Practical No. 1

Aim: Install and configure Linux operating system environment.

Course outcomes: Install Linux Operating System

Theory:

Linux is a community of open-source Unix like operating systems that are based on the Linux Kernel. It was initially released by Linus Torvalds on September 17, 1991. It is a free and open-source operating system and the source code can be modified and distributed to anyone commercially or noncommercially under the GNU General Public License. Initially, Linux was created for personal computers and gradually it was used in other machines like servers, mainframe computers, supercomputers, etc. Nowadays, Linux is also used in embedded systems like routers, automation controls, televisions, digital video recorders, video game consoles, smartwatches, etc. The biggest success of Linux is Android (operating system) it is based on the Linux kernel that is running on smartphones and tablets. Due to android Linux has the largest installed base of all general-purpose operating systems. Linux is generally packaged in a Linux distribution.

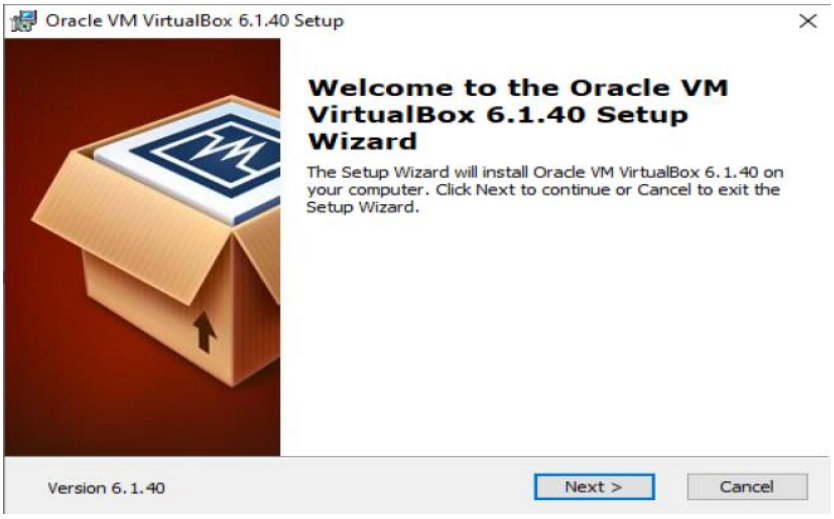
Steps to Download Virtual Box:

Download setup Virtual box using following link as per the version of operating system like windows, Linux OS, etc.

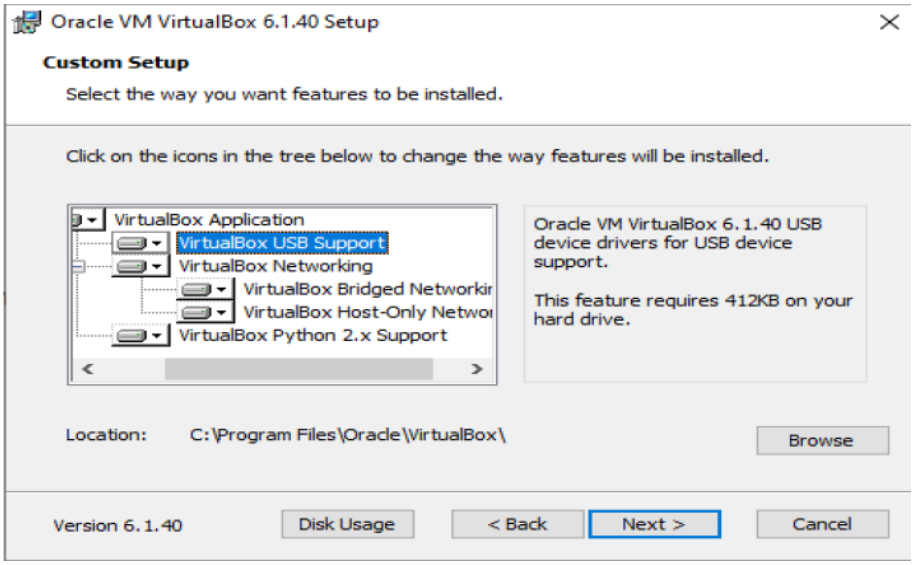
https://www.virtualbox.org/

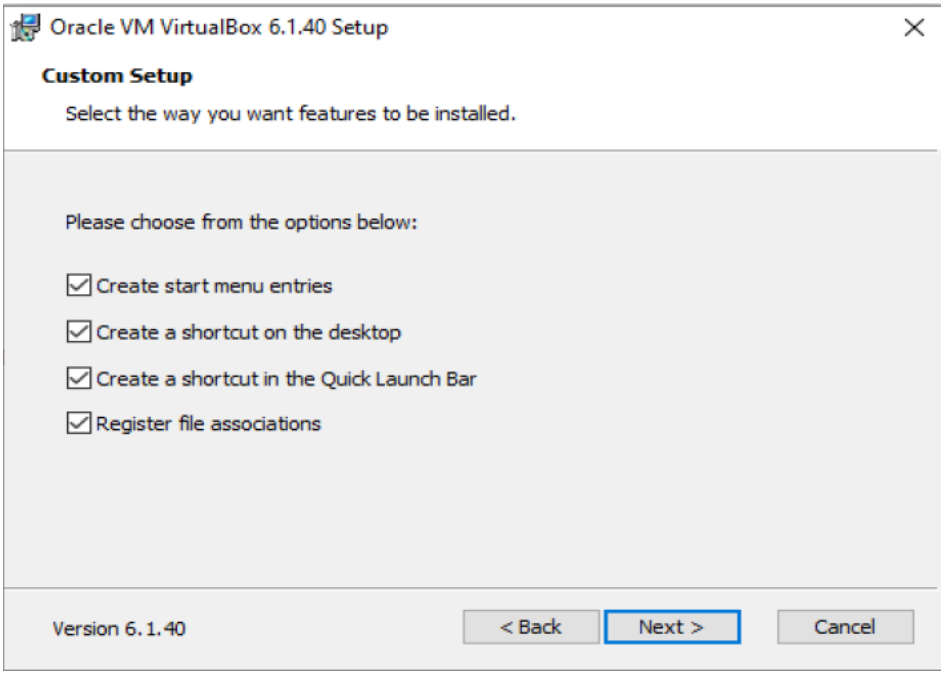
Steps to Install Virtual Box:

Step 1: After clicking on setup it asks for making changes in device so click on YES for start installation.

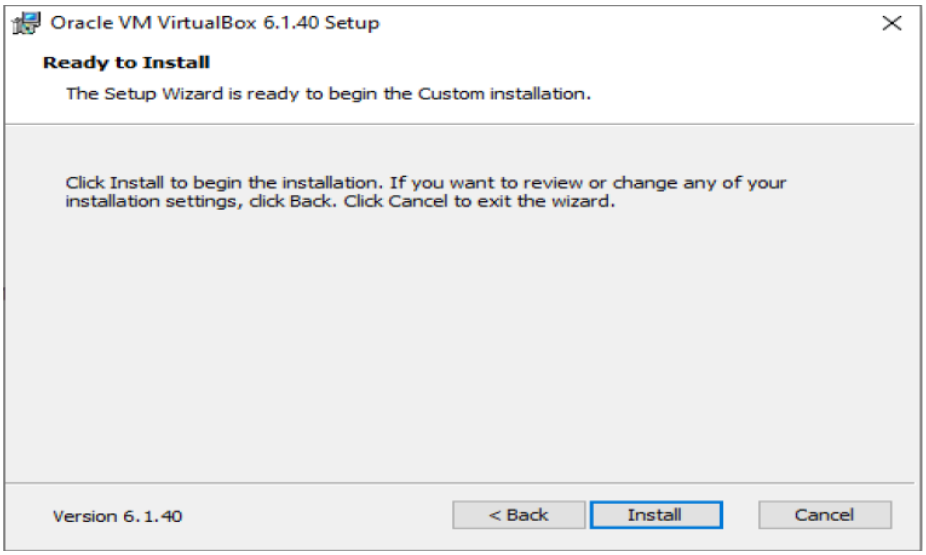


Step 2: Click Next.

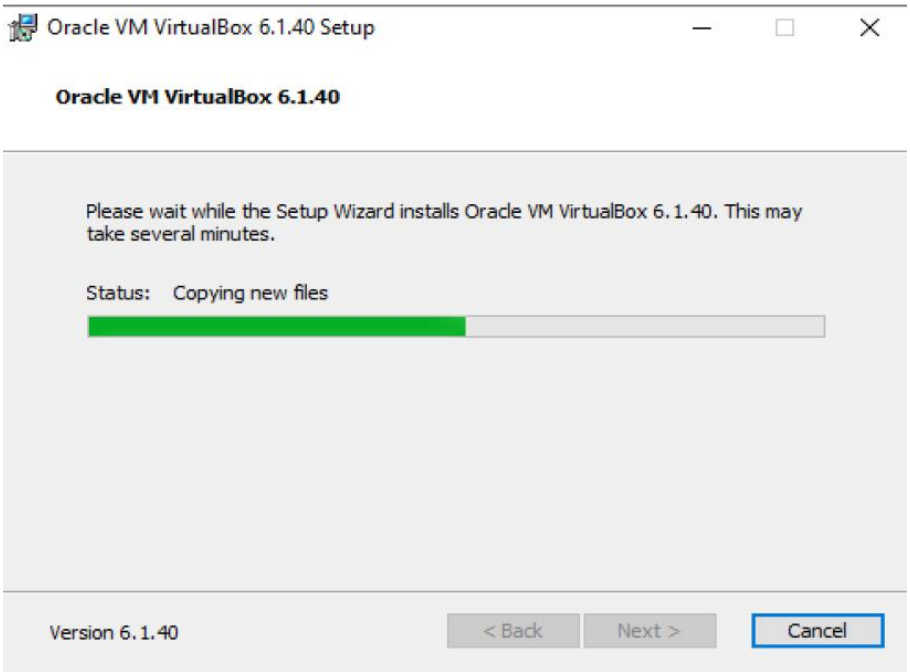




Step 3: Click Install.



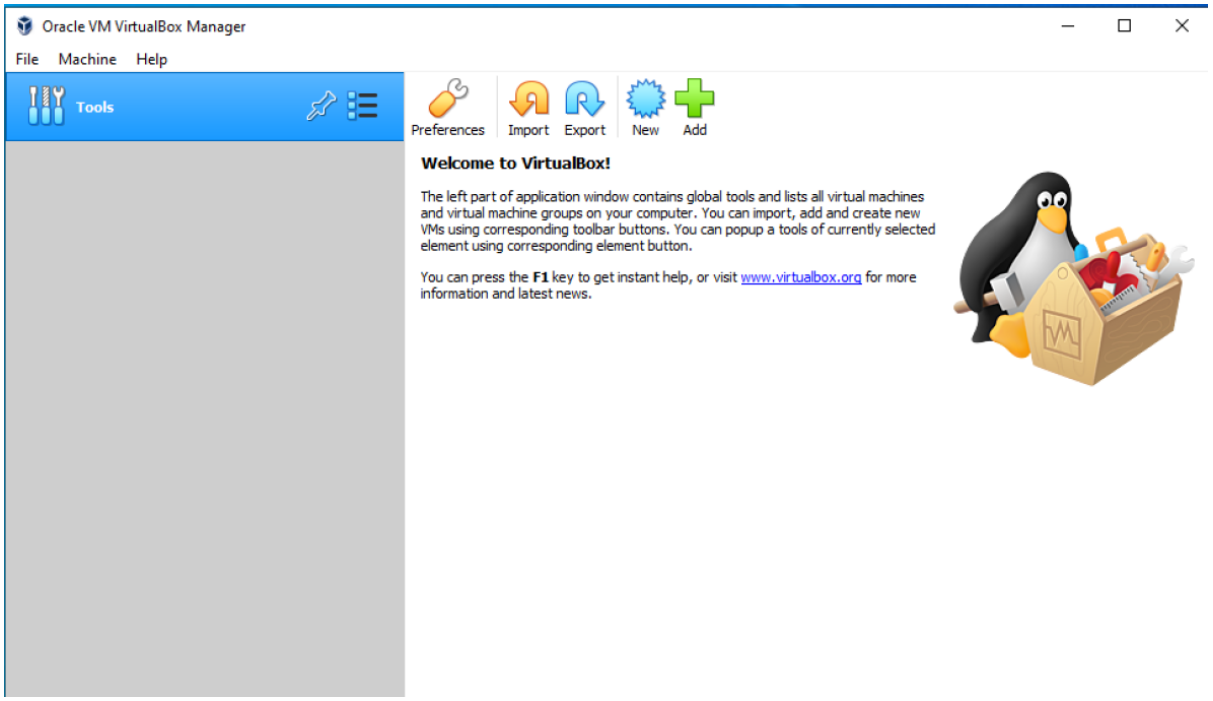
Process of copying files:



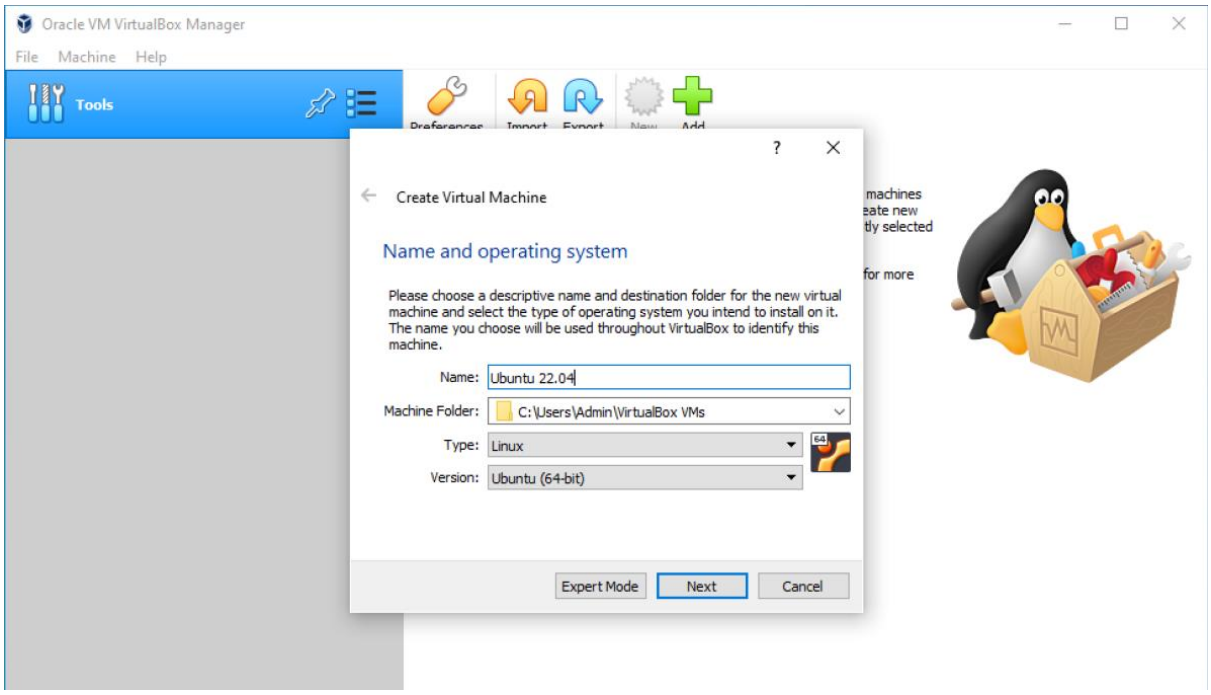
Step 4: Click Finish After completion of installation

Install Linux Operating System using Virtual Box:

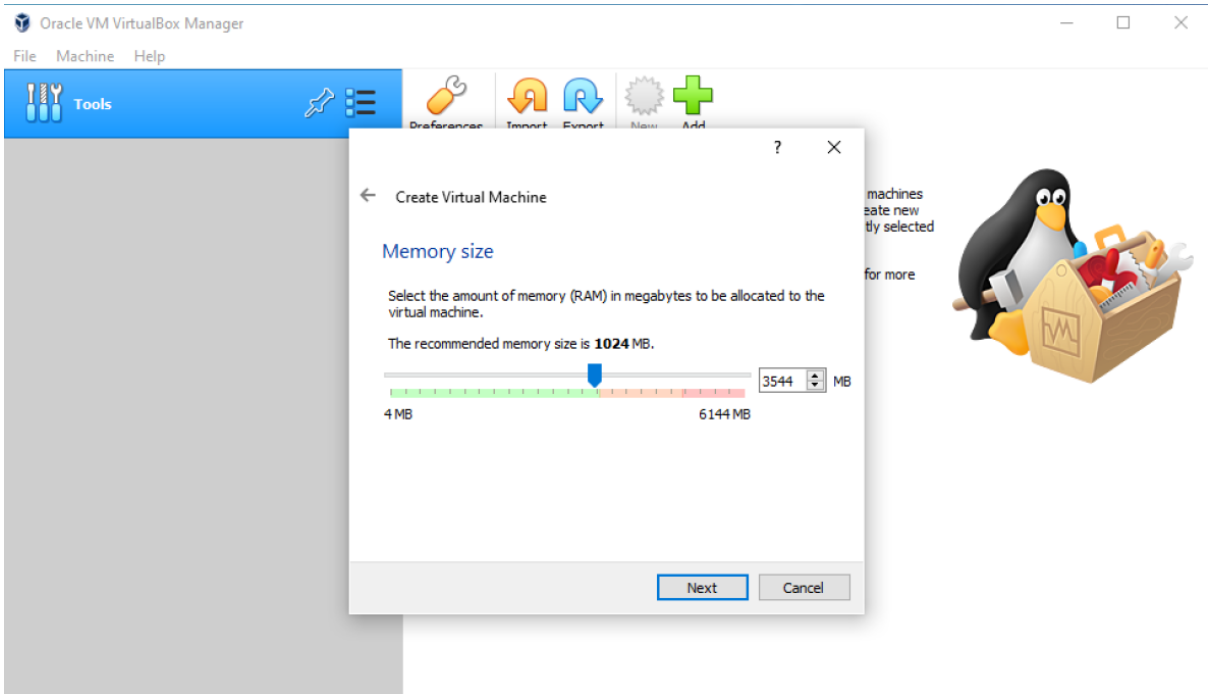
Step 1: Open Virtual box and click on New.



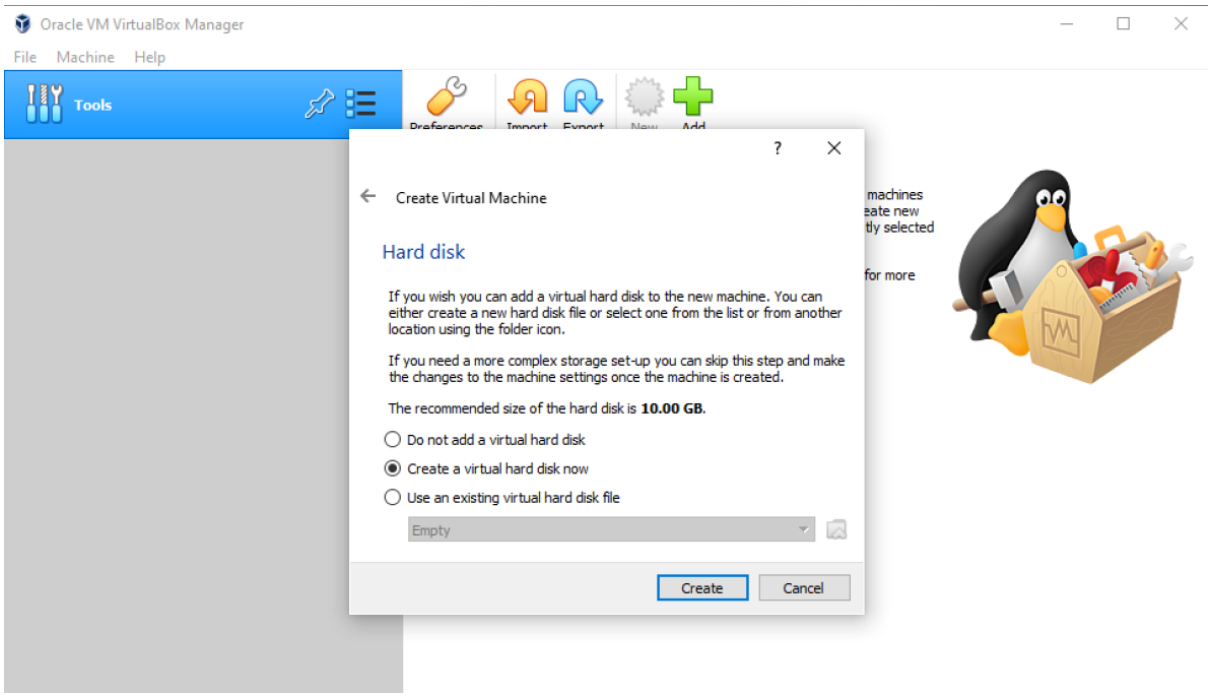
Step 2: Select .iso file of Ubuntu



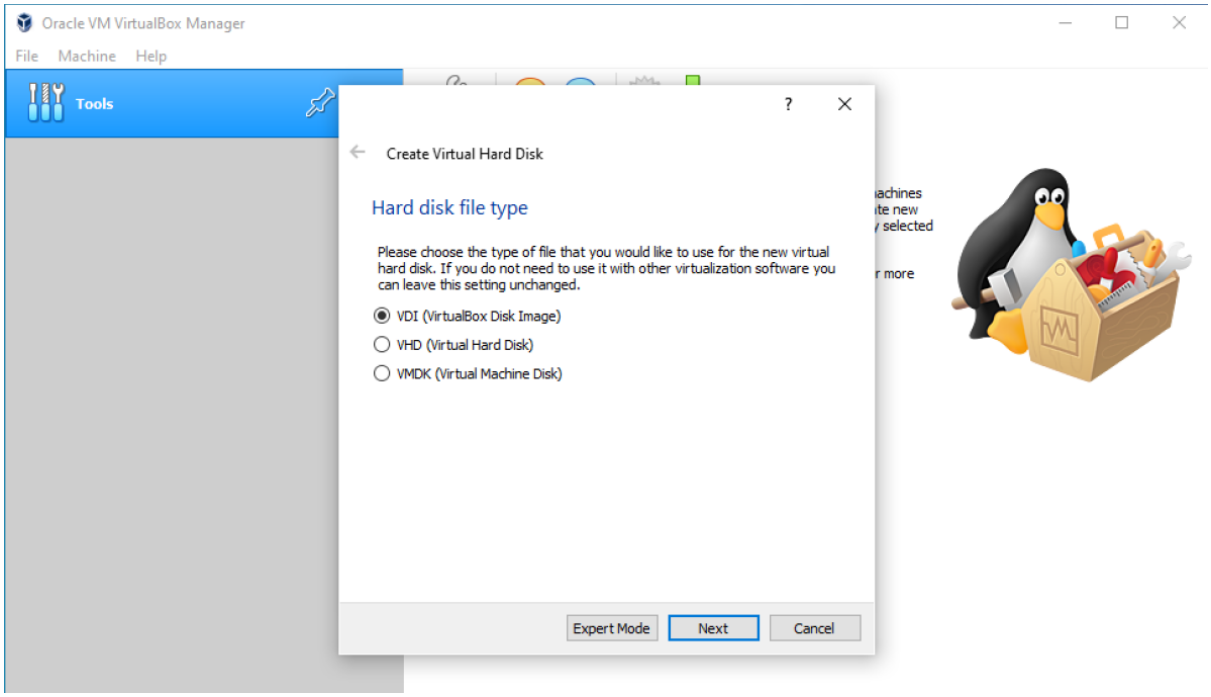
Step 3: Select Memory size

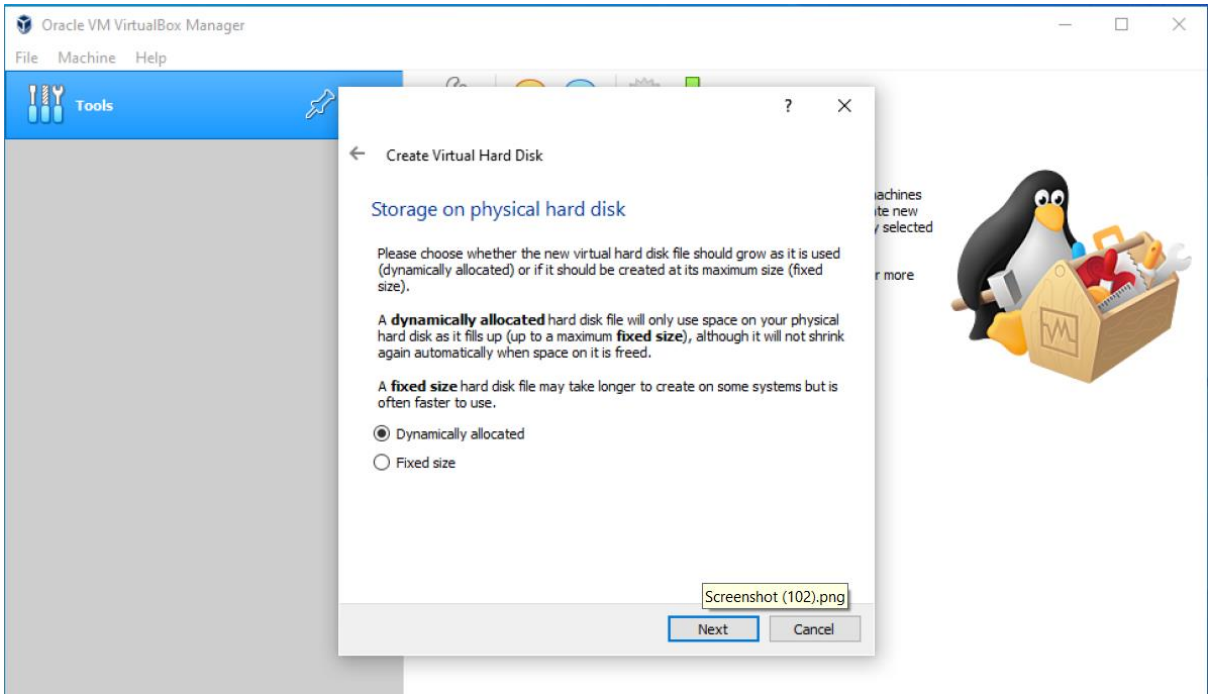


Step 4: Click on create for creating virtual hard disk

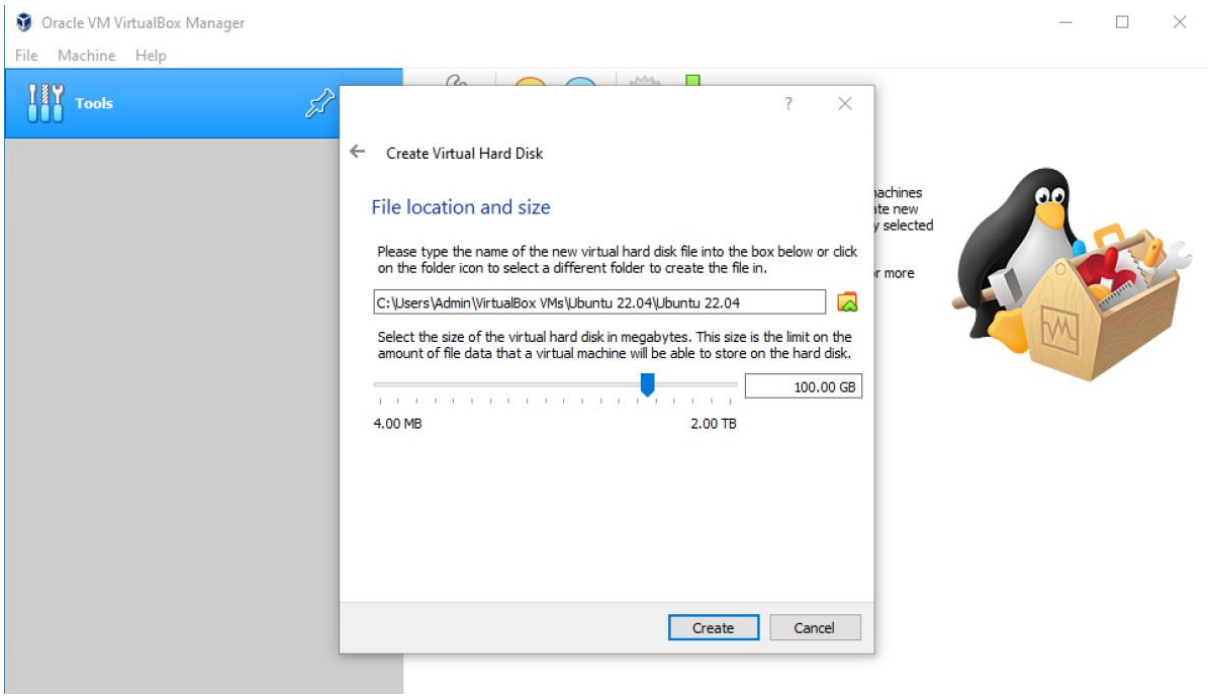


Step 5: Click Next

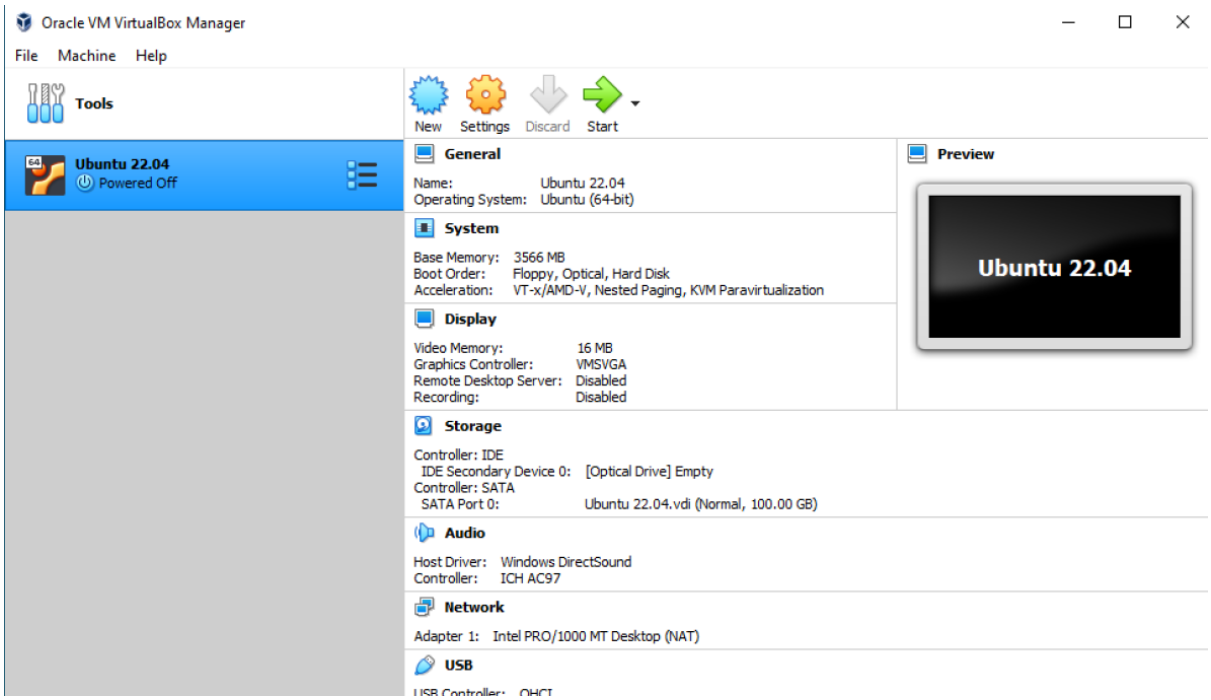




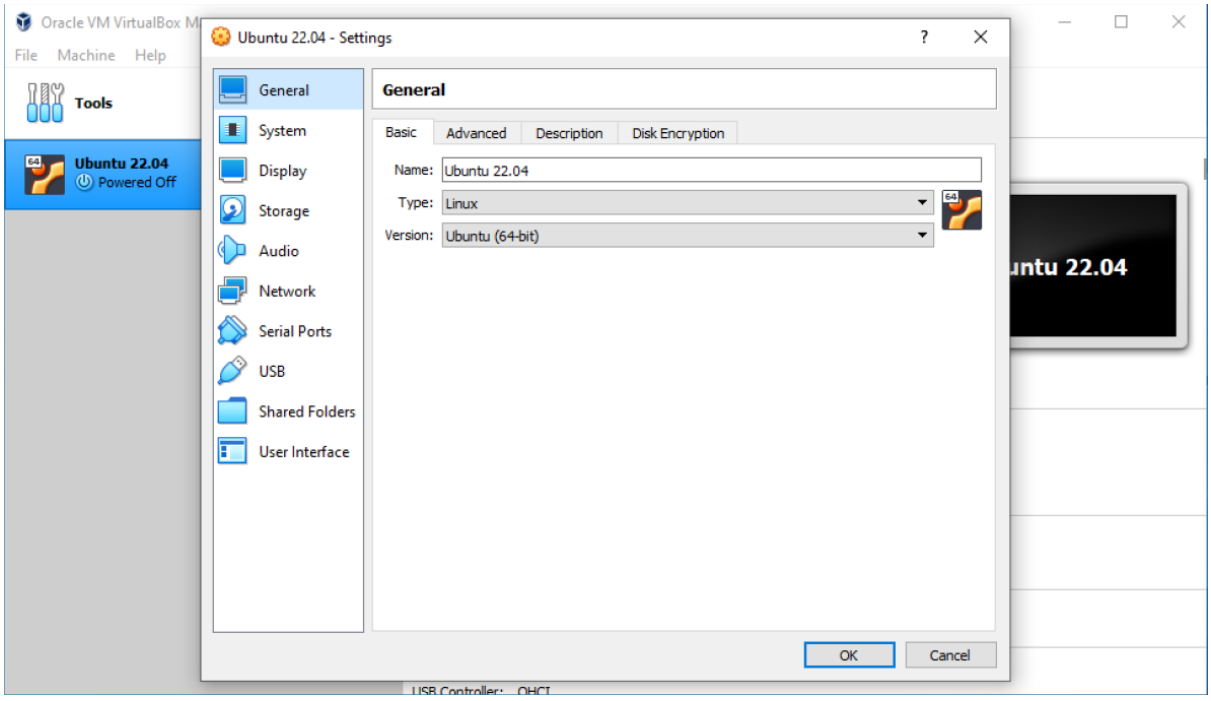
Step 6: Crete File location and size

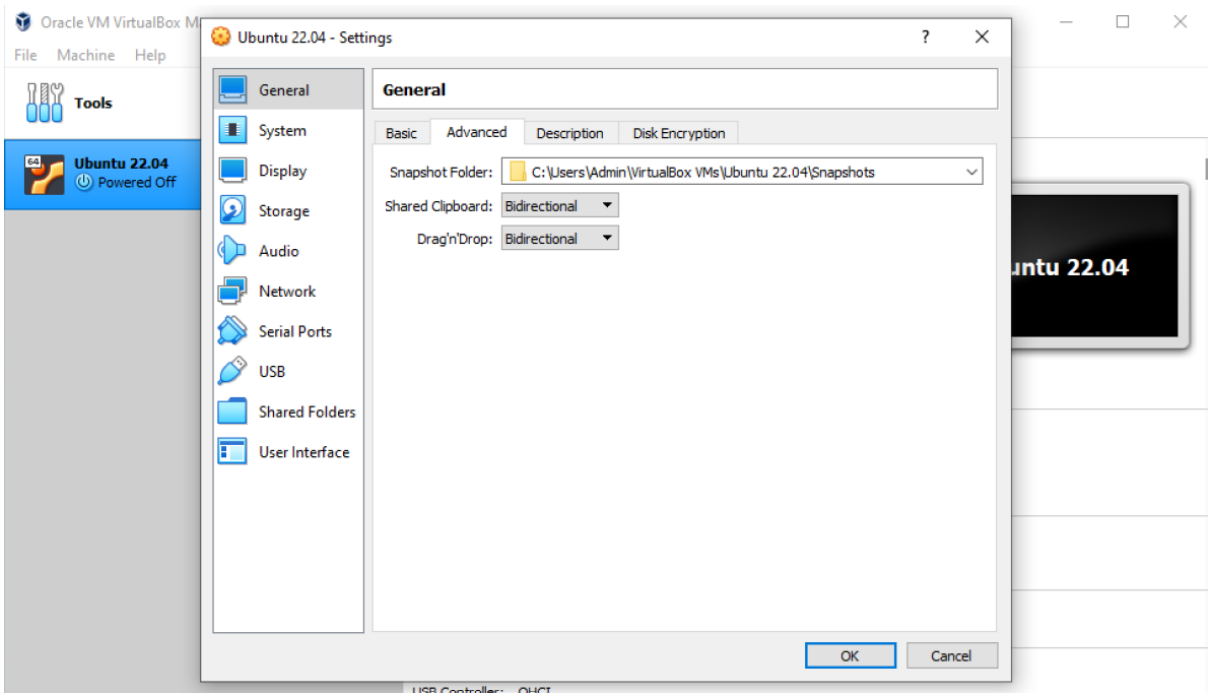


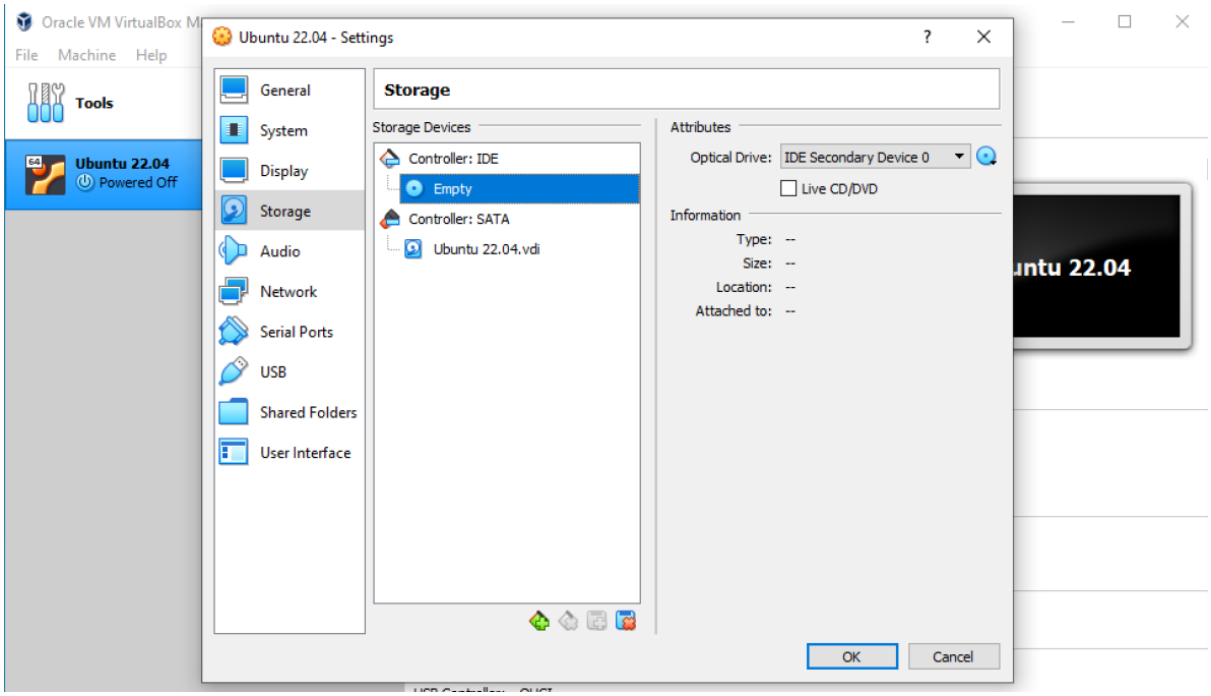
Step 7: Select Ubuntu and click on Start button

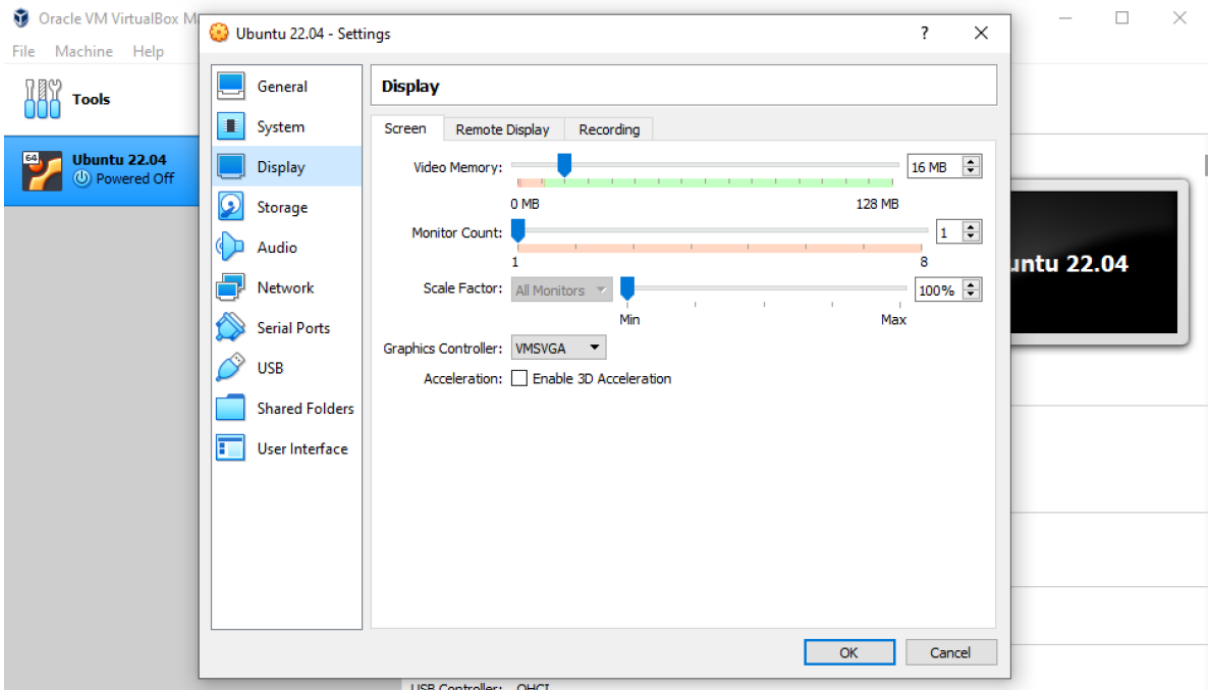


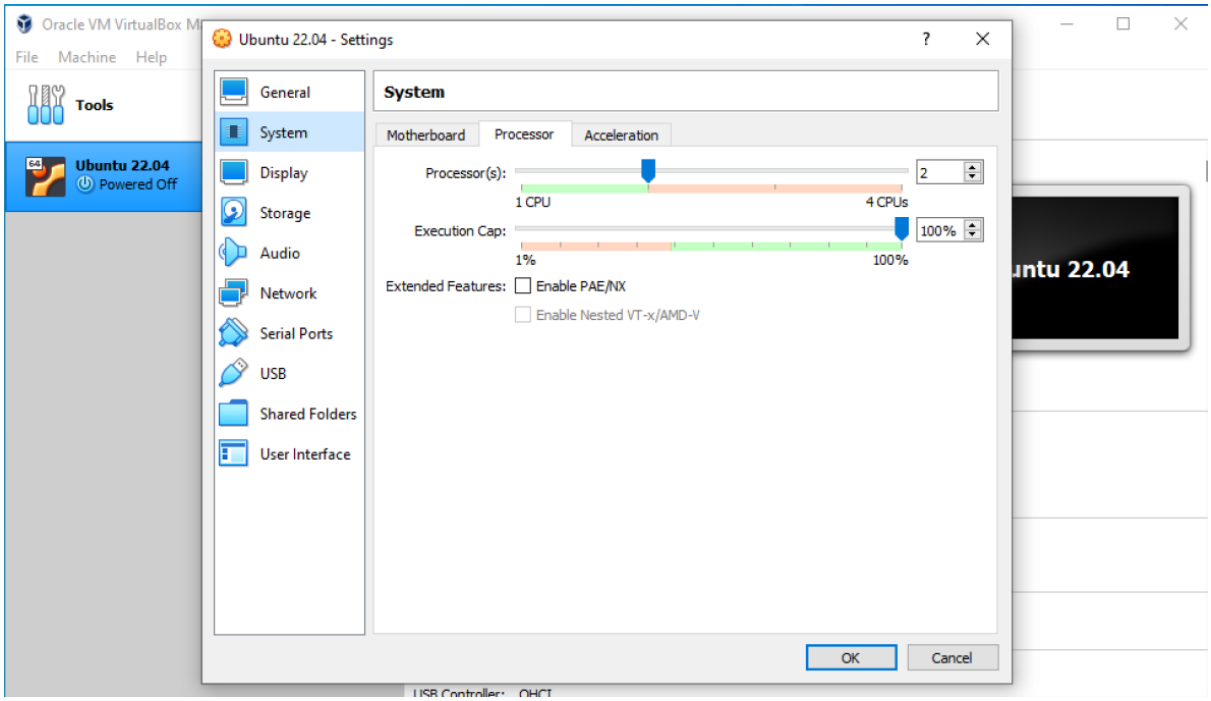
Step 8: Manage General settings, system settings, display, storage, etc.

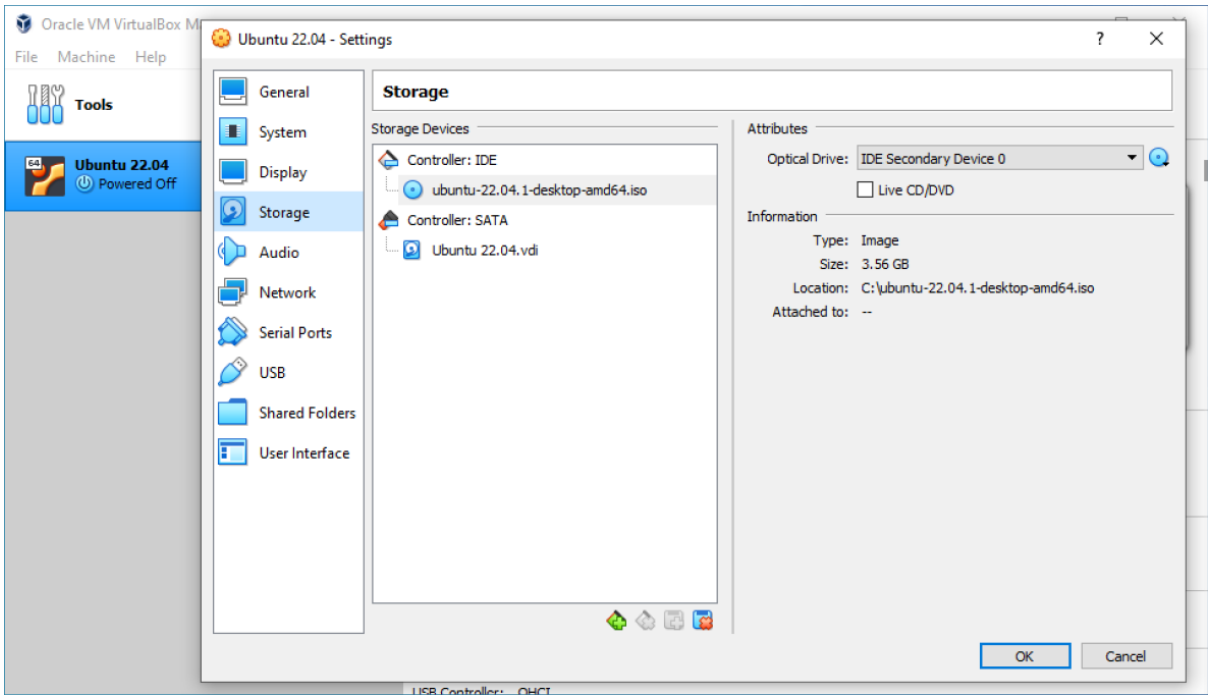



Oracle VM VirtualBox M
File Machine Help
Tools
Ubuntu 22.04
Powered Off
Ubuntu 22.04 - Settings
General
System
Display
Storage
Audio
Network
Serial Ports
USB
Shared Folders
User Interface
General
Basic
Advanced
Description
Disk Encryption
Snapshot Folder:
C:\Users\Admin\VirtualBox VMs\Ubuntu 22.04\Snapshots
Shared Clipboard:
Bidirectional
Drag'n'Drop:
Bidirectional
OK
Cancel

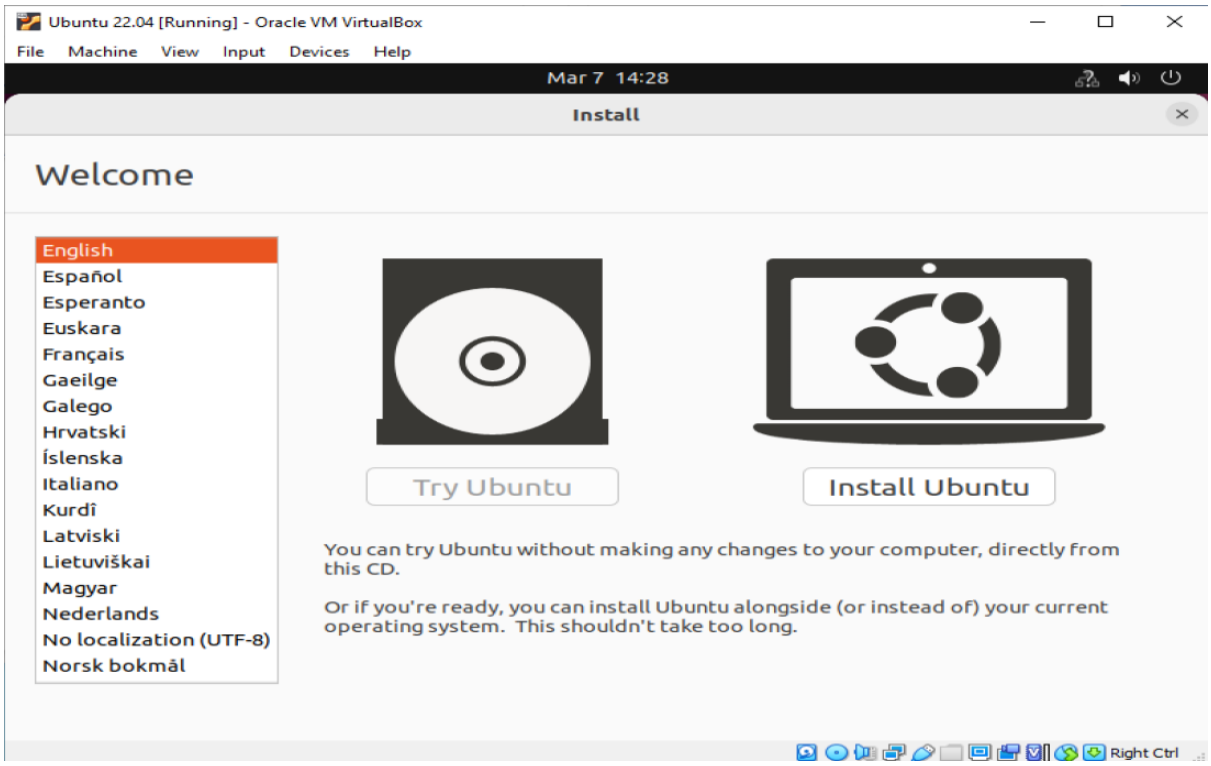

Oracle VM VirtualBox M
File Machine Help
Tools
Ubuntu 22.04
Powered Off
Ubuntu 22.04 - Settings
General
System
Display
Storage
Audio
Network
Serial Ports
USB
Shared Folders
User Interface
Storage
Storage Devices
Controller: IDE
Empty
Controller: SATA
Ubuntu 22.04.vdi
Attributes
Optical Drive:
IDE Secondary Device 0
Live CD/DVD
Information
Type: --
Size: --
Location: --
Attached to: --
OK
Cancel


Oracle VM VirtualBox M
File Machine Help
Tools
Ubuntu 22.04
Powered Off
Ubuntu 22.04 - Settings
General
System
Display
Storage
Audio
Network
Serial Ports
USB
Shared Folders
User Interface
Display
Screen
Remote Display
Recording
Video Memory:
16 MB
0 MB
128 MB
Monitor Count:
1
1
8
Scale Factor:
All Monitors
100%
Min
Max
Graphics Controller:
VMSVGA
Acceleration:
Enable 3D Acceleration
OK
Cancel
untu 22.04

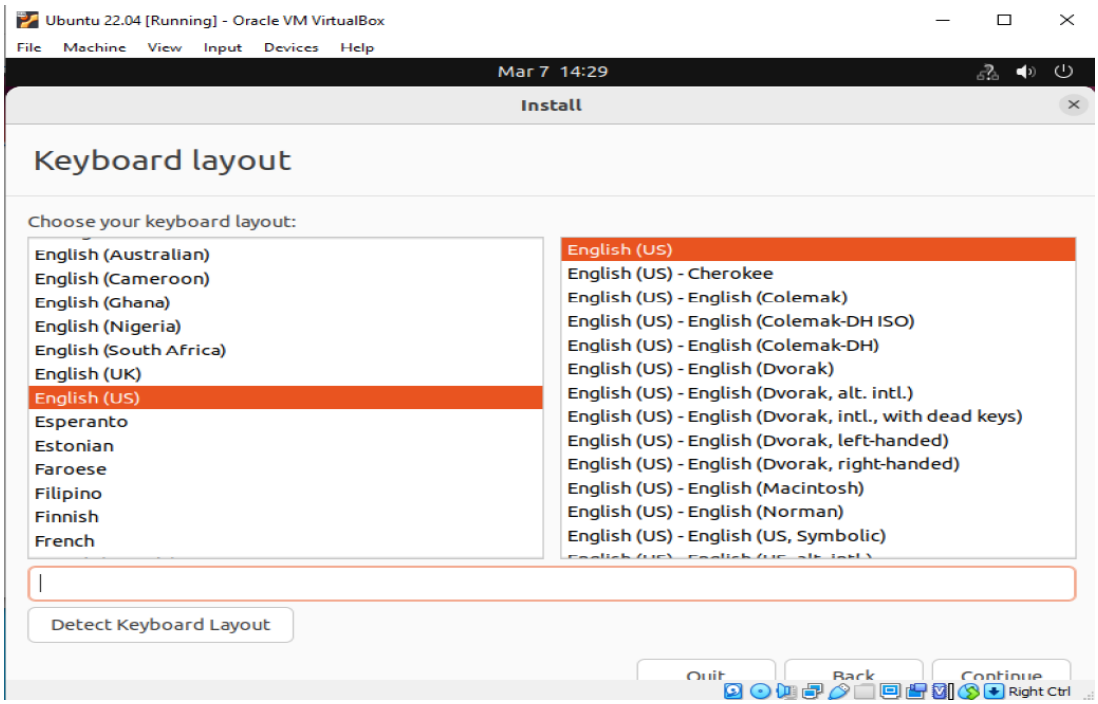

Oracle VM VirtualBox M
File Machine Help
Tools
Ubuntu 22.04
Powered Off
Ubuntu 22.04 - Settings
General
System
Display
Storage
Audio
Network
Serial Ports
USB
Shared Folders
User Interface
System
Motherboard
Processor
Acceleration
Processor(s):
2
1 CPU
4 CPUs
Execution Cap:
100%
1%
100%
Extended Features:
Enable PAE/NX
Enable Nested VT-x/AMD-V
OK
Cancel
untu 22.04



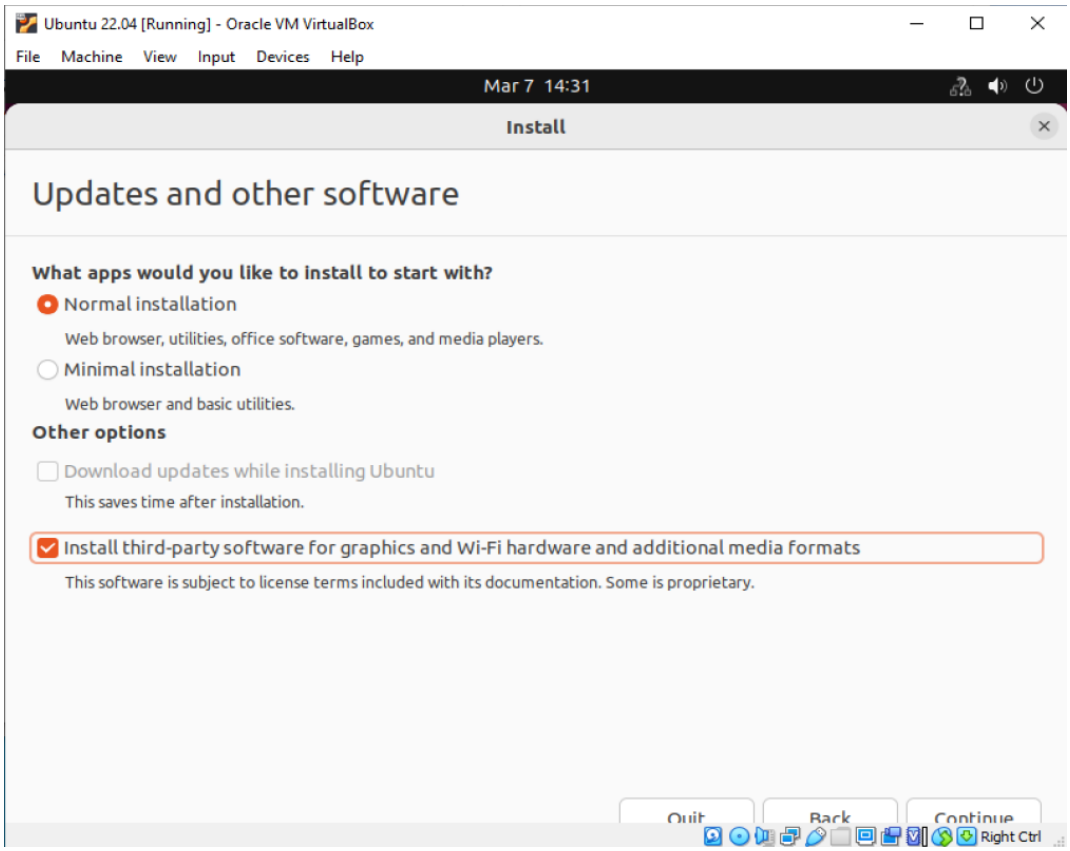
Step 9: After in running state of Ubuntu Select language as English and Click on Install Ubuntu.



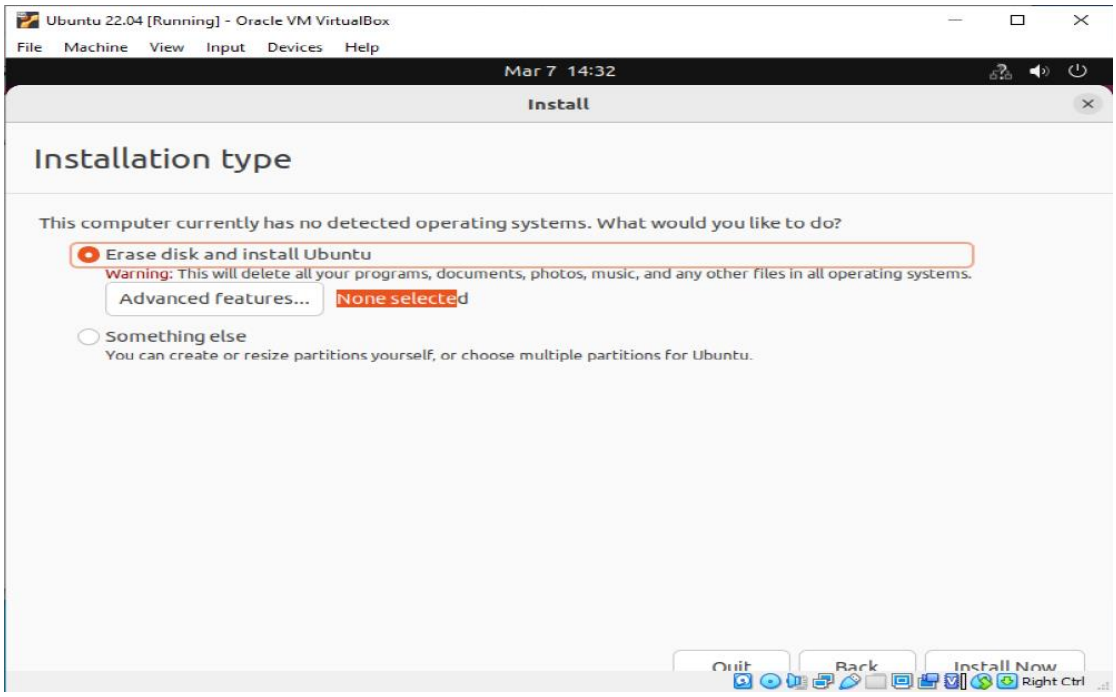
Step 10: Select Keyboard Layout and Continue

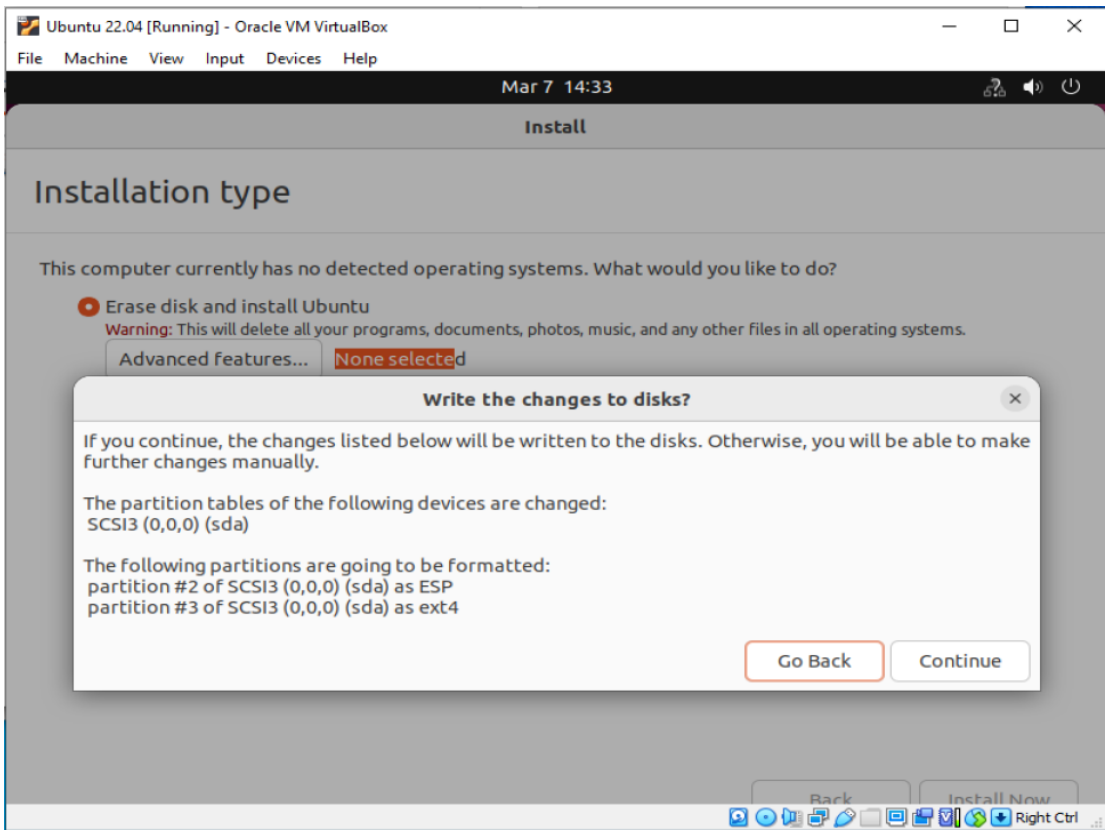


Step 11: Click Next

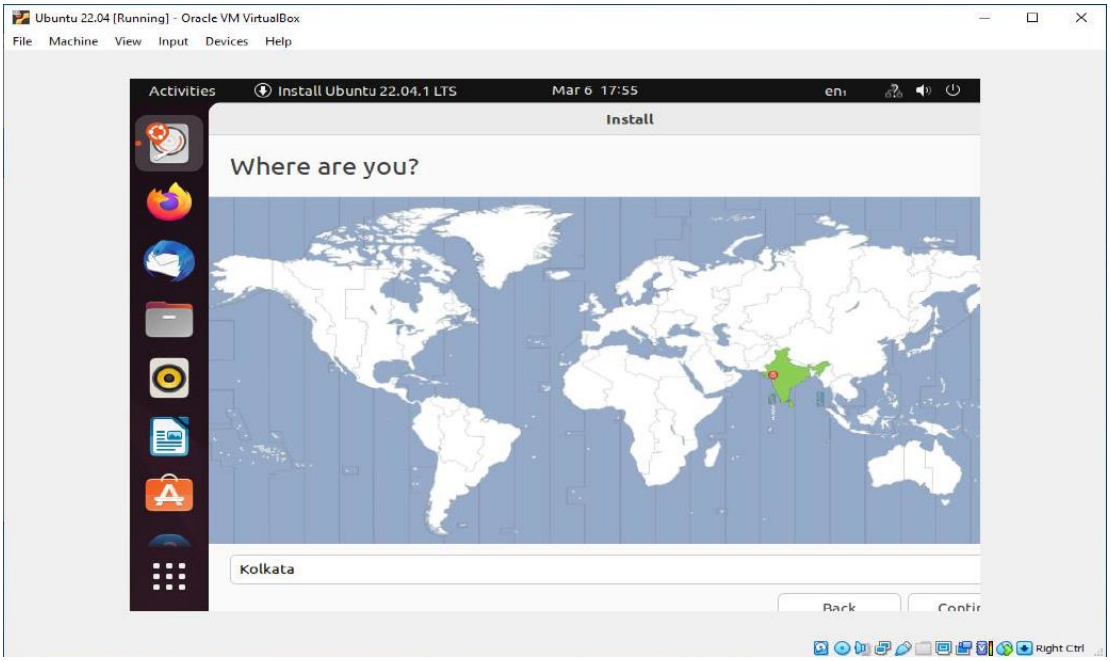


Step 12: Click on Install now

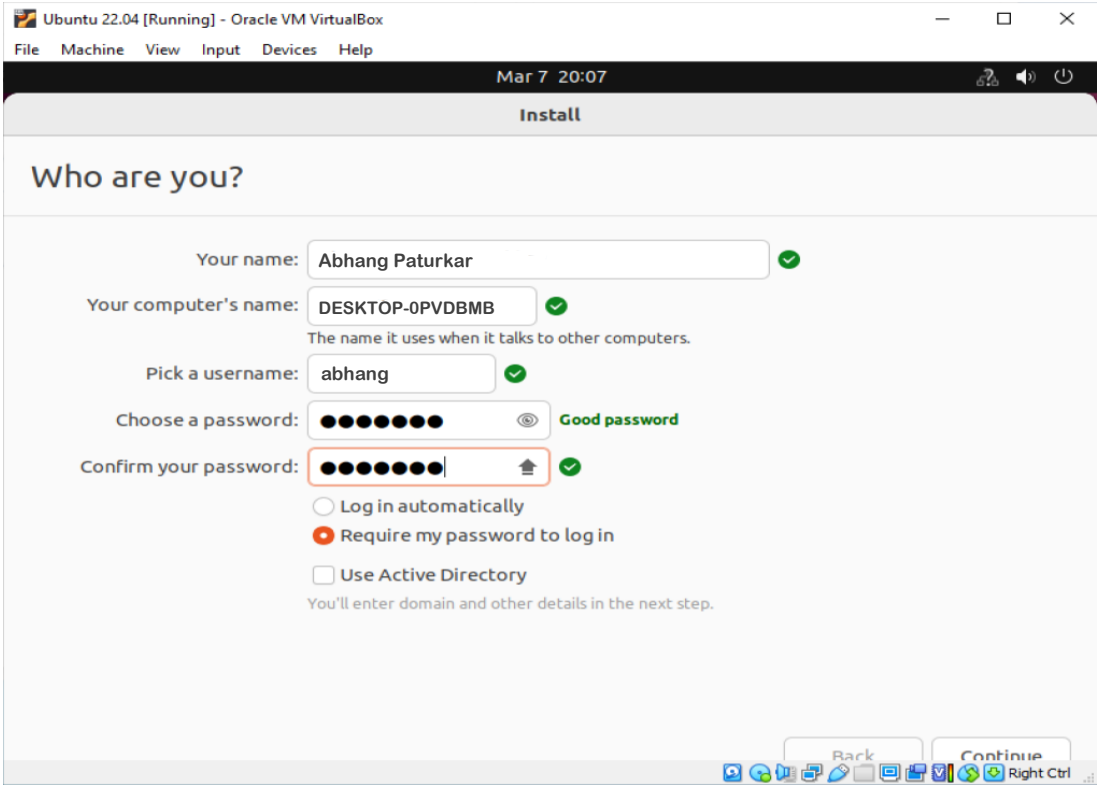


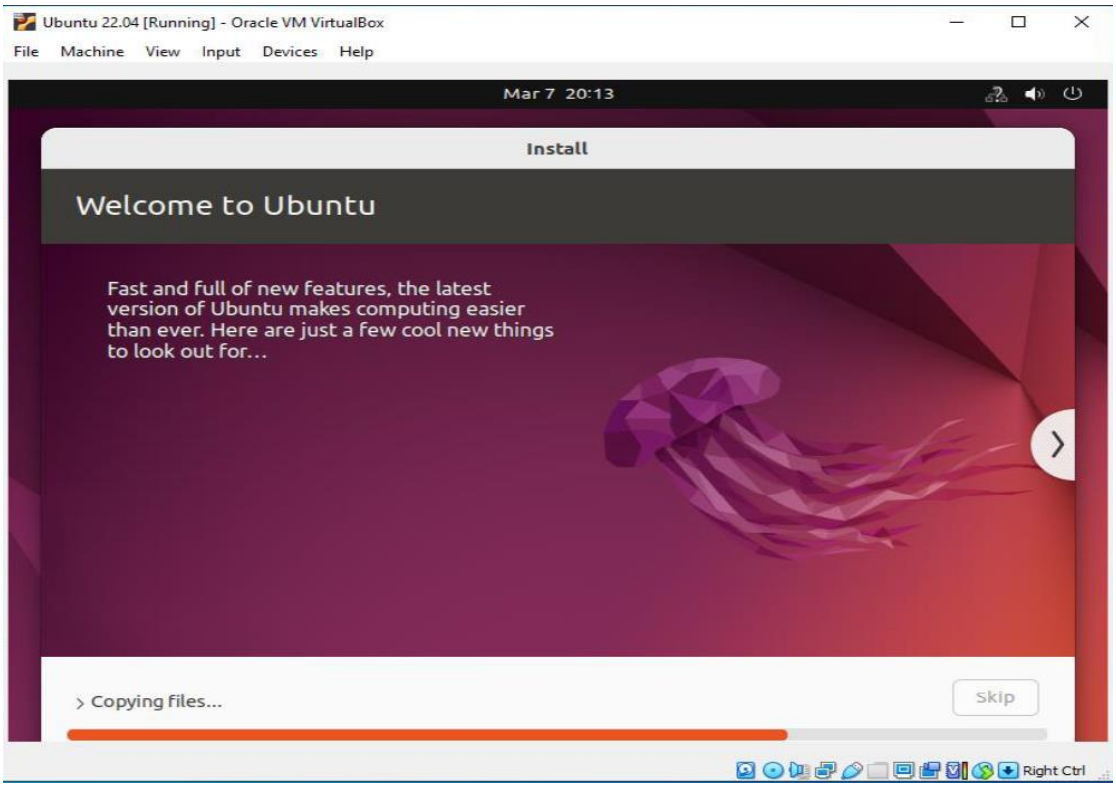


Step 13: Select Location and click Continue

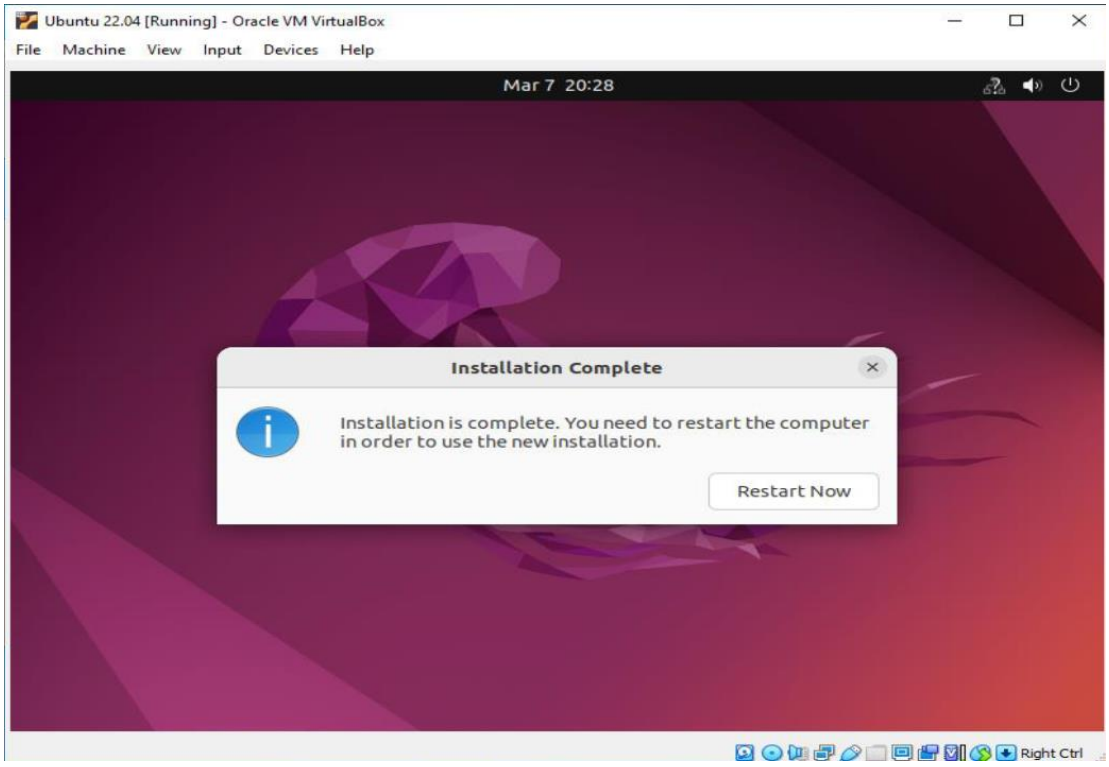


Step 14: Enter Username and Password

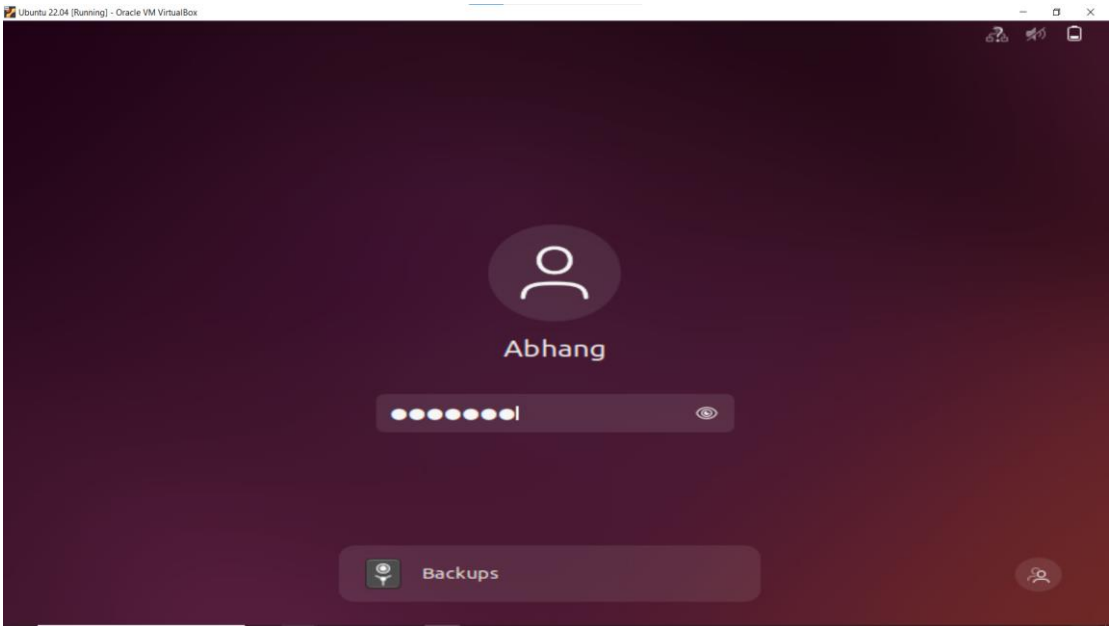




Step 15: After installation Click on Restart now for restarting the system.



Step 16: Enter Password for opening Ubuntu terminal.



Conclusion: In this practical, we have successfully configured and installed Ubuntu operating system in virtual box for performing commands of Linux on it.

Practical No. 2

Aim: Use pipe to concatenate the General-Purpose Linux command.

Course outcomes: Evaluate the basic Linux OS commands and its utilities.

Resource Requirements: Laptop or computer with Linux terminal

Theory:

A pipe is a form of redirection (transfer of standard output to some other destination) that is used in Linux and other Unix-like operating systems to send the output of one command/program/process to another command/program/process for further processing. The Unix/Linux systems allow stdout of a command to be connected to stdin of another command. You can make it do so by using the pipe character **'|'**.
Pipe is used to combine two or more commands, and in this, the output of one command acts as input to another command, and this command's output may act as input to the next command and so on. It can also be visualized as a temporary connection between two or more commands/ programs/ processes. The command line programs that do the further processing are referred to as filters.

Syntax:

command_1 | command_2 | command_3 | .... | command_n

Examples:

1. Sort: The data present in the file is unordered. So, this will sort the given file.

   Syntax : cat file_name | sort

   Command : cat student | sort
   cat subject | sort

2. uniq: Common words in the file display only one time.

   Syntax : cat file_name | sort | uniq

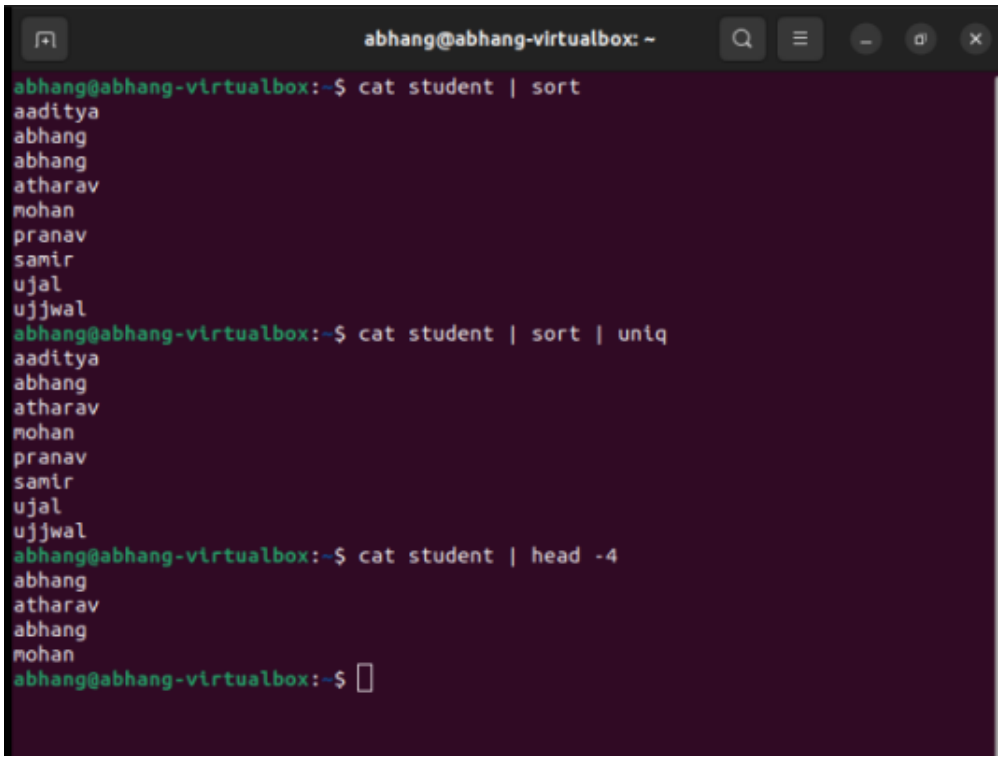
   Command : cat subject | sort| uniq

3. head -4: It prints upper 4 lines in the file

   Syntax : cat file_name | head -4

   Command : cat student | head -4

   Output:



4. grep: This command is used for word searching.

   Syntax : cat file_name | grep <search word>

   Command : cate subject | grep Java

5. tail -5: It prints 5 last lines in the file.

   Syntax : cat file_name | tail -5

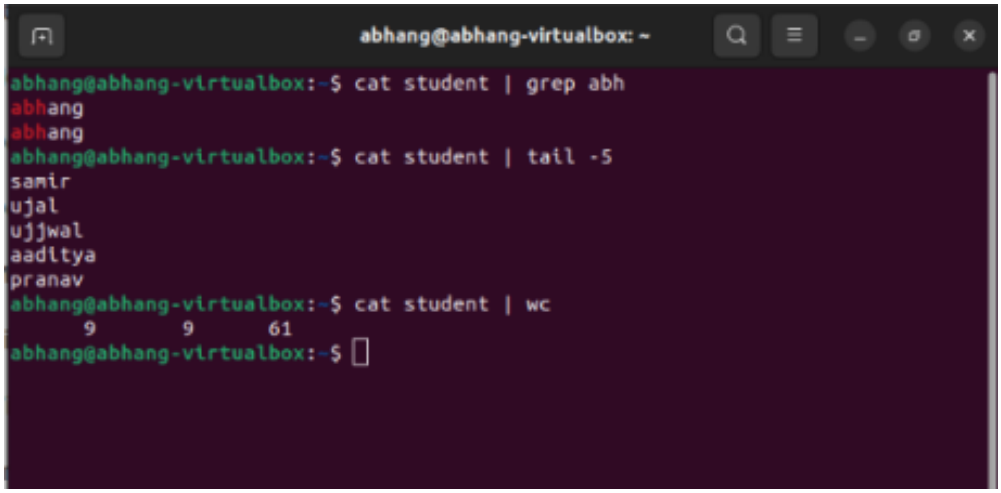
   Command : cat student | tail -5

6. wc: It prints the word count in the file

   Syntax : cat file_name | wc

   Command : cat subject | wc
   cat student | wc

Output:

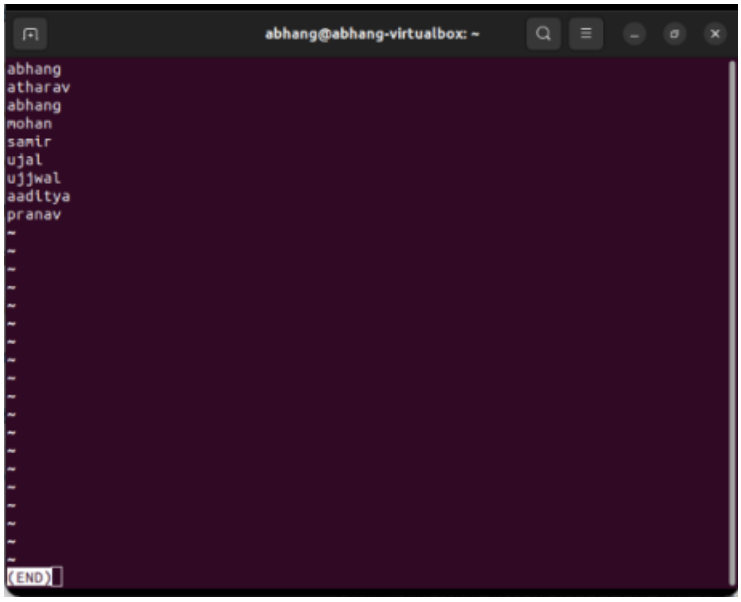


7. less: It helps to scroll the file and exit after pressing ‘q’ when we reach at the end of the file.

   Syntax : cat file_name | less

   Command : cat subject | less

   Output:



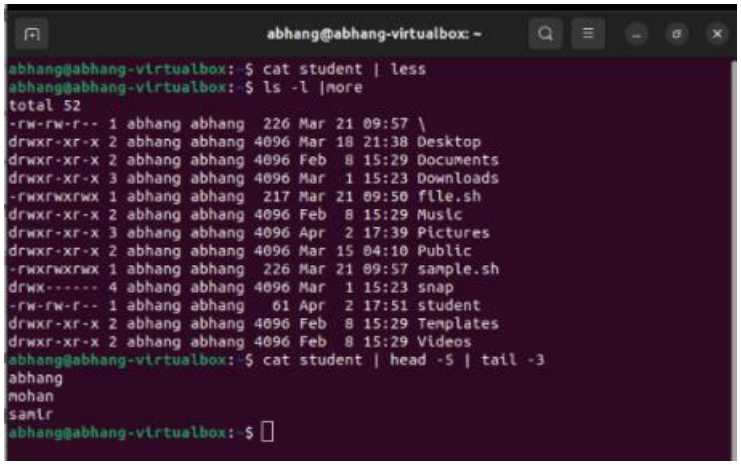
8. -l | more: Listing all files and directories and give it as input to more command.

   Command : $ ls -l | more

9. head | tail: Use head and tail to print in a particular range in a file.

   Command : $ cat Student | head -7 | tail -3

   Output:



10. ls | wc -l: Count of number of files in the /etc directory.

    Command: $ ls | wc -l

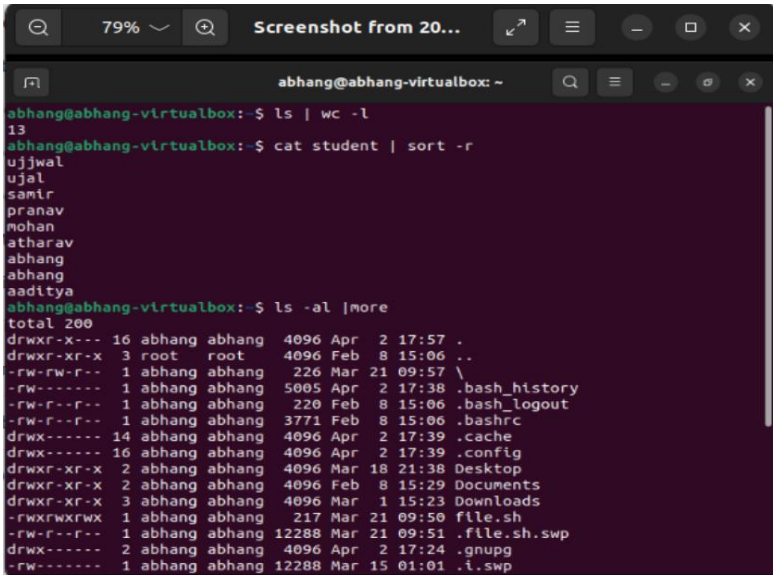
11. Sort -r: Reverse the list present in the given file.

    Command: $ cat Student | sort -r

12. Ls -al | more: List command used to view the files in a directory.

    Command: $ ls -al | more

    Output:



13. grep a: Get specific data from a file which word contains letter ‘a’

    Command: $ cat Student | grep a

14. 16. grep -i s: Get specific data from a file which word contains letter ‘s’ (Not a case sensitive).

    Command : $ cat Student | grep -i s

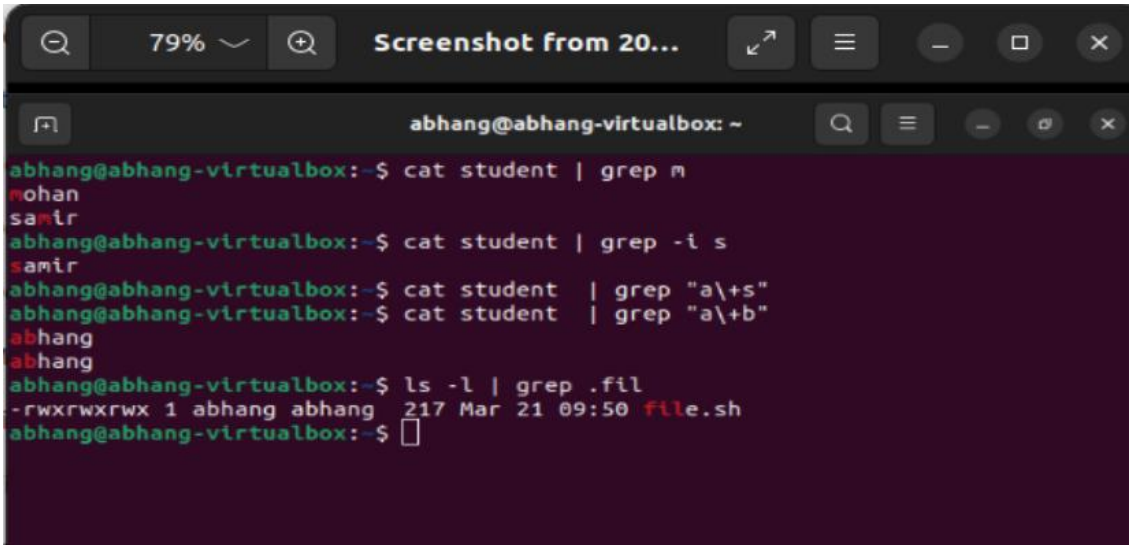
15. grep “a\+s”: Give the words in a file which contains combination of letters ‘a’ and ‘s’.

    Command : $ cate file | grep “a\+s”

16. grep .fil: Display the file with extension .fil

    Command: $ ls -l | grep .fil

    Output:



Conclusion: In this practical, we have successfully performed Linux commands using pipe to concatenate the General-Purpose Linux command.

Practical No. 3

Aim: Use pattern Searching using grep family commands.

Course outcomes: Evaluate the basic Linux OS commands and its utilities.

Resource requirements: Laptop or computer with Linux terminal

Theory:

The grep filter searches a file for a particular pattern of characters, and displays all lines that contain that pattern. The pattern that is searched in the file is referred to as the regular expression (grep stands for global search for regular expression and print out).

Syntax:

grep [options] pattern [files]

1. grep <search word>: This command is used to searching word in file and print it.

   Syntax : cat file_name | grep <search word>

   Command : cat student | grep Sofia

2. grep -i: Check for same words and letters ignore case sensitivity.

   Syntax : cat file_name | grep -i <search word>

   Command : cat subject | grep -i java

   cat student | grep -i a

3. grep -v: This prints all the lines that do not match the pattern.

   Syntax : cat file_name | grep -v <pattern>

   Command : cat subject | grep -v Java

4. grep -c: This prints only a count of the lines that match with the pattern.

   Syntax : cat file_name | grep -c <pattern>

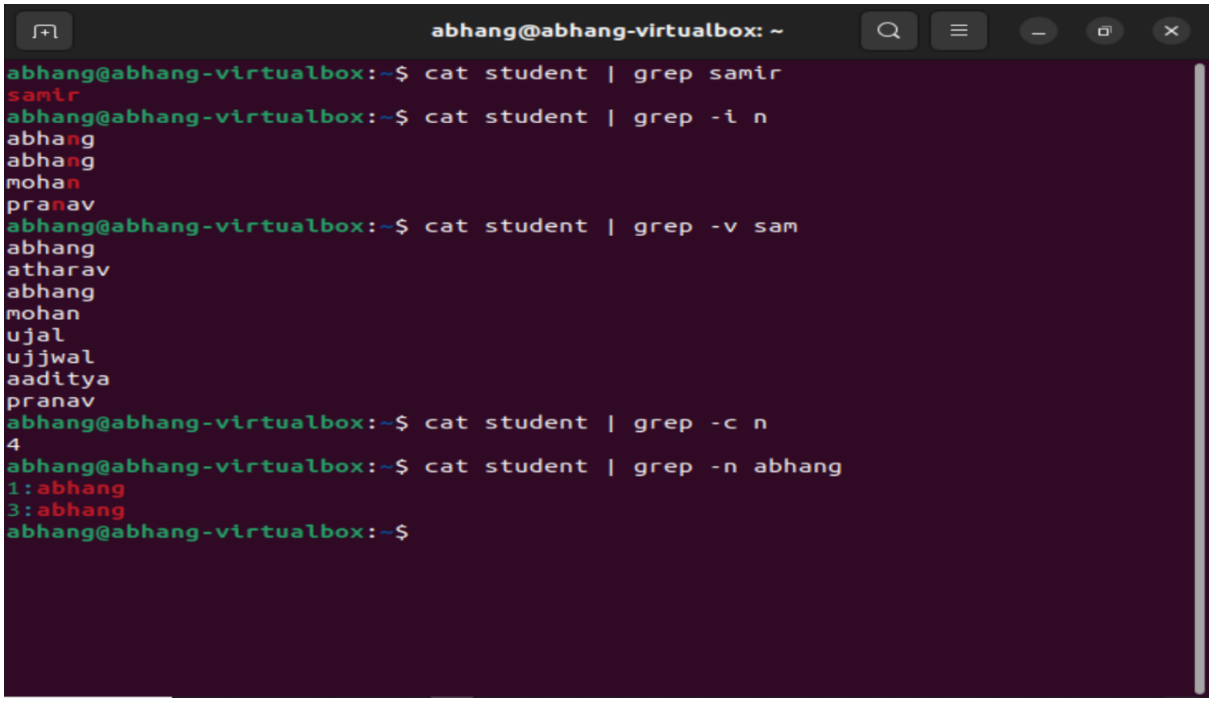
   Command : cat subject | grep -c Java

5. grep -n: Display the matched lines and their line numbers.

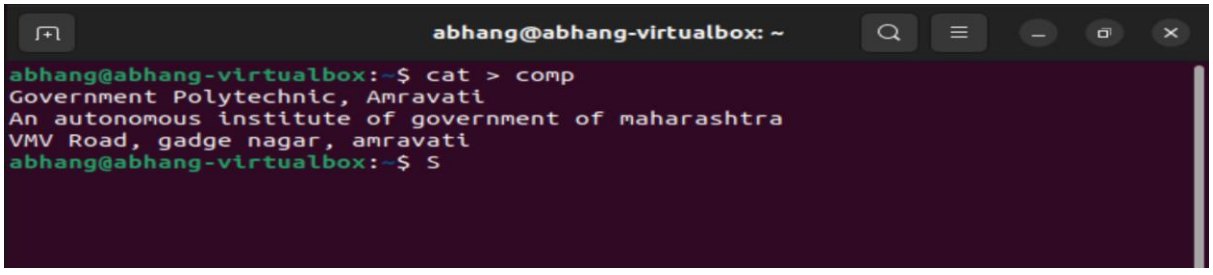
   Syntax : cat file_name | grep -n <pattern>

   Command : cat subject | grep -n Python

   Output:



New file creation by using cat command file name is “comp”.



6. grep -A1: Command is used to display the line after the result.
7. grep -B1: Command is used to display the line before the result.
8. grep -C1: Command is used to display the line after and line before the result.

Syntax : cat file_name | grep -A1 <pattern>

cat file_name | grep -B1 <pattern>

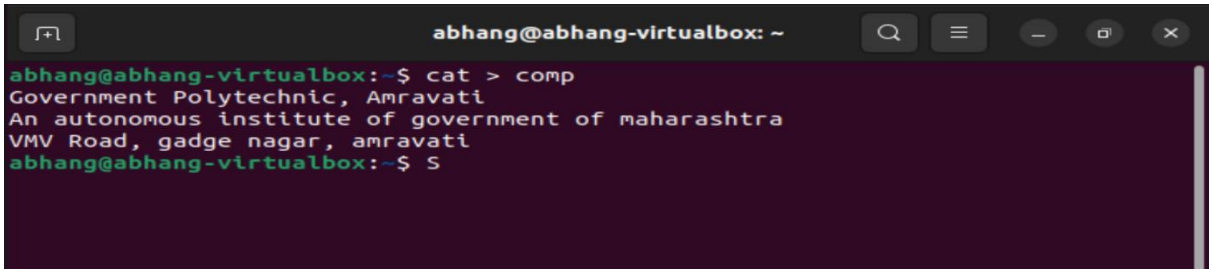
cat file_name | grep -C1 <pattern>

Commands : cat info | grep -A1 red

cat info | grep -B1 red

cat info | grep -C1 red

Output:



9. grep -o: Displaying only the matched pattern.

   Command : $ grep -o “Linux” Subject

10. grep -w: Checking for the whole words in a file.

    Command : $ grep -w “Linux” Subject

11. grep “^pattern”: Matching the lines that start with a string/pattern.

    Command : $ grep “^Java” Subject

12. grep “pattern$”: Matching the lines that end with a string.

    Command : $ grep “Python$” Subject

13. grep -r: Search recursively for a pattern in the directory -r.

    Command : $ grep -r “Linux” *

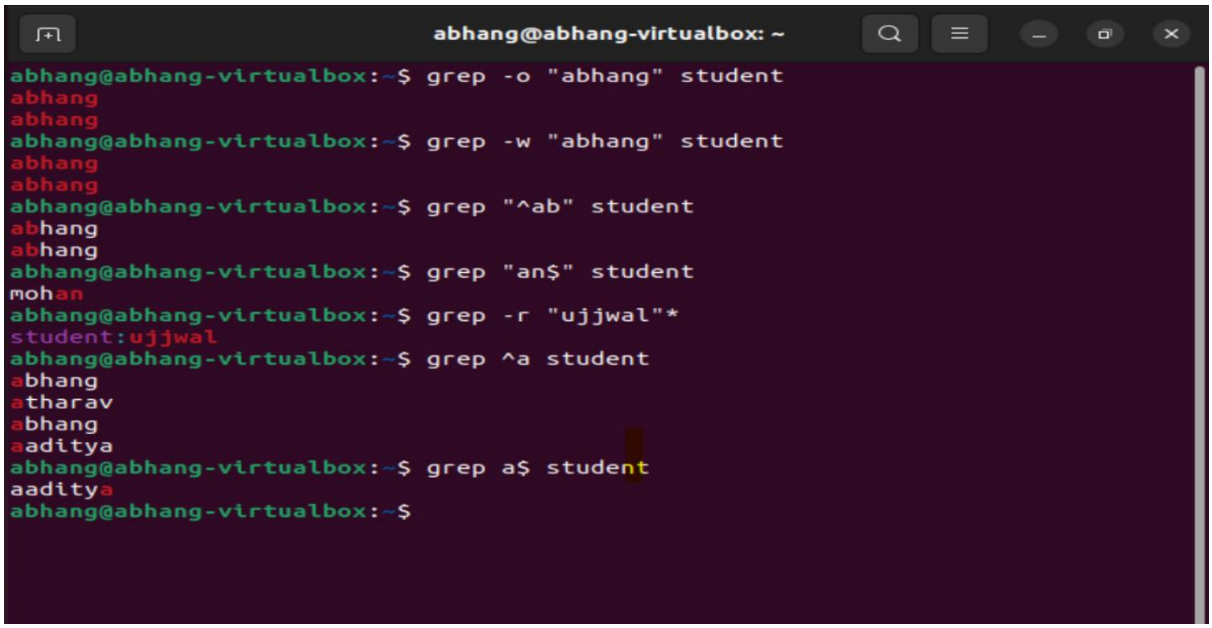
14. grep ^J: Matching the words that start with a letter.

    Command : $ grep ^J Student

15. grep s$: Matching the words that end with a letter.

    Command : $ grep s$ Student\

Output :



Conclusion: In this practical, we have successfully performed Linux commands to demonstrate the use of pattern Searching using grep family commands.

Practical No. 4

Aim: Write a Shell script using following Control Structures:
a) if then else structure and nested if then Structure.
b) Case Statement

Course Outcome: Develop shell program for solving different problems.

Resource requirement: Laptop or Computer (with Linux terminal),basic knowledge of conditional statements in Linux.

Theory:

1) if..else..fi

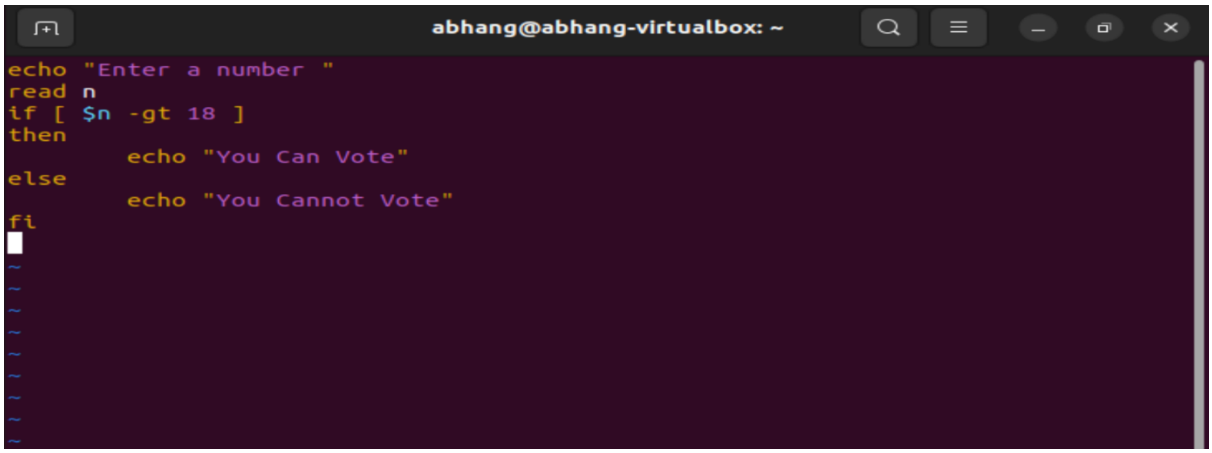
The **if...else...fi** statement is the next form of control statement that allows Shell to execute statements in a controlled way and make the right choice.

Syntax :

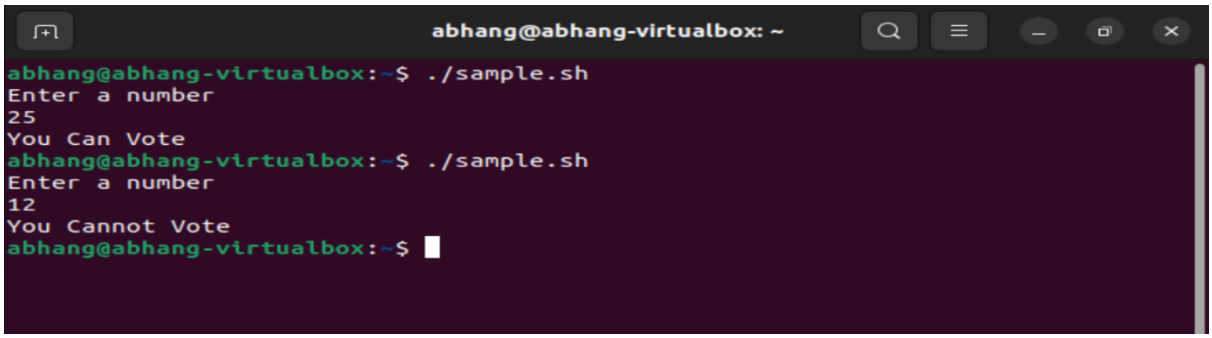
if [expression]
then
Statement(s) to be executed if expression is true
else
Statement(s) to be executed if expression is not true
fi

The Shell *expression* is evaluated in the above syntax. If the resulting value is *true*, given *statement(s)* are executed. If the *expression* is *false*, then no statement will be executed.

Program :



Output :



2) If…elif…fi

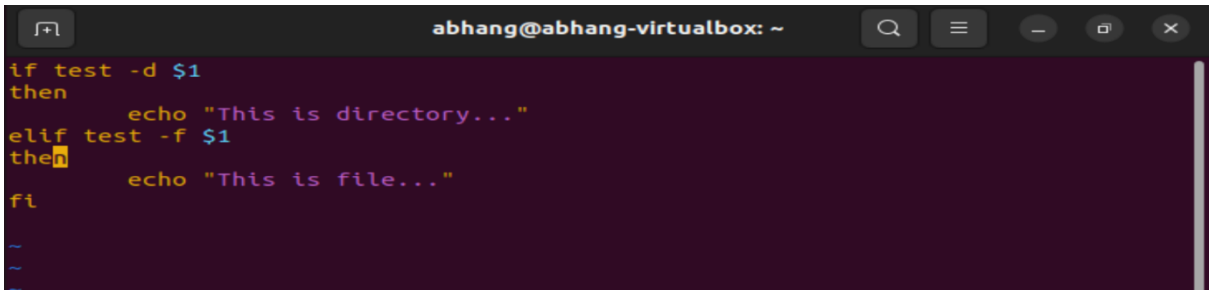
The **if...elif...fi** statement is the one level advance form of control statement that allows Shell to make correct decision out of several conditions.

Syntax :

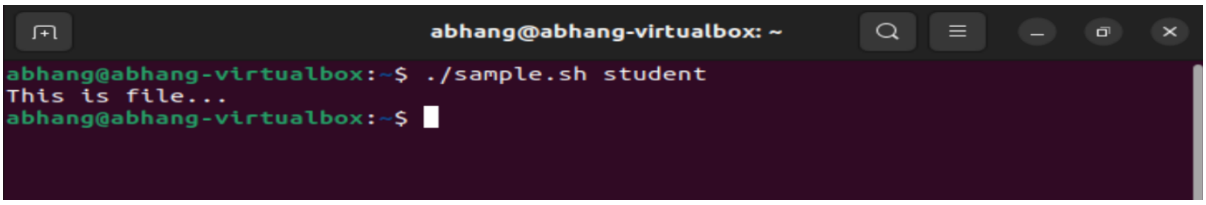
if [expression 1]
then
Statement(s) to be executed if expression 1 is true
elif [expression 2]
then
Statement(s) to be executed if expression 2 is true
elif [expression 3]
then
Statement(s) to be executed if expression 3 is true
else
Statement(s) to be executed if no expression is true
Fi

This code is just a series of *if* statements, where each *if* is part of the *else* clause of the previous statement. Here statement(s) are executed based on the true condition, if none of the condition is true then *else* block is executed.

Program :



Output :



3) if..then..else..if..then..fi..fi..(Nested if)

Nested if-else block can be used when, one condition is satisfies then it again checks another condition. In the syntax, if expression1 is false then it processes else part, and again expression2 will be check.

Syntax :

if [expression1]
then
statement1
statement2
else
if [expression2]
then
statement3
fi
fi

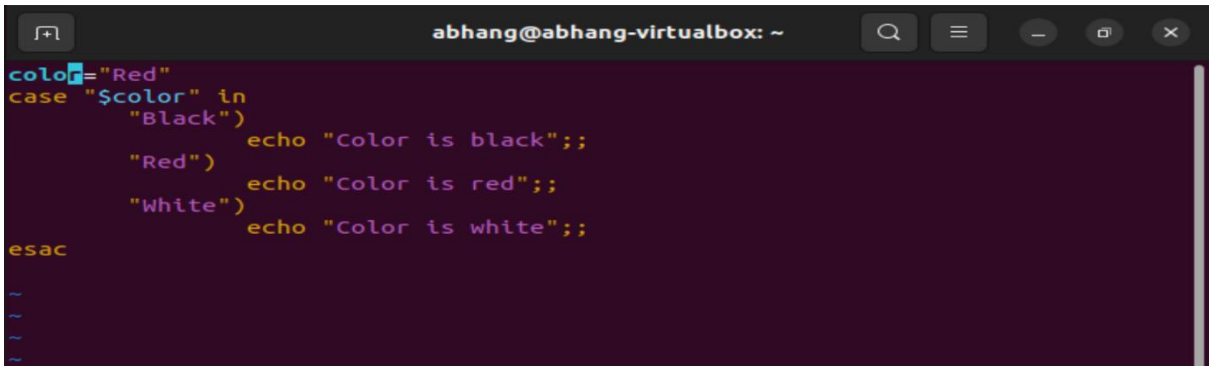
4) switch statement:

Case statement works as a switch statement if specified value match with the pattern then it will execute a block of that particular pattern When a match is found all of the associated statements until the double semicolon (;;) is executed. A case will be terminated when the last command is executed. If there is no match, the exit status of the case is zero.

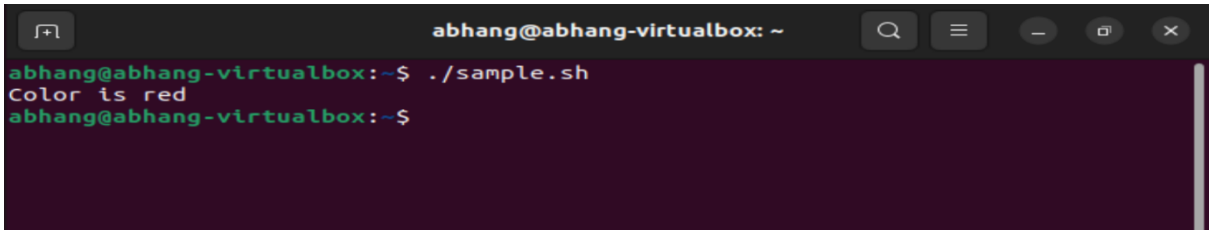
Syntax :

case in
Pattern 1) Statement 1;;
Pattern n) Statement n;;
Esac

Program :



Output:



Conclusion: In this Practical, we have successfully executed shell scripts using different looping structures like if- then- else, nested if, ladder structure and using case to test a variable for multiple values in vi editor.

Practical No. 5

Aim: Write a shell script using expr to perform arithmetic operations.

Course Outcome: Develop shell program for solving different problems.

Resource requirement: Laptop or Computer (with Linux terminal), basic knowledge of conditional statements in Linux.

Theory:

Arithmetic Operators:

The arithmetic operators are used to perform mathematical operations.
The following arithmetic operators are supported by Bourne Shell.
Assume variable **a** holds 10 and variable **b** holds 20 then –

Operator	Description	Example
+ (Addition)	Adds values on either side of the operator	`expr $a + $b` will give 30
- (Subtraction)	Subtracts right hand operand from left hand operand	`expr $a - $b` will give -10
* (Multiplication)	Multiplies values on either side of the operator	`expr $a \* $b` will give 200
/ (Division)	Divides left hand operand by right hand operand	`expr $b / $a` will give 2
% (Modulus)	Divides left hand operand by right hand operand and returns remainder	`expr $b % $a` will give 0

expr Command:

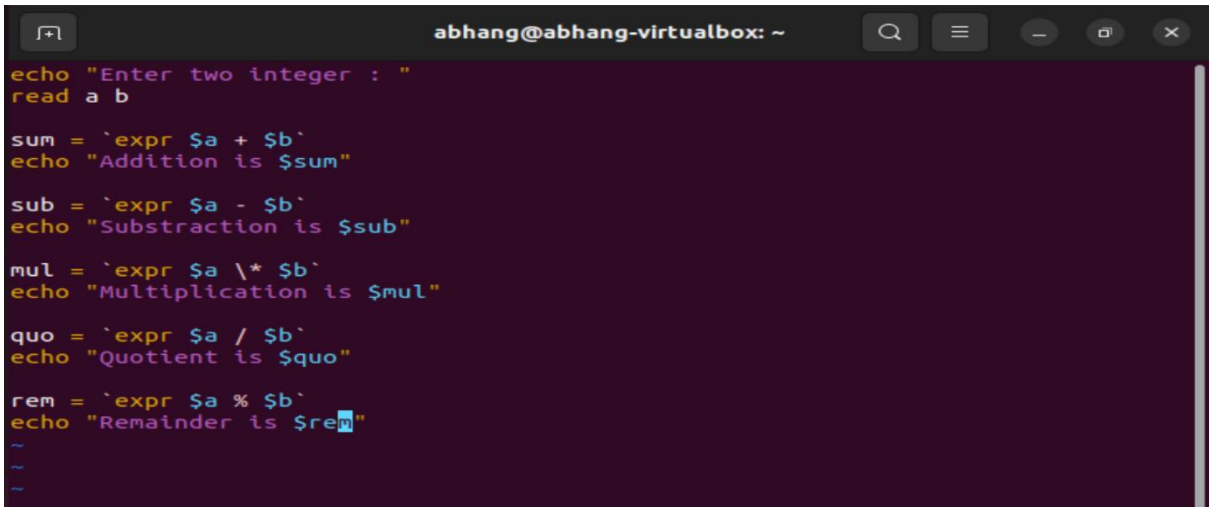
The **expr** command in Unix evaluates a given expression and displays its corresponding output. It is used for:

- Basic operations like addition, subtraction, multiplication, division, and modulus on integers.
- Evaluating regular expressions, string operations like substring, length of strings etc.

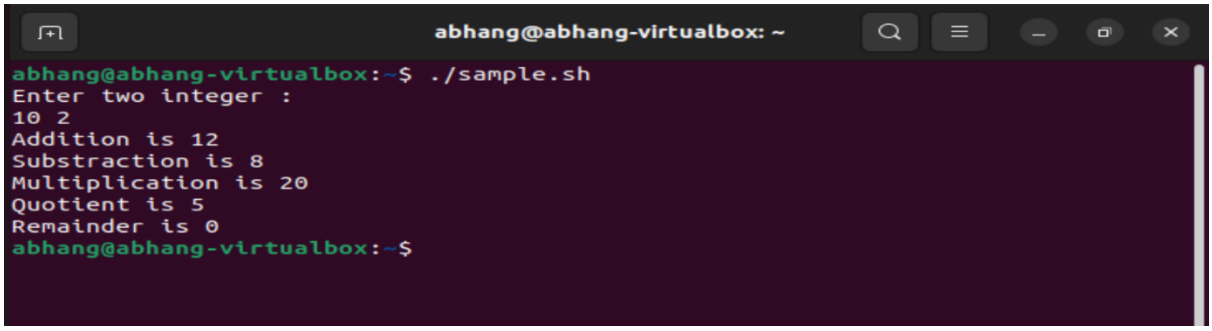
Syntax:

$expr expression

Program :



Output:



Conclusion: In this practical, we have successfully developed program to perform mathematical operations using expr command.