Kernel

Remember that the Linux Kernel is the core set of programs that talk to your hardware

Applications and Programs do not have access to the hardware directly

- But, if Applications are not allowed to interact with the hardware, how can they work?
- Imagine an addition program – that cannot take its operands from the user !!

The Kernel provides its services to the Applications in the form of *System Calls*

- A System Call is a request by an Application to the Kernel, to perform a task on its behalf
- These tasks may include creating a file or reading a character from the keyboard
- The Kernel performs these tasks on behalf of the Application, and returns any results, if applicable

Revisiting Linux Kernel

Remember that the Linux Kernel is the core set of programs that talk to your hardware

Applications and Programs do not have access to the hardware directly

- But, if Applications are not allowed to interact with the hardware, how can they work?
- Imagine an addition program – that cannot take its operands from the user !!

The Kernel provides its services to the Applications in the form of *System Calls*

- A System Call is a request by an Application to the Kernel, to perform a task on its behalf
- These tasks may include creating a file or reading a character from the keyboard
- The Kernel performs these tasks on behalf of the Application, and returns any results, if applicable

As an analogy, assume that the Computer is a Restaurant and an Application is a Customer

- Then, the Kernel is the employee that serves the Customer and the Kitchen is the hardware
- The customers are not allowed to visit the Kitchen, they simply tell the server, what they want !!
- The server brings the food out for them (and collects the payment and tips from the Customer :-D)

System Calls

In our Restaurant analogy, System Calls are the *orders that the Customers* place

- The server offers the Customer a “fixed menu”
- The Customer must pick something from the menu – nothing out of it will be accepted by the server

System Calls

In our Restaurant analogy, System Calls are the *orders that the Customers* place

- The server offers the Customer a “fixed menu”
- The Customer must pick something from the menu – nothing out of it will be accepted by the server

System Calls

In our Restaurant analogy, System Calls are the *orders that the Customers* place

- The server offers the Customer a “fixed menu”
- The Customer must pick something from the menu – nothing out of it will be accepted by the server
- Even for those items available in the menu, the actual serving need not be sequential
  - How many times have you seen people who walked in after you in a Restaurant, get their food before you :-P?

System Calls

In our Restaurant analogy, System Calls are the *orders that the Customers* place

- The server offers the Customer a “fixed menu”
- The Customer must pick something from the menu – nothing out of it will be accepted by the server
- Even for those items available in the menu, the actual serving need not be sequential
  - How many times have you seen people who walked in after you in a Restaurant, get their food before you :-P?

The Kernel behaves in a similar fashion

- Applications request for a particular operation, by making a System Call (out of a fixed set of System Calls)

System Calls

In our Restaurant analogy, System Calls are the *orders that the Customers* place

- The server offers the Customer a “fixed menu”
- The Customer must pick something from the menu – nothing out of it will be accepted by the server
- Even for those items available in the menu, the actual serving need not be sequential
  - How many times have you seen people who walked in after you in a Restaurant, get their food before you :-P?

The Kernel behaves in a similar fashion

- Applications request for a particular operation, by making a System Call (out of a fixed set of System Calls)
- The System Call contains the details of the required service, and any inputs necessary to fulfil it
- … then, the Application may wait for the call to finish, or, may do other stuff while Kernel does its job

System Calls

In our Restaurant analogy, System Calls are the *orders that the Customers* place

- The server offers the Customer a “fixed menu”
- The Customer must pick something from the menu – nothing out of it will be accepted by the server
- Even for those items available in the menu, the actual serving need not be sequential
  - How many times have you seen people who walked in after you in a Restaurant, get their food before you :-P?

The Kernel behaves in a similar fashion

- Applications request for a particular operation, by making a System Call (out of a fixed set of System Calls)
- The System Call contains the details of the required service, and any inputs necessary to fulfil it
- … then, the Application may wait for the call to finish, or, may do other stuff while Kernel does its job
- The Kernel may or may not serve the System Calls in the order it receives them
  - There are many aspects which determine the order – you’ll read more about it in your OS course

System Calls

In our Restaurant analogy, System Calls are the *orders that the Customers* place

- The server offers the Customer a “fixed menu”
- The Customer must pick something from the menu – nothing out of it will be accepted by the server
- Even for those items available in the menu, the actual serving need not be sequential
  - How many times have you seen people who walked in after you in a Restaurant, get their food before you :-P?

The Kernel behaves in a similar fashion

- Applications request for a particular operation, by making a System Call (out of a fixed set of System Calls)
- The System Call contains the details of the required service, and any inputs necessary to fulfil it
- … then, the Application may wait for the call to finish, or, may do other stuff while Kernel does its job
- The Kernel may or may not serve the System Calls in the order it receives them
  - There are many aspects which determine the order – you’ll read more about it in your OS course
- Typically, System Calls are made to take input, produce output, open a file, start a network connection etc.

Command-line interface to System Calls

Assume you have to perform a simple task

- Create a file in your Home Directory
- Write some text in it
- Save the contents

Command-line interface to System Calls

Assume you have to perform a simple task

- Create a file in your Home Directory
- Write some text in it
- Save the contents

There are three System Calls that are required to be made for this

Command-line interface to System Calls

Assume you have to perform a simple task

- Create a file in your Home Directory
- Write some text in it
- Save the contents

There are three System Calls that are required to be made for this

- A call to create a file on the disk and “open” the file to accept data
  - This involves creating an entry for the file at a designated place on the disk and opening a “virtual channel”

Command-line interface to System Calls

Assume you have to perform a simple task

- Create a file in your Home Directory
- Write some text in it
- Save the contents

There are three System Calls that are required to be made for this

- A call to create a file on the disk and “open” the file to accept data
  - This involves creating an entry for the file at a designated place on the disk and opening a “virtual channel”
- Writing some data at another designated place on the disk
  - This too, involves a disk write operation, via the “virtual channel”

Command-line interface to System Calls

Assume you have to perform a simple task

- Create a file in your Home Directory
- Write some text in it
- Save the contents

There are three System Calls that are required to be made for this

- A call to create a file on the disk and “open” the file to accept data
  - This involves creating an entry for the file at a designated place on the disk and opening a “virtual channel”
- Writing some data at another designated place on the disk
  - This too, involves a disk write operation, via the “virtual channel”
- “Close” the file, so that the “virtual channel” is no longer valid

Command-line interface to System Calls

Assume you have to perform a simple task

- Create a file in your Home Directory
- Write some text in it
- Save the contents

There are three System Calls that are required to be made for this

- A call to create a file on the disk and “open” the file to accept data
  - This involves creating an entry for the file at a designated place on the disk and opening a “virtual channel”
- Writing some data at another designated place on the disk
  - This too, involves a disk write operation, via the “virtual channel”
- “Close” the file, so that the “virtual channel” is no longer valid

While it seems a complex task, you can achieve so using the `cat` application in Linux

Command-line interface to System Calls

Assume you have to perform a simple task

- Create a file in your Home Directory
- Write some text in it
- Save the contents

There are three System Calls that are required to be made for this

- A call to create a file on the disk and “open” the file to accept data
  - This involves creating an entry for the file at a designated place on the disk and opening a “virtual channel”
- Writing some data at another designated place on the disk
  - This too, involves a disk write operation, via the “virtual channel”
- “Close” the file, so that the “virtual channel” is no longer valid

While it seems a complex task, you can achieve so using the `cat` application in Linux

I will show the examples in this lecture using *bash* over *Lubuntu*, but is similar for other distributions

The first bash experience

On your Linux installation, open the “Terminal” application

The first bash experience

On your Linux installation, open the “Terminal” application

The Terminal application is more like a “meta-application”

- You can use it to access other Applications installed on your Linux installation

The first bash experience

On your Linux installation, open the “Terminal” application

The Terminal application is more like a “meta-application”

- You can use it to access other Applications installed on your Linux installation

You can open it either from the Menu or by using some keyboard short-cut

The first bash experience

On your Linux installation, open the “Terminal” application

The Terminal application is more like a “meta-application”

- You can use it to access other Applications installed on your Linux installation

You can open it either from the Menu or by using some keyboard short-cut

- On distributions from the Ubuntu family, the short-cut is `Ctrl + Alt + T`
  - The + here means that you keep the key on the left side pressed, and then press the key on its right

The first bash experience

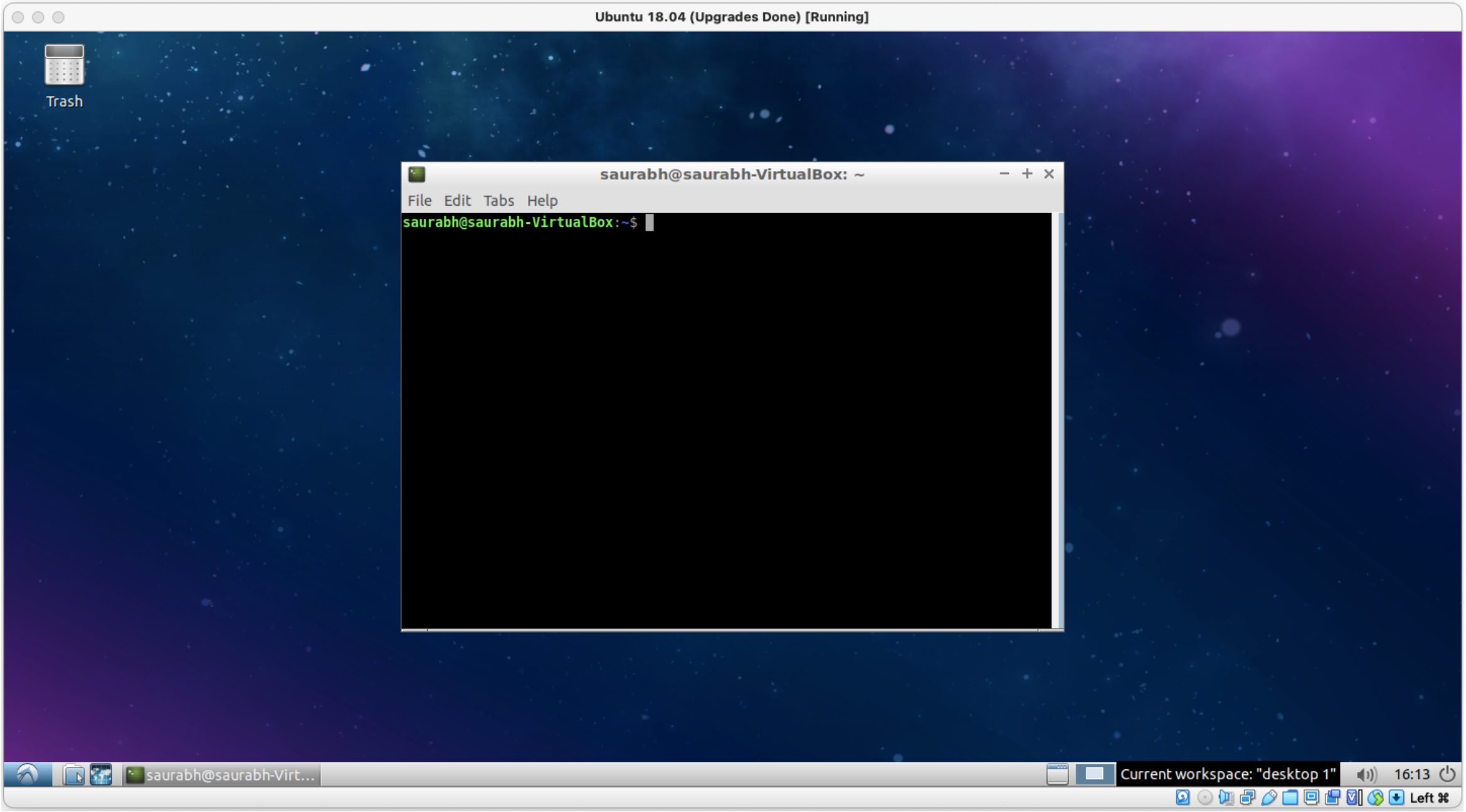
On your Linux installation, open the “Terminal” application

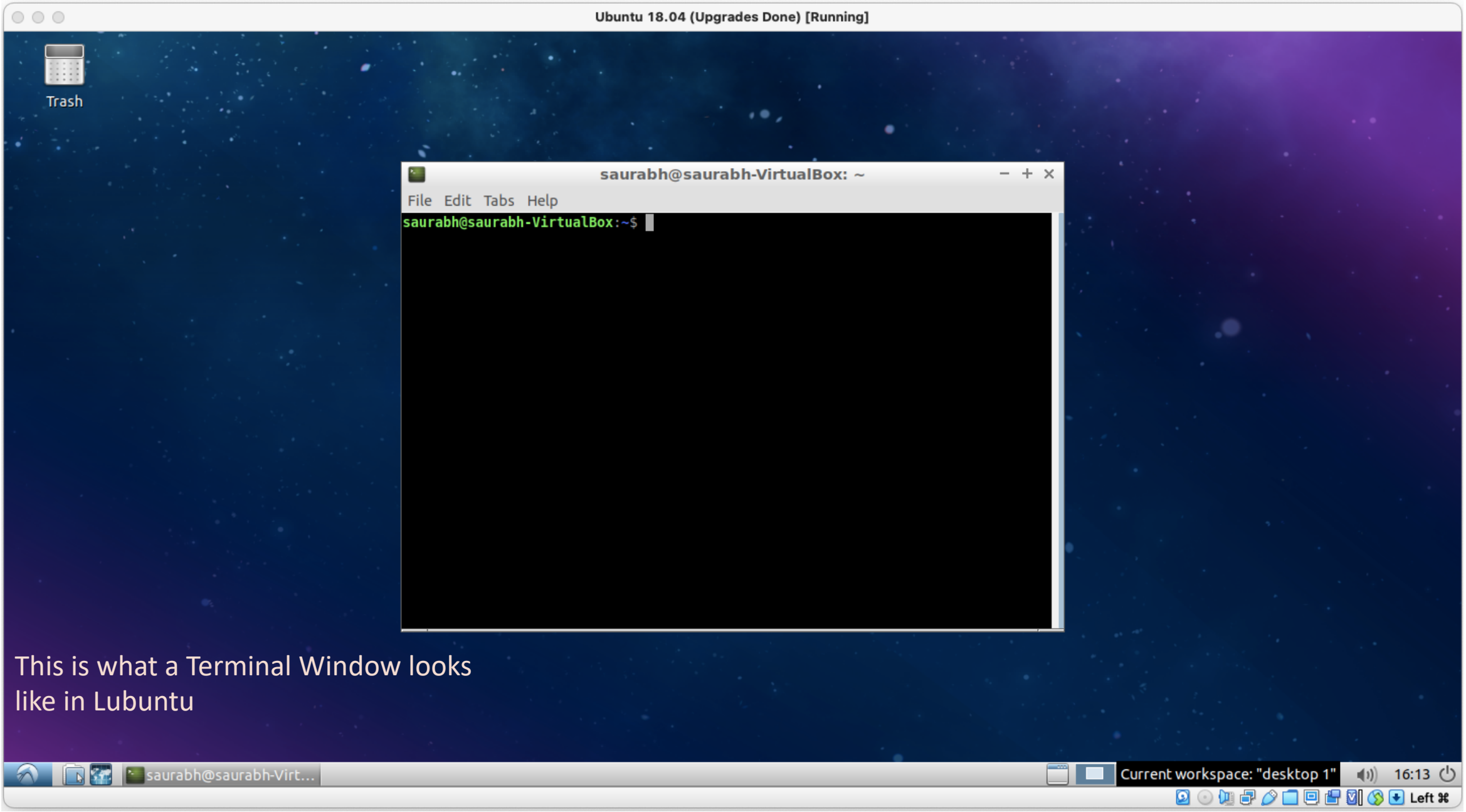
The Terminal application is more like a “meta-application”

- You can use it to access other Applications installed on your Linux installation

You can open it either from the Menu or by using some keyboard short-cut

- On distributions from the Ubuntu family, the short-cut is `Ctrl + Alt + T`
  - The + here means that you keep the key on the left side pressed, and then press the key on its right
- If you have installed a server distribution, your OS is “basically a terminal only” !!


Ubuntu 18.04 (Upgrades Done) [Running]
Trash
saurabh@saurabh-VirtualBox: ~
File Edit Tabs Help
saurabh@saurabh-VirtualBox:~$
saurabh@saurabh-Virt...
Current workspace: "desktop 1"
16:13
Left ⌘



This is what a Terminal Window looks like in Lubuntu

The first bash experience

On your Linux installation, open the “Terminal” application

The Terminal application is more like a “meta-application”

- You can use it to access other Applications installed on your Linux installation

You can open it either from the Menu or by using some keyboard short-cut

- On distributions from the Ubuntu family, the short-cut is `Ctrl + Alt + T`
  - The + here means that you keep the key on the left side pressed, and then press the key on its right
- If you have installed a server distribution, your OS is “basically a terminal only” !!

In a Terminal, you interact with a base Application – called a shell

- Although there are multiple types of shells, the most popular (and the default in Lubuntu) is *bash*

The first bash experience

On your Linux installation, open the “Terminal” application

The Terminal application is more like a “meta-application”

- You can use it to access other Applications installed on your Linux installation

You can open it either from the Menu or by using some keyboard short-cut

- On distributions from the Ubuntu family, the short-cut is `Ctrl + Alt + T`
  - The + here means that you keep the key on the left side pressed, and then press the key on its right
- If you have installed a server distribution, your OS is “basically a terminal only” !!

Let us create our first file using bash

The first bash experience

On your Linux installation, open the “Terminal” application

The Terminal application is more like a “meta-application”

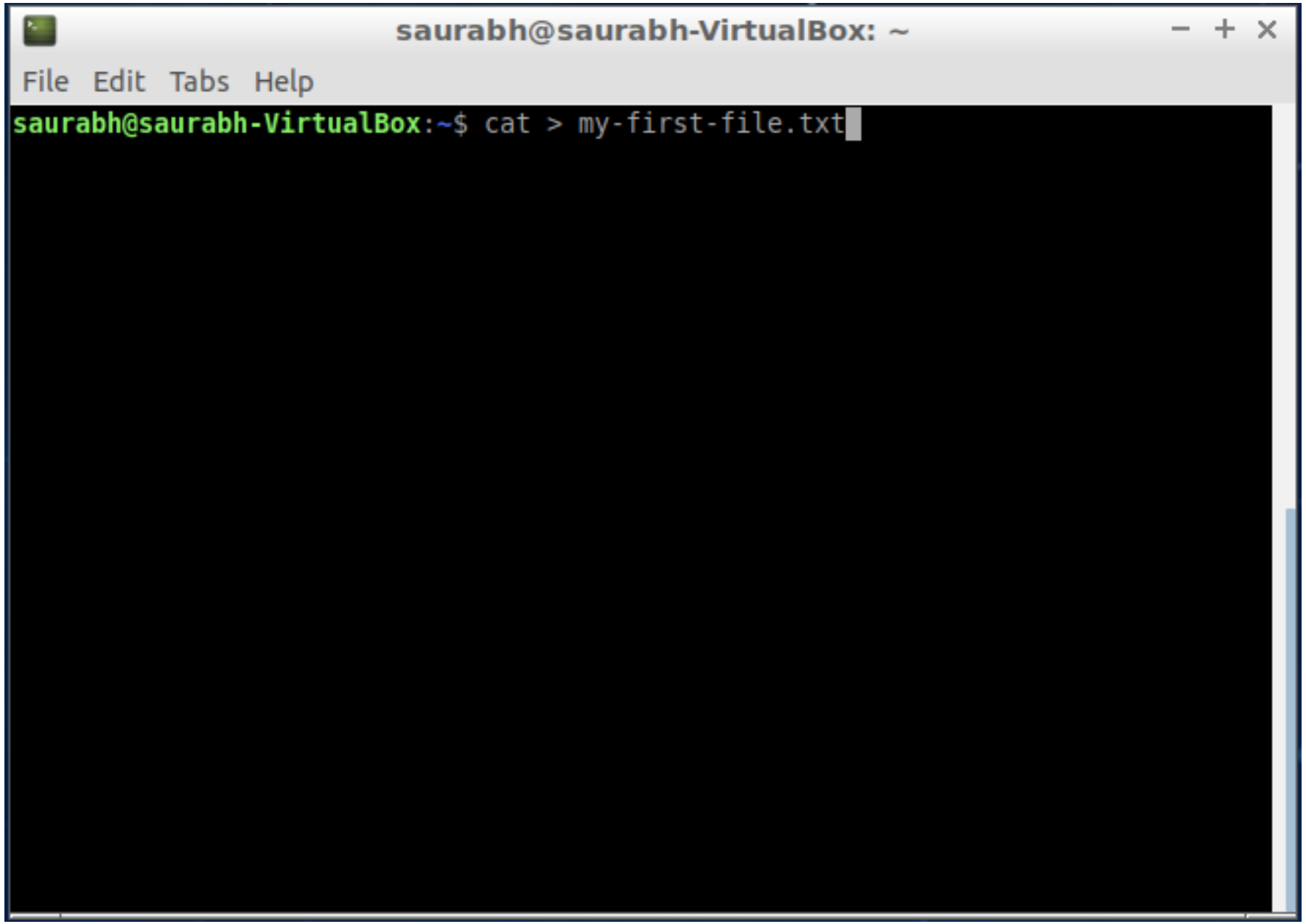
- You can use it to access other Applications installed on your Linux installation

You can open it either from the Menu or by using some keyboard short-cut

- On distributions from the Ubuntu family, the short-cut is `Ctrl + Alt + T`
  - The + here means that you keep the key on the left side pressed, and then press the key on its right
- If you have installed a server distribution, your OS is “basically a terminal only” !!

Let us create our first file using bash

- Type the command: `cat > my-first-file.txt`


saurabh@saurabh-VirtualBox: ~
File Edit Tabs Help
saurabh@saurabh-VirtualBox:~$ cat > my-first-file.txt

The first bash experience

On your Linux installation, open the “Terminal” application

The Terminal application is more like a “meta-application”

- You can use it to access other Applications installed on your Linux installation

You can open it either from the Menu or by using some keyboard short-cut

- On distributions from the Ubuntu family, the short-cut is `Ctrl + Alt + T`
  - The + here means that you keep the key on the left side pressed, and then press the key on its right
- If you have installed a server distribution, your OS is “basically a terminal only” !!

Let us create our first file using bash

- Type the command: `cat > my-first-file.txt`
- This will take the cursor to the next line, expecting some input from you

The first bash experience

On your Linux installation, open the “Terminal” application

The Terminal application is more like a “meta-application”

- You can use it to access other Applications installed on your Linux installation

You can open it either from the Menu or by using some keyboard short-cut

- On distributions from the Ubuntu family, the short-cut is `Ctrl + Alt + T`
  - The + here means that you keep the key on the left side pressed, and then press the key on its right
- If you have installed a server distribution, your OS is “basically a terminal only” !!

Let us create our first file using bash

- Type the command: `cat > my-first-file.txt`
- This will take the cursor to the next line, expecting some input from you
- Type the content of the file – you can type it in multiple lines

saurabh@saurabh-VirtualBox: ~

File Edit Tabs Help

```
saurabh@saurabh-VirtualBox:~$ cat > my-first-file.txt
This is my first file on Linux.
I am using the cat Application to create this file.
We are invoking the cat Application from bash.
```

The first bash experience

On your Linux installation, open the “Terminal” application

The Terminal application is more like a “meta-application”

- You can use it to access other Applications installed on your Linux installation

You can open it either from the Menu or by using some keyboard short-cut

- On distributions from the Ubuntu family, the short-cut is `Ctrl + Alt + T`
  - The + here means that you keep the key on the left side pressed, and then press the key on its right
- If you have installed a server distribution, your OS is “basically a terminal only” !!

Let us create our first file using bash

- Type the command: `cat > my-first-file.txt`
- This will take the cursor to the next line, expecting some input from you
- Type the content of the file – you can type it in multiple lines
- When you are done, press `Ctrl + D`, the prompt from where you started, will return

```
saurabh@saurabh-VirtualBox: ~
File  Edit  Tabs  Help
saurabh@saurabh-VirtualBox:~$ cat > my-first-file.txt
This is my first file on Linux.
I am using the cat Application to create this file.
We are invoking the cat Application from bash.
saurabh@saurabh-VirtualBox:~$
```

```
saurabh@saurabh-VirtualBox: ~
File  Edit  Tabs  Help
saurabh@saurabh-VirtualBox:~$ cat > my-first-file.txt
This is my first file on Linux.
I am using the cat Application to create this file.
We are invoking the cat Application from bash.
saurabh@saurabh-VirtualBox:~$
```

If you do not see any message after the content of the file, it means the command completed successfully !!

```
saurabh@saurabh-VirtualBox: ~
File  Edit  Tabs  Help
saurabh@saurabh-VirtualBox:~$ cat > my-first-file.txt
This is my first file on Linux.
I am using the cat Application to create this file.
We are invoking the cat Application from bash.
saurabh@saurabh-VirtualBox:~$ cat my-first-file.txt
This is my first file on Linux.
I am using the cat Application to create this file.
We are invoking the cat Application from bash.
saurabh@saurabh-VirtualBox:~$
```

```
saurabh@saurabh-VirtualBox: ~
File  Edit  Tabs  Help
saurabh@saurabh-VirtualBox:~$ cat > my-first-file.txt
This is my first file on Linux.
I am using the cat Application to create this file.
We are invoking the cat Application from bash.
saurabh@saurabh-VirtualBox:~$ cat my-first-file.txt
This is my first file on Linux.
I am using the cat Application to create this file.
We are invoking the cat Application from bash.
saurabh@saurabh-VirtualBox:~$
```

Type the same command again, without the “>”, and the cat command will show you the contents of your newly created file !!

Using `man` pages

Another useful application that can be extremely helpful on a Terminal is `man`

- Here, `man` stands for manual

Using `man` pages

Another useful application that can be extremely helpful on a Terminal is `man`

- Here, `man` stands for manual

The `man` command is your help source to know more about another command

- By the way, I am using the term “command” interchangeably with “Application” here
- In general, you type the name of the command you need help with, right after man

saurabh@saurabh-VirtualBox: ~

File Edit Tabs Help

```
saurabh@saurabh-VirtualBox:~$ cat > my-first-file.txt
This is my first file on Linux.
I am using the cat Application to create this file.
We are invoking the cat Application from bash.
saurabh@saurabh-VirtualBox:~$ cat my-first-file.txt
This is my first file on Linux.
I am using the cat Application to create this file.
We are invoking the cat Application from bash.
saurabh@saurabh-VirtualBox:~$ man cat
```

```
saurabh@saurabh-VirtualBox: ~
File  Edit  Tabs  Help
CAT(1)                    User Commands                    CAT(1)

NAME
       cat - concatenate files and print on the standard output

SYNOPSIS
       cat [OPTION]... [FILE]...

DESCRIPTION
       Concatenate FILE(s) to standard output.

       With no FILE, or when FILE is -, read standard input.

       -A, --show-all
              equivalent to -vET

       -b, --number-nonblank
              number nonempty output lines, overrides -n

       -e     equivalent to -vE

       -E, --show-ends
              display $ at end of each line
 Manual page cat(1) line 1 (press h for help or q to quit)
```

Some more popular bash commands - I

The `ls` command

- The `ls` command shows you the list of files inside the “current directory”
  - A directory is a collection of files; it is usually hierarchical, i.e. a directory can have files as well as other directories

```
saurabh@saurabh-VirtualBox: ~
File  Edit  Tabs  Help
saurabh@saurabh-VirtualBox:~$ ls
Desktop    Downloads  Music             Pictures  Templates
Documents  examples   my-first-file.txt  Public    Videos
saurabh@saurabh-VirtualBox:~$
```

```
saurabh@saurabh-VirtualBox: ~
File  Edit  Tabs  Help
saurabh@saurabh-VirtualBox:~$ ls
Desktop    Downloads  Music             Pictures  Templates
Documents  examples   my-first-file.txt  Public    Videos
saurabh@saurabh-VirtualBox:~$
```

The entries in blue are sub-directories of the current directory

Some more popular bash commands - I

The `ls` command

- The `ls` command shows you the list of files inside the “current directory”
  - A directory is a collection of files; it is usually hierarchical, i.e. a directory can have files as well as other directories
- Use the `-l` suffix after `ls`, to see many more details associated with the entries shown with `ls`

saurabh@saurabh-VirtualBox: ~

File Edit Tabs Help

```
saurabh@saurabh-VirtualBox:~$ ls
Desktop    Downloads  Music              Pictures  Templates
Documents  examples   my-first-file.txt  Public    Videos
saurabh@saurabh-VirtualBox:~$ ls -l
total 40
drwxr-xr-x 2 saurabh saurabh 4096 Oct 26 13:20 Desktop
drwxr-xr-x 2 saurabh saurabh 4096 Oct 26 13:20 Documents
drwxr-xr-x 2 saurabh saurabh 4096 Oct 26 13:20 Downloads
drwxrwxr-x 3 saurabh saurabh 4096 Oct 26 14:35 examples
drwxr-xr-x 2 saurabh saurabh 4096 Oct 26 13:20 Music
-rw-rw-r-- 1 saurabh saurabh  131 Dec 12 16:39 my-first-file.txt
drwxr-xr-x 2 saurabh saurabh 4096 Oct 26 13:20 Pictures
drwxr-xr-x 2 saurabh saurabh 4096 Oct 26 13:20 Public
drwxr-xr-x 2 saurabh saurabh 4096 Oct 26 13:20 Templates
drwxr-xr-x 2 saurabh saurabh 4096 Oct 26 13:20 Videos
saurabh@saurabh-VirtualBox:~$
```

```
saurabh@saurabh-VirtualBox: ~
File  Edit  Tabs  Help
saurabh@saurabh-VirtualBox:~$ ls
Desktop    Downloads  Music              Pictures  Templates
Documents  examples   my-first-file.txt  Public    Videos
saurabh@saurabh-VirtualBox:~$ ls -l
total 40
drwxr-xr-x 2 saurabh saurabh 4096 Oct 26 13:20 Desktop
drwxr-xr-x 2 saurabh saurabh 4096 Oct 26 13:20 Documents
drwxr-xr-x 2 saurabh saurabh 4096 Oct 26 13:20 Downloads
drwxrwxr-x 3 saurabh saurabh 4096 Oct 26 14:35 examples
drwxr-xr-x 2 saurabh saurabh 4096 Oct 26 13:20 Music
-rw-rw-r-- 1 saurabh saurabh  131 Dec 12 16:39 my-first-file.txt
drwxr-xr-x 2 saurabh saurabh 4096 Oct 26 13:20 Pictures
drwxr-xr-x 2 saurabh saurabh 4096 Oct 26 13:20 Public
drwxr-xr-x 2 saurabh saurabh 4096 Oct 26 13:20 Templates
drwxr-xr-x 2 saurabh saurabh 4096 Oct 26 13:20 Videos
saurabh@saurabh-VirtualBox:~$
```

Use ”`man ls`“ command to see the details of other suffixes that you can use with `ls`

Some more popular bash commands - I

The `ls` command

- The `ls` command shows you the list of files inside the “current directory”
  - A directory is a collection of files; it is usually hierarchical, i.e. a directory can have files as well as other directories
- Use the `-l` suffix after `ls`, to see many more details associated with the entries shown with `ls`

Every command has an associated *syntax*

Some more popular bash commands - I

The `ls` command

- The `ls` command shows you the list of files inside the “current directory”
  - A directory is a collection of files; it is usually hierarchical, i.e. a directory can have files as well as other directories
- Use the `-l` suffix after `ls`, to see many more details associated with the entries shown with `ls`

Every command has an associated *syntax*

- Syntax is a common term that you will keep hearing a lot – it is the equivalent of “Grammar” in English
  - It means that for a command to make sense, you must provide it in a particular format
  - You can see the syntax of a particular command using its associated `man` page

Some more popular bash commands - I

The `ls` command

- The `ls` command shows you the list of files inside the “current directory”
  - A directory is a collection of files; it is usually hierarchical, i.e. a directory can have files as well as other directories
- Use the `-l` suffix after `ls`, to see many more details associated with the entries shown with `ls`

Every command has an associated *syntax*

- Syntax is a common term that you will keep hearing a lot – it is the equivalent of “Grammar” in English
  - It means that for a command to make sense, you must provide it in a particular format
  - You can see the syntax of a particular command using its associated `man` page
- The suffixes to a command are also called *switches*

Some more popular bash commands - II

The `cd` command

- The `cd` command can change the current directory to some other directory on your file system

saurabh@saurabh-VirtualBox: ~/examples/introduction

File Edit Tabs Help

```
saurabh@saurabh-VirtualBox:~$ ls
Desktop    Downloads  Music              Pictures  Templates
Documents  examples   my-first-file.txt  Public    Videos
saurabh@saurabh-VirtualBox:~$ cd examples/
saurabh@saurabh-VirtualBox:~/examples$ ls
introduction
saurabh@saurabh-VirtualBox:~/examples$ cd introduction/
saurabh@saurabh-VirtualBox:~/examples/introduction$ ls
HelloWorld.class  HelloWorld.java
saurabh@saurabh-VirtualBox:~/examples/introduction$
```

```
saurabh@saurabh-VirtualBox: ~/examples/introduction
File  Edit  Tabs  Help
saurabh@saurabh-VirtualBox:~$ ls
Desktop    Downloads  Music              Pictures  Templates
Documents  examples   my-first-file.txt  Public    Videos
saurabh@saurabh-VirtualBox:~$ cd examples/
saurabh@saurabh-VirtualBox:~/examples$ ls
introduction
saurabh@saurabh-VirtualBox:~/examples$ cd introduction/
saurabh@saurabh-VirtualBox:~/examples/introduction$ ls
HelloWorld.class  HelloWorld.java
saurabh@saurabh-VirtualBox:~/examples/introduction$
```

Observe the hierarchical nature of directories here

Some more popular bash commands - II

The `cd` command

- The `cd` command can change the current directory to some other directory on your file system
- If you use `cd` command without a directory name, your current directory is changed to your home directory

saurabh@saurabh-VirtualBox: ~

File Edit Tabs Help

```
saurabh@saurabh-VirtualBox:~$ ls
Desktop    Downloads  Music              Pictures  Templates
Documents  examples   my-first-file.txt  Public    Videos
saurabh@saurabh-VirtualBox:~$ cd examples/
saurabh@saurabh-VirtualBox:~/examples$ ls
introduction
saurabh@saurabh-VirtualBox:~/examples$ cd introduction/
saurabh@saurabh-VirtualBox:~/examples/introduction$ ls
HelloWorld.class  HelloWorld.java
saurabh@saurabh-VirtualBox:~/examples/introduction$ cd
saurabh@saurabh-VirtualBox:~$ ls
Desktop    Downloads  Music              Pictures  Templates
Documents  examples   my-first-file.txt  Public    Videos
saurabh@saurabh-VirtualBox:~$
```

```
saurabh@saurabh-VirtualBox: ~
File  Edit  Tabs  Help
saurabh@saurabh-VirtualBox:~$ ls
Desktop    Downloads  Music              Pictures  Templates
Documents  examples   my-first-file.txt  Public    Videos
saurabh@saurabh-VirtualBox:~$ cd examples/
saurabh@saurabh-VirtualBox:~/examples$ ls
introduction
saurabh@saurabh-VirtualBox:~/examples$ cd introduction/
saurabh@saurabh-VirtualBox:~/examples/introduction$ ls
HelloWorld.class  HelloWorld.java
saurabh@saurabh-VirtualBox:~/examples/introduction$ cd
saurabh@saurabh-VirtualBox:~$ ls
Desktop    Downloads  Music              Pictures  Templates
Documents  examples   my-first-file.txt  Public    Videos
saurabh@saurabh-VirtualBox:~$
```

...and we are back to the home directory

Some more popular bash commands - II

The `cd` command

- The `cd` command can change the current directory to some other directory on your file system
- If you use `cd` command without a directory name, your current directory is changed to your home directory
- On Linux, current directory is represented by a dot (`.`) and the parent directory of a directory by two dots (`..`)

saurabh@saurabh-VirtualBox: ~

File Edit Tabs Help

```
saurabh@saurabh-VirtualBox:~$ ls
Desktop    Downloads  Music              Pictures  Templates
Documents  examples   my-first-file.txt  Public    Videos
saurabh@saurabh-VirtualBox:~$ cd examples/
saurabh@saurabh-VirtualBox:~/examples$ ls
introduction
saurabh@saurabh-VirtualBox:~/examples$ cd introduction/
saurabh@saurabh-VirtualBox:~/examples/introduction$ ls
HelloWorld.class  HelloWorld.java
saurabh@saurabh-VirtualBox:~/examples/introduction$ cd .
saurabh@saurabh-VirtualBox:~/examples/introduction$ cd ..
saurabh@saurabh-VirtualBox:~/examples$ cd ..
saurabh@saurabh-VirtualBox:~$
```

```
saurabh@saurabh-VirtualBox: ~
File  Edit  Tabs  Help
saurabh@saurabh-VirtualBox:~$ ls
Desktop    Downloads  Music              Pictures  Templates
Documents  examples   my-first-file.txt  Public    Videos
saurabh@saurabh-VirtualBox:~$ cd examples/
saurabh@saurabh-VirtualBox:~/examples$ ls
introduction
saurabh@saurabh-VirtualBox:~/examples$ cd introduction/
saurabh@saurabh-VirtualBox:~/examples/introduction$ ls
HelloWorld.class  HelloWorld.java
saurabh@saurabh-VirtualBox:~/examples/introduction$ cd .
saurabh@saurabh-VirtualBox:~/examples/introduction$ cd ..
saurabh@saurabh-VirtualBox:~/examples$ cd ..
saurabh@saurabh-VirtualBox:~$
```

The "`cd ..`" command can take you up in the directory hierarchy, one step at a time

Some more popular bash commands - II

The `cd` command

- The `cd` command can change the current directory to some other directory on your file system
- If you use `cd` command without a directory name, your current directory is changed to your home directory
- On Linux, current directory is represented by a dot (`.`) and the parent directory of a directory by two dots (`..`)

Your home directory is the place on your Linux installation where your personal files reside

- On Ubuntu, this directory is named something like `/home/saurabh`
  - Given that the username of your account is `saurabh`

Some more popular bash commands - II

The `cd` command

- The `cd` command can change the current directory to some other directory on your file system
- If you use `cd` command without a directory name, your current directory is changed to your home directory
- On Linux, current directory is represented by a dot (`.`) and the parent directory of a directory by two dots (`..`)

Your home directory is the place on your Linux installation where your personal files reside

- On Ubuntu, this directory is named something like `/home/saurabh`
  - Given that the username of your account is `saurabh`

The `pwd` command

- The `pwd` command shows the *full path* of the current directory

saurabh@saurabh-VirtualBox: ~

File Edit Tabs Help

```
saurabh@saurabh-VirtualBox:~$ ls
Desktop    Downloads  Music              Pictures  Templates
Documents  examples   my-first-file.txt  Public    Videos
saurabh@saurabh-VirtualBox:~$ cd examples/
saurabh@saurabh-VirtualBox:~/examples$ ls
introduction
saurabh@saurabh-VirtualBox:~/examples$ cd introduction/
saurabh@saurabh-VirtualBox:~/examples/introduction$ ls
HelloWorld.class  HelloWorld.java
saurabh@saurabh-VirtualBox:~/examples/introduction$ cd
saurabh@saurabh-VirtualBox:~$ ls
Desktop    Downloads  Music              Pictures  Templates
Documents  examples   my-first-file.txt  Public    Videos
saurabh@saurabh-VirtualBox:~$ pwd
/home/saurabh
saurabh@saurabh-VirtualBox:~$
```

Some more popular bash commands - II

The `cd` command

- The `cd` command can change the current directory to some other directory on your file system
- If you use `cd` command without a directory name, your current directory is changed to your home directory
- On Linux, current directory is represented by a dot (`.`) and the parent directory of a directory by two dots (`..`)

Your home directory is the place on your Linux installation where your personal files reside

- On Ubuntu, this directory is named something like `/home/saurabh`
  - Given that the username of your account is `saurabh`

The `pwd` command

- The `pwd` command shows the *full path* of the current directory

The path of a directory shows you its position in the directory hierarchy

Some more popular bash commands - II

The `cd` command

- The `cd` command can change the current directory to some other directory on your file system
- If you use `cd` command without a directory name, your current directory is changed to your home directory
- On Linux, current directory is represented by a dot (`.`) and the parent directory of a directory by two dots (`..`)

Your home directory is the place on your Linux installation where your personal files reside

- On Ubuntu, this directory is named something like `/home/saurabh`
  - Given that the username of your account is `saurabh`

The `pwd` command

- The `pwd` command shows the *full path* of the current directory

The path of a directory shows you its position in the directory hierarchy

- `/` is the top-most directory in Linux
- On Ubuntu, home is a sub-directory of `/`, and your home directory is a sub-directory of `/home`

Some more popular bash commands - III

The `mkdir` command

- You can create a new directory using the `mkdir` command

saurabh@saurabh-VirtualBox: ~/test

File Edit Tabs Help

```
saurabh@saurabh-VirtualBox:~$ ls
Desktop    Downloads  Music              Pictures  Templates
Documents  examples   my-first-file.txt  Public    Videos
saurabh@saurabh-VirtualBox:~$ cd examples/
saurabh@saurabh-VirtualBox:~/examples$ ls
introduction
saurabh@saurabh-VirtualBox:~/examples$ cd introduction/
saurabh@saurabh-VirtualBox:~/examples/introduction$ ls
HelloWorld.class  HelloWorld.java
saurabh@saurabh-VirtualBox:~/examples/introduction$ cd
saurabh@saurabh-VirtualBox:~$ ls
Desktop    Downloads  Music              Pictures  Templates
Documents  examples   my-first-file.txt  Public    Videos
saurabh@saurabh-VirtualBox:~$ pwd
/home/saurabh
saurabh@saurabh-VirtualBox:~$ mkdir test
saurabh@saurabh-VirtualBox:~$ cd test
saurabh@saurabh-VirtualBox:~/test$ pwd
/home/saurabh/test
saurabh@saurabh-VirtualBox:~/test$
```

saurabh@saurabh-VirtualBox: ~/test

File Edit Tabs Help

```
saurabh@saurabh-VirtualBox:~$ ls
Desktop    Downloads  Music              Pictures  Templates
Documents  examples   my-first-file.txt  Public    Videos
saurabh@saurabh-VirtualBox:~$ cd examples/
saurabh@saurabh-VirtualBox:~/examples$ ls
introduction
saurabh@saurabh-VirtualBox:~/examples$ cd introduction/
saurabh@saurabh-VirtualBox:~/examples/introduction$ ls
HelloWorld.class  HelloWorld.java
saurabh@saurabh-VirtualBox:~/examples/introduction$ cd
saurabh@saurabh-VirtualBox:~$ ls
Desktop    Downloads  Music              Pictures  Templates
Documents  examples   my-first-file.txt  Public    Videos
saurabh@saurabh-VirtualBox:~$ pwd
/home/saurabh
saurabh@saurabh-VirtualBox:~$ mkdir test
saurabh@saurabh-VirtualBox:~$ cd test
saurabh@saurabh-VirtualBox:~/test$ pwd
/home/saurabh/test
saurabh@saurabh-VirtualBox:~/test$
```

We just created a new directory called `test` as a sub-directory of our home directory

Some more popular bash commands - III

The `mkdir` command

- You can create a new directory using the `mkdir` command

The `grep` command

- The grep command can search for files with specific text

```
saurabh@saurabh-VirtualBox: ~
File  Edit  Tabs  Help
saurabh@saurabh-VirtualBox:~$ pwd
/home/saurabh
saurabh@saurabh-VirtualBox:~$ grep -i linux *
grep: Desktop: Is a directory
grep: Documents: Is a directory
grep: Downloads: Is a directory
grep: examples: Is a directory
grep: Music: Is a directory
my-first-file.txt:This is my first file on Linux.
grep: Pictures: Is a directory
grep: Public: Is a directory
grep: Templates: Is a directory
grep: Videos: Is a directory
saurabh@saurabh-VirtualBox:~$ grep -i linux *.txt
This is my first file on Linux.
saurabh@saurabh-VirtualBox:~$
```

```
saurabh@saurabh-VirtualBox: ~
File  Edit  Tabs  Help
saurabh@saurabh-VirtualBox:~$ pwd
/home/saurabh
saurabh@saurabh-VirtualBox:~$ grep -i linux *
grep: Desktop: Is a directory
grep: Documents: Is a directory
grep: Downloads: Is a directory
grep: examples: Is a directory
grep: Music: Is a directory
my-first-file.txt:This is my first file on Linux.
grep: Pictures: Is a directory
grep: Public: Is a directory
grep: Templates: Is a directory
grep: Videos: Is a directory
saurabh@saurabh-VirtualBox:~$ grep -i linux *.txt
This is my first file on Linux.
saurabh@saurabh-VirtualBox:~$
```

Fire `man grep` as the next command to know what just happened !!

Some more popular bash commands - III

The `mkdir` command

- You can create a new directory using the `mkdir` command

The `grep` command

- The `grep` command can search for files with specific text

The `find` command

- The `find` command can search for files with a specific name

saurabh@saurabh-VirtualBox: ~

File Edit Tabs Help

```
saurabh@saurabh-VirtualBox:~$ find ./ -name "HelloWorld.java"
./examples/introduction/HelloWorld.java
saurabh@saurabh-VirtualBox:~$ find ./ -name "HelloWorld.*"
./examples/introduction/HelloWorld.class
./examples/introduction/HelloWorld.java
saurabh@saurabh-VirtualBox:~$ find ./ -name "*.java"
./examples/introduction/HelloWorld.java
saurabh@saurabh-VirtualBox:~$ find ./ -iname "hello*"
./examples/introduction/HelloWorld.class
./examples/introduction/HelloWorld.java
saurabh@saurabh-VirtualBox:~$ find ./ -name "hello*"
saurabh@saurabh-VirtualBox:~$
```

```
saurabh@saurabh-VirtualBox: ~
File  Edit  Tabs  Help
saurabh@saurabh-VirtualBox:~$ find ./ -name "HelloWorld.java"
./examples/introduction/HelloWorld.java
saurabh@saurabh-VirtualBox:~$ find ./ -name "HelloWorld.*"
./examples/introduction/HelloWorld.class
./examples/introduction/HelloWorld.java
saurabh@saurabh-VirtualBox:~$ find ./ -name "*.java"
./examples/introduction/HelloWorld.java
saurabh@saurabh-VirtualBox:~$ find ./ -iname "hello*"
./examples/introduction/HelloWorld.class
./examples/introduction/HelloWorld.java
saurabh@saurabh-VirtualBox:~$ find ./ -name "hello*"
saurabh@saurabh-VirtualBox:~$
```

Check the man page for find – you can understand the differences between these versions of the `find` command

Some more popular bash commands - III

The `mkdir` command

- You can create a new directory using the `mkdir` command

The `grep` command

- The `grep` command can search for files with specific text

The `find` command

- The `find` command can search for files with a specific name
- Note that "`./`" here, means "search for files in the current directory, and its subdirectories only"

Some more popular bash commands - III

The `mkdir` command

- You can create a new directory using the `mkdir` command

The `grep` command

- The `grep` command can search for files with specific text

The `find` command

- The `find` command can search for files with a specific name
- Note that "`./`" here, means "search for files in the current directory, and its subdirectories only"

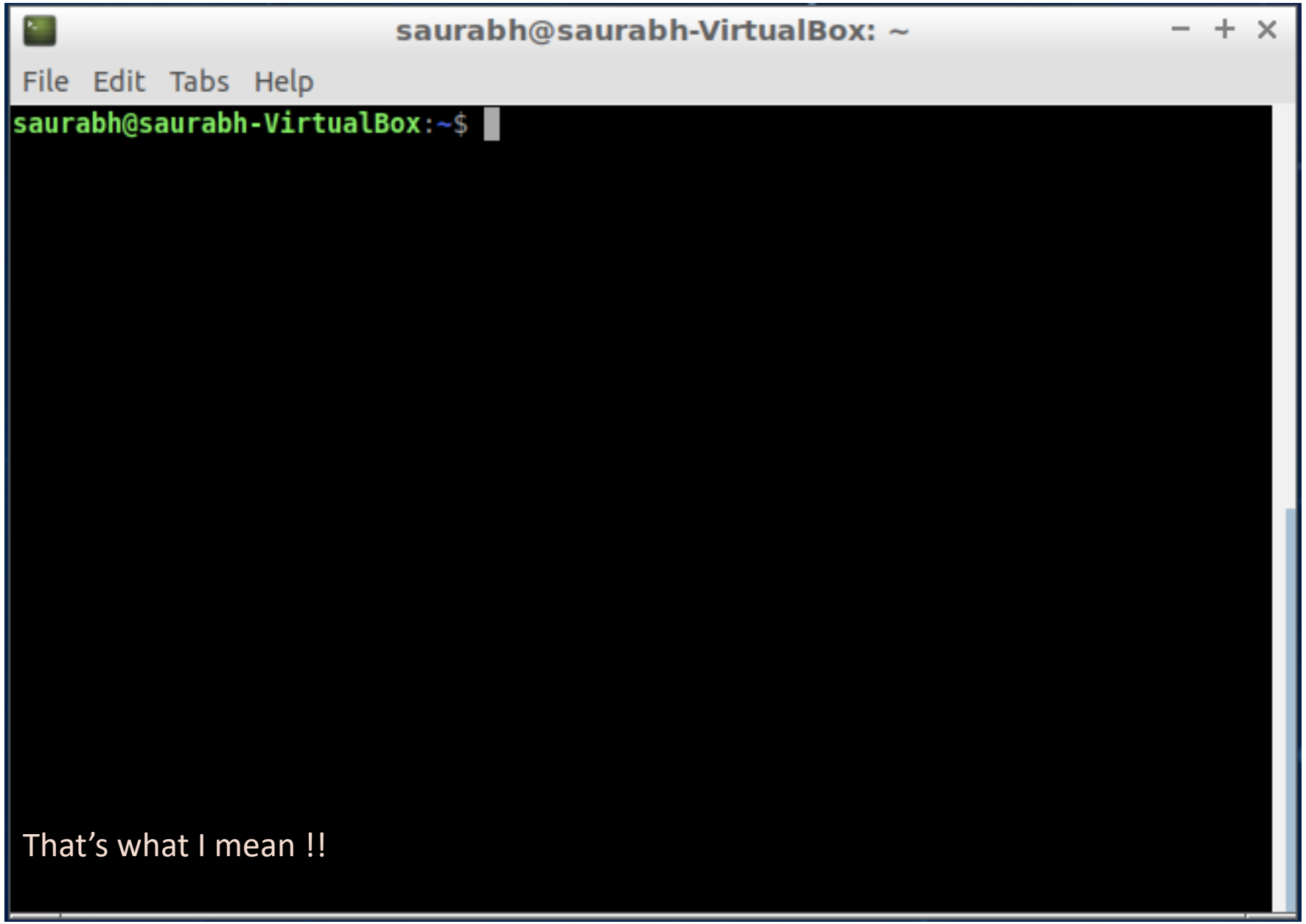
The `clear` command

- The `clear` command simply clears your Terminal window

saurabh@saurabh-VirtualBox: ~

File Edit Tabs Help

```
saurabh@saurabh-VirtualBox:~$ find ./ -name "HelloWorld.java"
./examples/introduction/HelloWorld.java
saurabh@saurabh-VirtualBox:~$ find ./ -name "HelloWorld.*"
./examples/introduction/HelloWorld.class
./examples/introduction/HelloWorld.java
saurabh@saurabh-VirtualBox:~$ find ./ -name "*.java"
./examples/introduction/HelloWorld.java
saurabh@saurabh-VirtualBox:~$ find ./ -iname "hello*"
./examples/introduction/HelloWorld.class
./examples/introduction/HelloWorld.java
saurabh@saurabh-VirtualBox:~$ find ./ -name "hello*"
saurabh@saurabh-VirtualBox:~$ clear
```

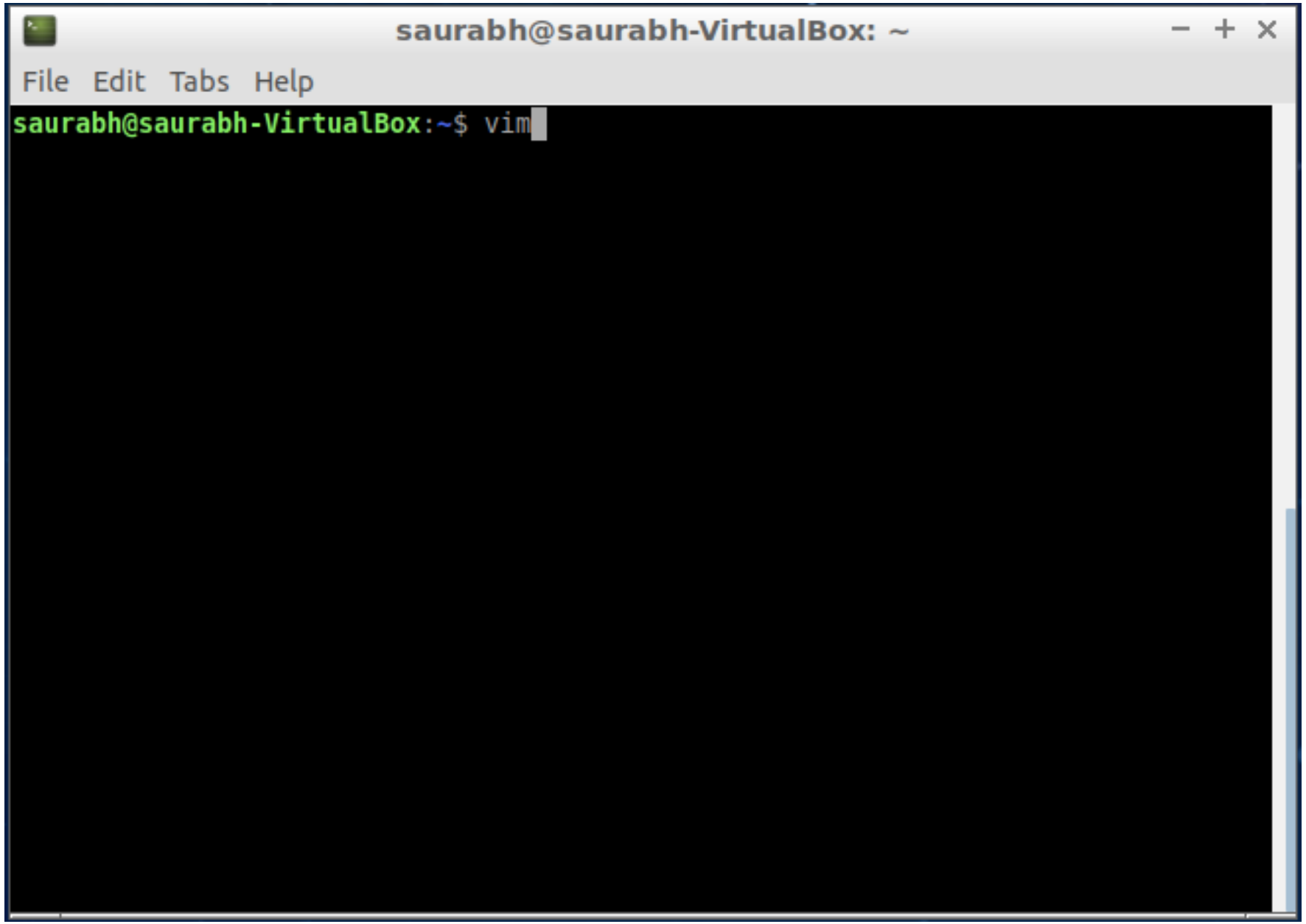


That’s what I mean !!

Editors on bash

If you want to edit your files, you usually have a number of options

- I personally use `vim` – it stands for "`vi` improved"


saurabh@saurabh-VirtualBox: ~
File Edit Tabs Help
saurabh@saurabh-VirtualBox:~$ vim

```
saurabh@saurabh-VirtualBox: ~
File  Edit  Tabs  Help

                    VIM - Vi IMproved

                     version 8.0.1453
                  by Bram Moolenaar et al.
        Modified by pkg-vim-maintainers@lists.alioth.debian.org
              Vim is open source and freely distributable

                  Sponsor Vim development!
           type  :help sponsor<Enter>    for information

           type  :q<Enter>               to exit
           type  :help<Enter>  or  <F1>  for on-line help
           type  :help version8<Enter>   for version info

                                                 0,0-1         All
```

```
saurabh@saurabh-VirtualBox: ~
File  Edit  Tabs  Help

                    VIM - Vi IMproved

                     version 8.0.1453
                  by Bram Moolenaar et al.
        Modified by pkg-vim-maintainers@lists.alioth.debian.org
              Vim is open source and freely distributable

                  Sponsor Vim development!
           type  :help sponsor<Enter>    for information

           type  :q<Enter>               to exit
           type  :help<Enter>  or  <F1>  for on-line help
           type  :help version8<Enter>   for version info
```

Disclaimer: Don't try this at home... I mean if you don't know how to exit `vim` (there is a hint on this screen though !!)

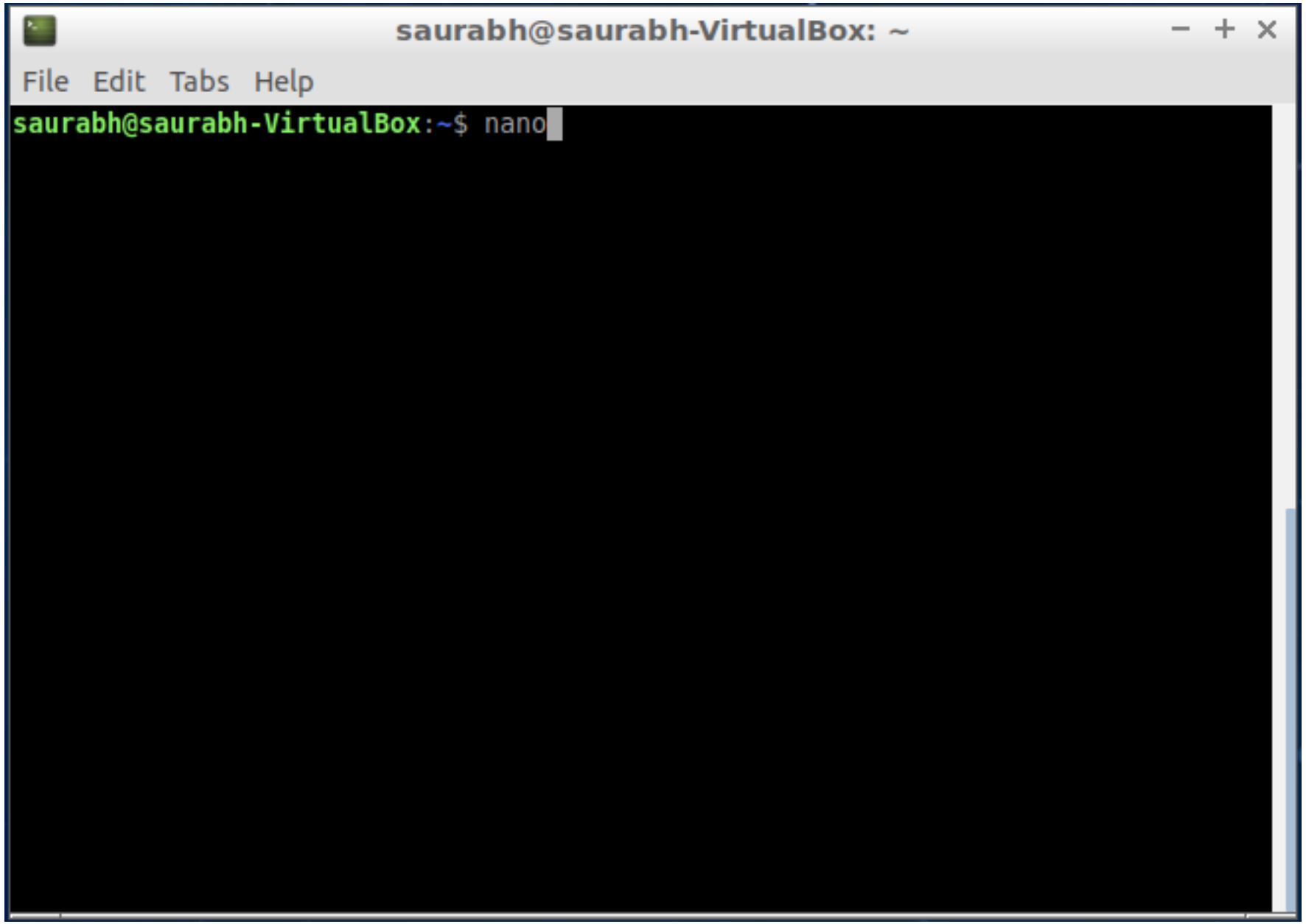
```
0,0-1         All
```

Editors on bash

If you want to edit your files, you usually have a number of options

- I personally use `vim` – it stands for "`vi` improved"
- You may not have `vim` already installed on your system, but there is a good chance that you have `vi`
- We can probably have a full course dedicated towards `vim`, so let us leave it here for you to explore
  - Do tell me your experiences though :-D

Another, not so controversial editor that you may use is `nano`


saurabh@saurabh-VirtualBox: ~
File Edit Tabs Help
saurabh@saurabh-VirtualBox:~$ nano

```
saurabh@saurabh-VirtualBox: ~
File  Edit  Tabs  Help
  GNU nano 2.9.3                    New Buffer

^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line
```

Editors on bash

If you want to edit your files, you usually have a number of options

- I personally use `vim` – it stands for "`vi` improved"
- You may not have `vim` already installed on your system, but there is a good chance that you have `vi`
- We can probably have a full course dedicated towards `vim`, so let us leave it here for you to explore
  - Do tell me your experiences though :-D

Another, not so controversial editor that you may use is `nano`

In addition, there are Applications like `sed`, which can be used to perform pattern-based edits

- `sed` is short for "Stream Editor" – it reads text like a stream of a river, and can make changes over it

```
saurabh@saurabh-VirtualBox: ~
File  Edit  Tabs  Help
saurabh@saurabh-VirtualBox:~$ cat my-first-file.txt
This is my first file on Linux.
I am using the cat Application to create this file.
We are invoking the cat Application from bash.
saurabh@saurabh-VirtualBox:~$ sed 's/first/second/g; s/cat /sed /g' my-first-fil
e.txt > my-second-file.txt
saurabh@saurabh-VirtualBox:~$ cat my-second-file.txt
This is my second file on Linux.
I am using the sed Application to create this file.
We are invoking the sed Application from bash.
saurabh@saurabh-VirtualBox:~$
```

```
saurabh@saurabh-VirtualBox: ~
File  Edit  Tabs  Help
saurabh@saurabh-VirtualBox:~$ cat my-first-file.txt
This is my first file on Linux.
I am using the cat Application to create this file.
We are invoking the cat Application from bash.
saurabh@saurabh-VirtualBox:~$ sed 's/first/second/g; s/cat /sed /g' my-first-fil
e.txt > my-second-file.txt
saurabh@saurabh-VirtualBox:~$ cat my-second-file.txt
This is my second file on Linux.
I am using the sed Application to create this file.
We are invoking the sed Application from bash.
saurabh@saurabh-VirtualBox:~$
```

This is just meant to excite you about learning bash…
you should really practice simpler commands before `sed` !!

The `sudo` command

Having a sharp knife can be really effective when you are cooking

- But you can accidently cut yourself too, if you are not careful

The `sudo` command

Having a sharp knife can be really effective when you are cooking

- But you can accidently cut yourself too, if you are not careful

On Linux systems, almost all the Applications, by default, run as a “user process”

- Basically, these Applications can do a lot, but not everything
- This is because there are some operations, which can be considered as “critical” by Linux
- For example, installing or removing Applications, or killing an Application started by another user

The `sudo` command

Having a sharp knife can be really effective when you are cooking

- But you can accidently cut yourself too, if you are not careful

On Linux systems, almost all the Applications, by default, run as a “user process”

- Basically, these Applications can do a lot, but not everything
- This is because there are some operations, which can be considered as “critical” by Linux
- For example, installing or removing Applications, or killing an Application started by another user

Linux has a mechanism to prevent certain commands, if you are “not an administrator”

- If you wish to take the administrator role, you can do so using the `sudo` command
- `sudo` stands for “superuser do” – where superuser is just a fancy name for the system’s administrator
- You just add the word `sudo` before a regular command, to run the same as “`root`”
- `root` is a polymorphic term in the Linux ecosystem, which broadly implies “superuser privileges”

saurabh@saurabh-VirtualBox: ~

File Edit Tabs Help

```
saurabh@saurabh-VirtualBox:~$ apt install build-essential
E: Could not open lock file /var/lib/dpkg/lock-frontend - open (13: Permission d
enied)
E: Unable to acquire the dpkg frontend lock (/var/lib/dpkg/lock-frontend), are y
ou root?
saurabh@saurabh-VirtualBox:~$
```

```
saurabh@saurabh-VirtualBox: ~
File  Edit  Tabs  Help
saurabh@saurabh-VirtualBox:~$ apt install build-essential
E: Could not open lock file /var/lib/dpkg/lock-frontend - open (13: Permission d
enied)
E: Unable to acquire the dpkg frontend lock (/var/lib/dpkg/lock-frontend), are y
ou root?
saurabh@saurabh-VirtualBox:~$
```

`apt` is a package management tool on Ubuntu – basically, you can use it to install or remove Applications

```
saurabh@saurabh-VirtualBox: ~
File  Edit  Tabs  Help
saurabh@saurabh-VirtualBox:~$ apt install build-essential
E: Could not open lock file /var/lib/dpkg/lock-frontend - open (13: Permission d
enied)
E: Unable to acquire the dpkg frontend lock (/var/lib/dpkg/lock-frontend), are y
ou root?
saurabh@saurabh-VirtualBox:~$
```

The error message literally tells you that you are “not root”, albeit, in a polite fashion…

```
saurabh@saurabh-VirtualBox: ~
File  Edit  Tabs  Help
saurabh@saurabh-VirtualBox:~$ apt install build-essential
E: Could not open lock file /var/lib/dpkg/lock-frontend - open (13: Permission d
enied)
E: Unable to acquire the dpkg frontend lock (/var/lib/dpkg/lock-frontend), are y
ou root?
saurabh@saurabh-VirtualBox:~$ sudo apt install build-essential
[sudo] password for saurabh:
```

```
saurabh@saurabh-VirtualBox: ~
File  Edit  Tabs  Help
saurabh@saurabh-VirtualBox:~$ apt install build-essential
E: Could not open lock file /var/lib/dpkg/lock-frontend - open (13: Permission d
enied)
E: Unable to acquire the dpkg frontend lock (/var/lib/dpkg/lock-frontend), are y
ou root?
saurabh@saurabh-VirtualBox:~$ sudo apt install build-essential
[sudo] password for saurabh:
```

So, let's give it what it wants – the `sudo` before the command…
and your password !!

```
saurabh@saurabh-VirtualBox: ~
File  Edit  Tabs  Help
saurabh@saurabh-VirtualBox:~$ apt install build-essential
E: Could not open lock file /var/lib/dpkg/lock-frontend - open (13: Permission d
enied)
E: Unable to acquire the dpkg frontend lock (/var/lib/dpkg/lock-frontend), are y
ou root?
saurabh@saurabh-VirtualBox:~$ sudo apt install build-essential
[sudo] password for saurabh:
```

Note: this does assume that the user "`saurabh`" here, is indeed, a superuser !! Otherwise, you'll get a polite scolding ;)

saurabh@saurabh-VirtualBox: ~

File Edit Tabs Help

```
22-2ubuntu1 [25.9 kB]
Get:9 http://in.archive.ubuntu.com/ubuntu bionic/main amd64 fakeroot amd64 1.22-
2ubuntu1 [62.3 kB]
Get:10 http://in.archive.ubuntu.com/ubuntu bionic/main amd64 libalgorithm-diff-p
erl all 1.19.03-1 [47.6 kB]
Get:11 http://in.archive.ubuntu.com/ubuntu bionic/main amd64 libalgorithm-diff-x
s-perl amd64 0.04-5 [11.1 kB]
Get:12 http://in.archive.ubuntu.com/ubuntu bionic/main amd64 libalgorithm-merge-
perl all 0.08-3 [12.0 kB]
Get:13 http://in.archive.ubuntu.com/ubuntu bionic/main amd64 libfile-fcntllock-p
erl amd64 0.22-3build2 [33.2 kB]
Fetched 12.2 MB in 2s (6,197 kB/s)
Selecting previously unselected package gcc.
(Reading database ... 159080 files and directories currently installed.)
Preparing to unpack .../00-gcc_4%3a7.4.0-1ubuntu2.3_amd64.deb ...
Unpacking gcc (4:7.4.0-1ubuntu2.3) ...
Selecting previously unselected package libstdc++-7-dev:amd64.
Preparing to unpack .../01-libstdc++-7-dev_7.5.0-3ubuntu1~18.04_amd64.deb ...
Unpacking libstdc++-7-dev:amd64 (7.5.0-3ubuntu1~18.04) ...
Selecting previously unselected package g++-7.
Preparing to unpack .../02-g++-7_7.5.0-3ubuntu1~18.04_amd64.deb ...
Unpacking g++-7 (7.5.0-3ubuntu1~18.04) ...

Progress: [ 11%] [######.....................................................]
```

saurabh@saurabh-VirtualBox: ~

File Edit Tabs Help

```
Preparing to unpack .../12-libfile-fcntllock-perl_0.22-3build2_amd64.deb ...
Unpacking libfile-fcntllock-perl (0.22-3build2) ...
Setting up libdpkg-perl (1.19.0.5ubuntu2.3) ...
Setting up gcc (4:7.4.0-1ubuntu2.3) ...
Setting up libstdc++-7-dev:amd64 (7.5.0-3ubuntu1~18.04) ...
Setting up libfile-fcntllock-perl (0.22-3build2) ...
Setting up dpkg-dev (1.19.0.5ubuntu2.3) ...
Setting up libfakeroot:amd64 (1.22-2ubuntu1) ...
Setting up libalgorithm-diff-perl (1.19.03-1) ...
Setting up g++-7 (7.5.0-3ubuntu1~18.04) ...
Setting up fakeroot (1.22-2ubuntu1) ...
update-alternatives: using /usr/bin/fakeroot-sysv to provide /usr/bin/fakeroot (
fakeroot) in auto mode
Setting up libalgorithm-merge-perl (0.08-3) ...
Setting up libalgorithm-diff-xs-perl (0.04-5) ...
Setting up g++ (4:7.4.0-1ubuntu2.3) ...
update-alternatives: using /usr/bin/g++ to provide /usr/bin/c++ (c++) in auto mo
de
Setting up build-essential (12.4ubuntu1) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for libc-bin (2.27-3ubuntu1.3) ...
saurabh@saurabh-VirtualBox:~$
```

```
saurabh@saurabh-VirtualBox: ~
File  Edit  Tabs  Help
Preparing to unpack .../12-libfile-fcntllock-perl_0.22-3build2_amd64.deb ...
Unpacking libfile-fcntllock-perl (0.22-3build2) ...
Setting up libdpkg-perl (1.19.0.5ubuntu2.3) ...
Setting up gcc (4:7.4.0-1ubuntu2.3) ...
Setting up libstdc++-7-dev:amd64 (7.5.0-3ubuntu1~18.04) ...
Setting up libfile-fcntllock-perl (0.22-3build2) ...
Setting up dpkg-dev (1.19.0.5ubuntu2.3) ...
Setting up libfakeroot:amd64 (1.22-2ubuntu1) ...
Setting up libalgorithm-diff-perl (1.19.03-1) ...
Setting up g++-7 (7.5.0-3ubuntu1~18.04) ...
Setting up fakeroot (1.22-2ubuntu1) ...
update-alternatives: using /usr/bin/fakeroot-sysv to provide /usr/bin/fakeroot (
fakeroot) in auto mode
Setting up libalgorithm-merge-perl (0.08-3) ...
Setting up libalgorithm-diff-xs-perl (0.04-5) ...
Setting up g++ (4:7.4.0-1ubuntu2.3) ...
update-alternatives: using /usr/bin/g++ to provide /usr/bin/c++ (c++) in auto mo
de
Setting up build-essential (12.4ubuntu1) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for libc-bin (2.27-3ubuntu1.3) ...
saurabh@saurabh-VirtualBox:~$
```

Congrats… you installed your first package on Linux !!

… and we have our C compiler here !!

We did install something with the `apt` command

- It succeeded too !!

… and we have our C compiler here !!

We did install something with the `apt` command

- It succeeded too !!

But what did we really install?

… and we have our C compiler here !!

We did install something with the `apt` command

- It succeeded too !!

But what did we really install?

We installed a set of packages that we will start using from next week

… and we have our `C` compiler here !!

We did install something with the `apt` command

- It succeeded too !!

But what did we really install?

We installed a set of packages that we will start using from next week

We installed the compiler for the `C` language – `gcc`

- A compiler is an Application that is capable of changing your instructions to machine instructions
- Basically, it can produce a sequence of 0s and 1s from instructions like “ADD”

… and we have our `C` compiler here !!

We did install something with the `apt` command

- It succeeded too !!

But what did we really install?

We installed a set of packages that we will start using from next week

We installed the compiler for the `C` language – `gcc`

- A compiler is an Application that is capable of changing your instructions to machine instructions
- Basically, it can produce a sequence of 0s and 1s from instructions like “ADD”

We also installed a few more packages that we will need while doing programming in C

… and we have our C compiler here !!

We did install something with the `apt` command

- It succeeded too !!

But what did we really install?

We installed a set of packages that we will start using from next week

We installed the compiler for the C language – `gcc`

- A compiler is an Application that is capable of changing your instructions to machine instructions
- Basically, it can produce a sequence of 0s and 1s from instructions like “ADD”

We also installed a few more packages that we will need while doing programming in C

The command I used on Ubuntu is:

- `sudo apt install build-essential`

… and we have our C compiler here !!

We did install something with the `apt` command

- It succeeded too !!

But what did we really install?

We installed a set of packages that we will start using from next week

We installed the compiler for the C language – `gcc`

- A compiler is an Application that is capable of changing your instructions to machine instructions
- Basically, it can produce a sequence of 0s and 1s from instructions like “ADD”

We also installed a few more packages that we will need while doing programming in C

The command I used on Ubuntu is:

- `sudo apt install build-essential`

Figure it out for your distribution !!

Homework !!

Find out what the “>” did in the examples related to `cat` and `sed`

- “<” also does something, probably the opposite of what “>” does? I don’t know… find out…
- May be giving this a read could be of help: https://www.guru99.com/linux-redirection.html

While you are doing so, read about another single character black magician – “|”

- They call him a “pipe” in the dark world of Linux !!
- These guys may know something about him: https://www.geeksforgeeks.org/piping-in-unix-or-linux/

Practice as much as you can… that’s the only way you can be more comfortable with bash !!

Additional Reading

If you are choosing vim as your editor, welcome to the club ;)

- Start here: https://danielmiessler.com/study/vim/
- If you like watching videos, here's one (search YouTube, there are shorter ones too): https://www.youtube.com/watch?v=a6Q8Na575qc

Another popular alternative to `apt` is `aptitude`

- Check out this tutorial: https://lintut.com/how-to-use-aptitude-on-debian-ubuntu-mint-linux/