### Practical File

### Of

### Operating System

### 22CS005

#### Submitted

#### in partial fulfillment for the award of the degree of

## BACHELEOR OF ENGINEERING

***in***

COMPUTER SCIENCE & ENGINEERING

****

**CHITKARA UNIVERSITY**

**CHANDIGARH-PATIALA NATIONAL HIGHWAY**

**RAJPURA (PATIALA) PUNJAB-140401 (INDIA)**

##### June, 2024

##### **Submitted To: Submitted By:**

##### Faculty name Name

##### Designation Roll No.

##### Chitkara University, Punjab Sem, Batch

**INDEX**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Practical Name** | **Teacher Sign** |
| 1 | 1. Installation: Configuration & Customizations of Linux 2. Introduction to GCC compiler: Basics of GCC, Compilation of program, Execution of program. 3. Time stamping in Linux. 4. Automating the execution using Make file. |  |
| 2 | Implement the basic and user status commands like: su, sudo, man, help, history, who, whoami, id, uname, uptime, free, tty, cal, date, hostname, reboot, clear |  |
| 3 | Implement the commands that is used for Creating and Manipulating files: cat, cp, mv, rm, ls and its options, touch and their options, which is, where is, what is |  |
| 4 | Implement Directory oriented commands: cd, pwd, mkdir, rmdir, Comparing Files using diff, cmp, comm. |  |
| 5 | Write a program to create and execute process using fork() and exec() system calls. |  |
| 6 |  |  |
| 7 |  |  |
| 8 |  |  |
| 9 |  |  |
| 10 |  |  |
| 11 |  |  |
| 12 |  |  |

**Program 1(a):** **Installation:** **Configuration & Customizations of Linux**

**Solution:**

**Linux:**

It is an open-source Unix-like operating system kernel initially developed by Linus Torvalds in 1991. It forms the core of Linux distributions, providing stability, security, and flexibility. Supported by a global community, Linux powers servers, desktops, smartphones, and embedded systems, offering a wide range of customization options.

**Virtual Box:**

VirtualBox is a free and open-source virtualization software that allows users to run multiple operating systems simultaneously on a single physical machine. It supports various guest operating systems, provides features like snapshotting and shared folders, and is widely used for development, testing, and running legacy applications.

**Linux Distributions:**

Linux distributions, or distros, are variants of the Linux operating system, each offering a unique combination of software packages, desktop environments, and system configurations. Some popular types include:

1. Debian-based distros like Ubuntu, known for stability and ease of use.

2. Red Hat-based distros such as Fedora and CentOS, favored for enterprise use and robustness.

3. Arch Linux and its derivatives appeal to users seeking minimalism and customization.

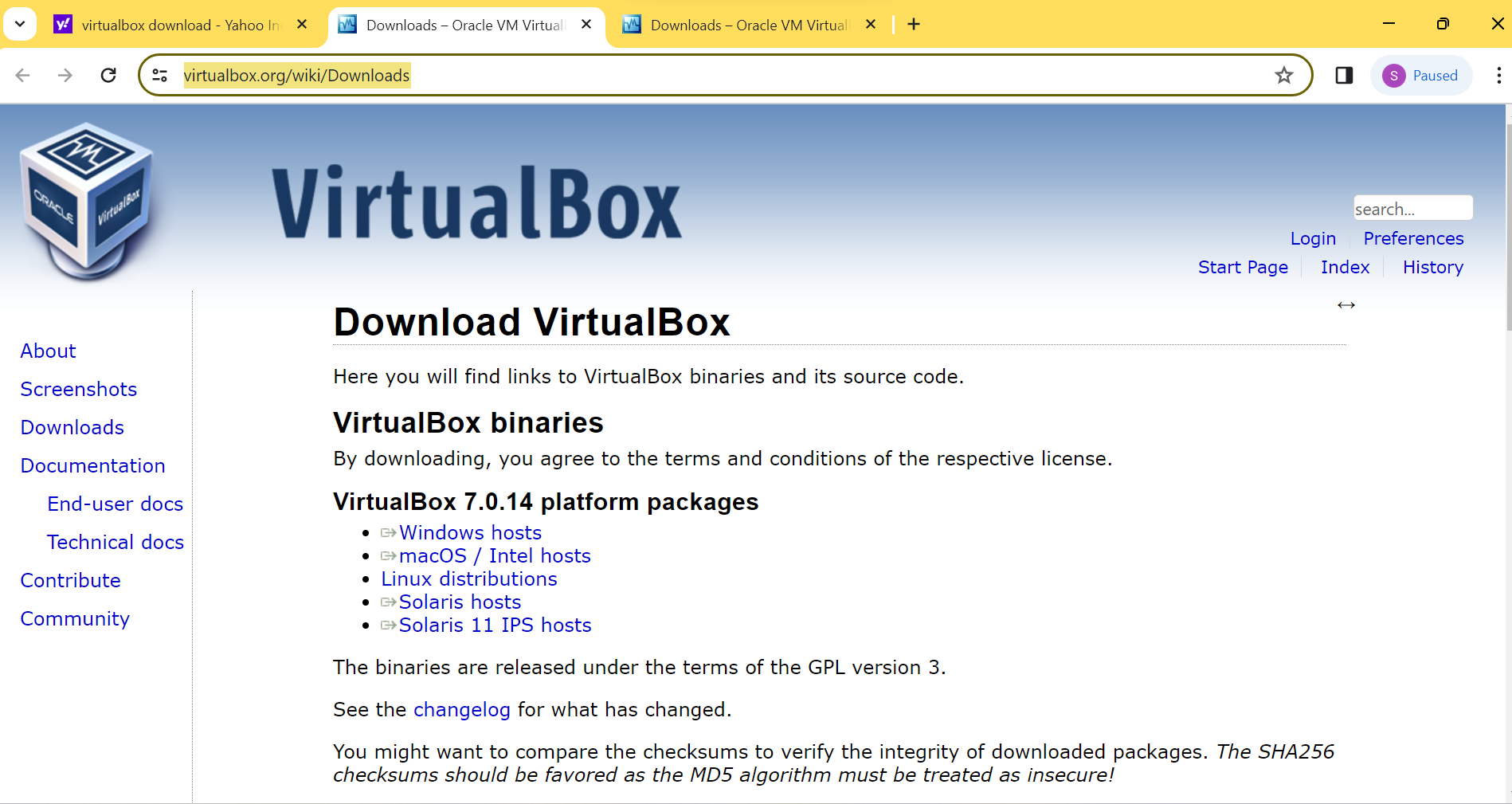
4. Gentoo and Slackware target advanced users with their DIY approach and manual configuration.

**Installation of Virtual Box:**

To install VirtualBox, first, go to the download page on the VirtualBox website(https://www.virtualbox.org/wiki/Downloads).

Select and download the correct package for your primary OS (the host OS, in

virtualization terminology).

****

Once the package is installed, install VirtualBox as you would typically install a

program. Feel free to use the default installation options.

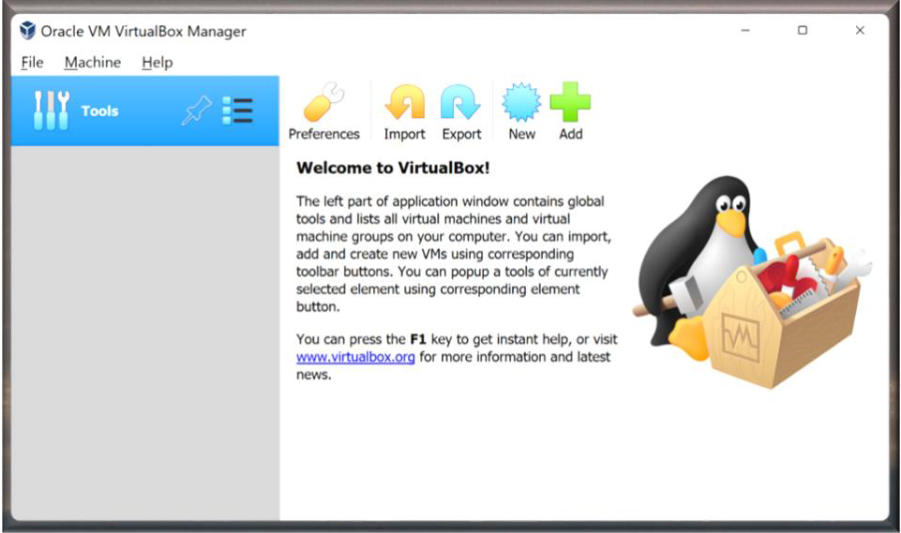
**Virtual Machine:**

In VirtualBox, a virtual machine (VM) is a software emulation of a physical computer that operates within a host operating system. It allows users to run multiple independent instances of operating systems simultaneously, enabling testing, development, and isolation of environments on a single physical machine.

**Creating VM:**

Once VirtualBox is installed on your machine, start the program and you should

see the VirtualBox Manager window open.

****

In this screen you will see options to create and add VMs to VirtualBox, as well as

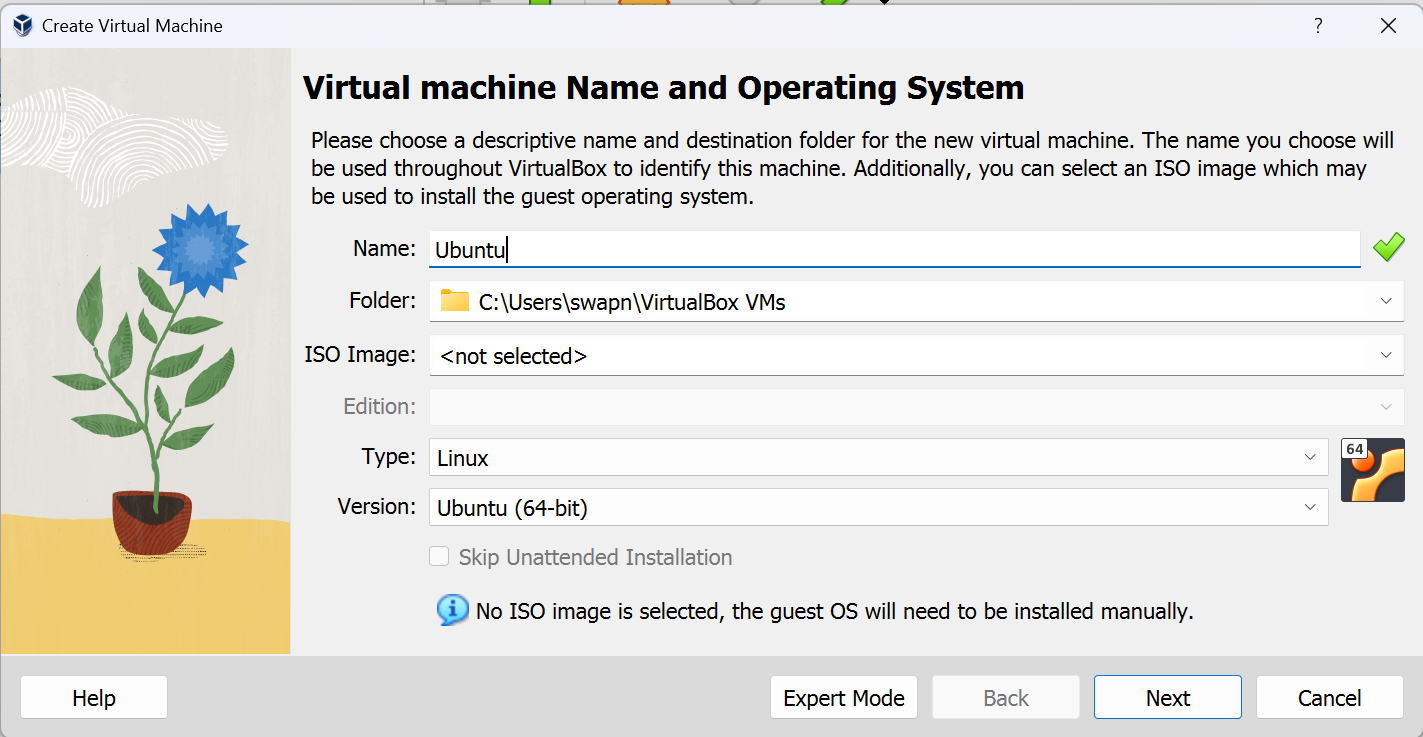
any VMs you have previously created.

Select the “New” button to create a new VM. The prompt will ask for a name for

the VM, a file path for where to save the VM files. There are options for the “type”

of OS you would like to install on your VM, such as Windows, macOS, or Linux. It

will also ask for the “version” of the OS you will install, such as Ubuntu or Debian.

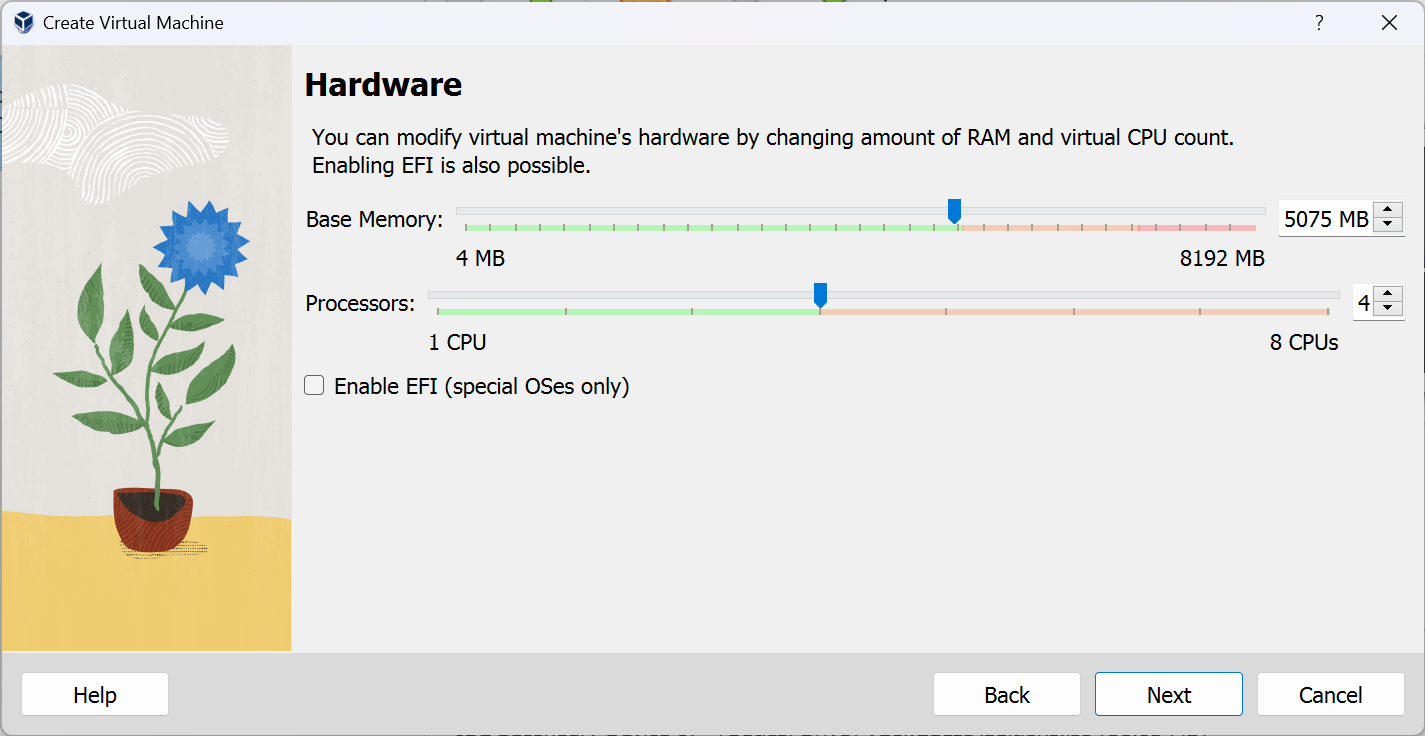
****

The following screen will ask to select the memory size to allocate for your

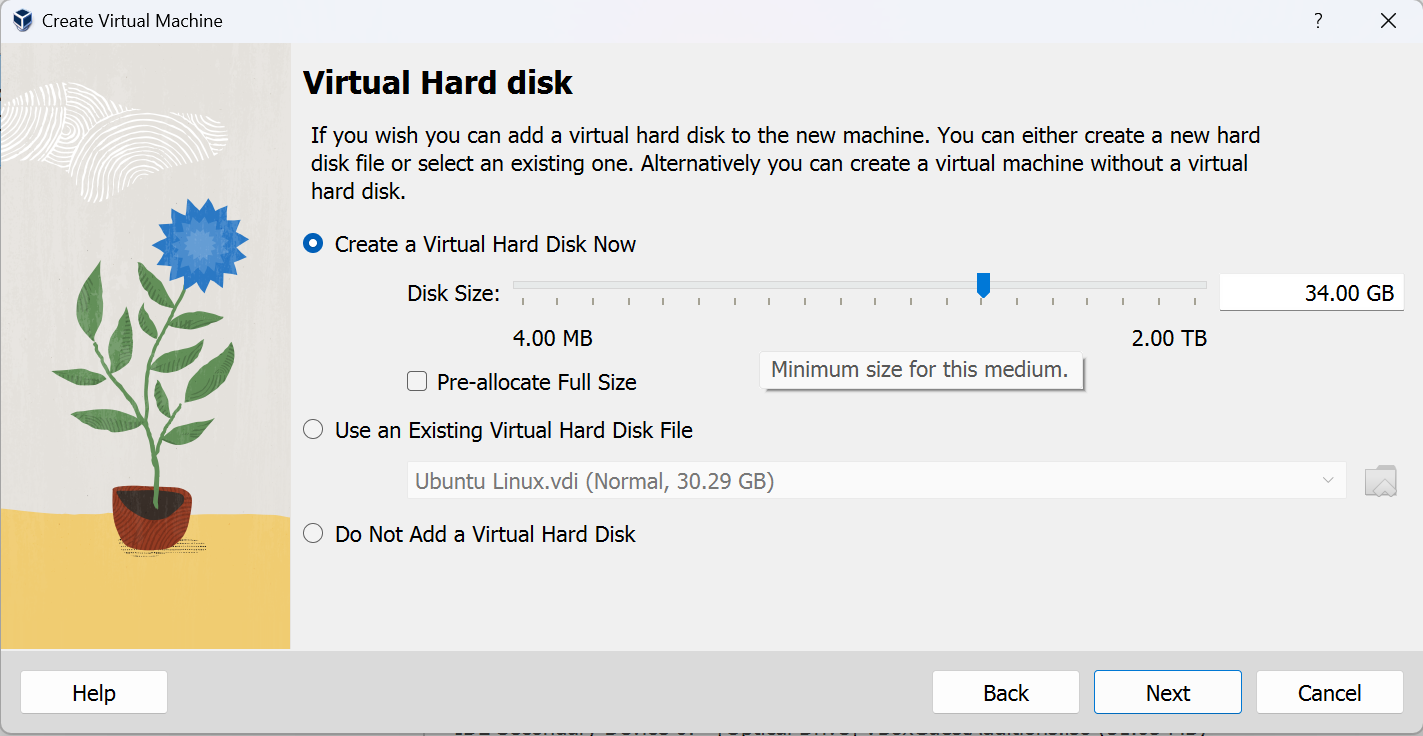
machine. This will set aside a certain amount of RAM on your host machine to be

used for your VM when it is running.

Allocate 5075 MB or complete ram which is available in green section only. Next select the number of CPU allotment.

****

Now select the Disk Size for virtual hard disk which should more than 30 GB.

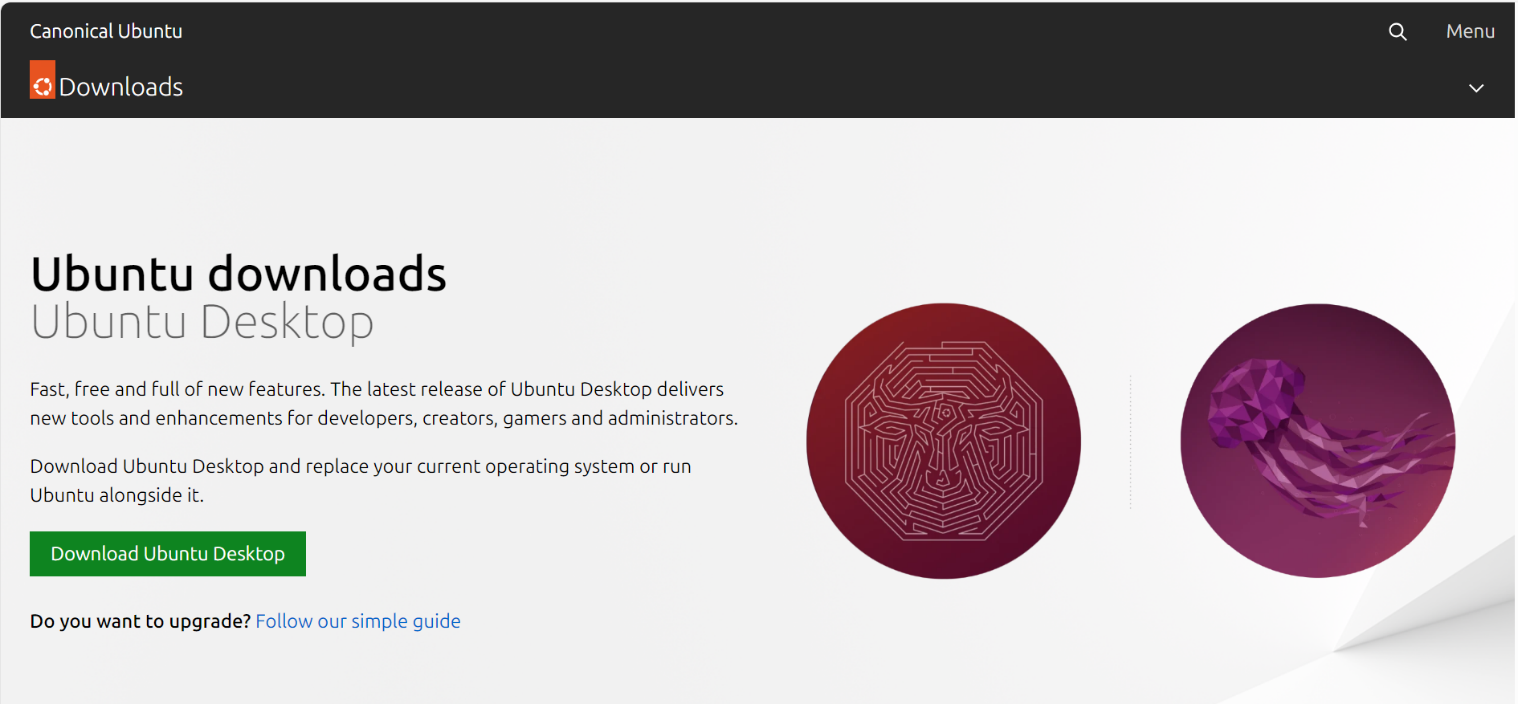
****

**Ubuntu:**

Ubuntu is a popular Linux distribution known for its user-friendly interface and community-driven development. It offers a free, open-source operating system with regular updates and long-term support options. Ubuntu includes essential software like web browsers, office suites, and multimedia applications. Its vast repository provides access to thousands of additional packages for diverse needs. With a focus on accessibility and security, Ubuntu is favored by both beginners and experienced users for its stability and versatility.

**Installation of Ubuntu:**

For installation of ubuntu ,first we need to download the ubuntu for the website(<https://ubuntu.com/download>).



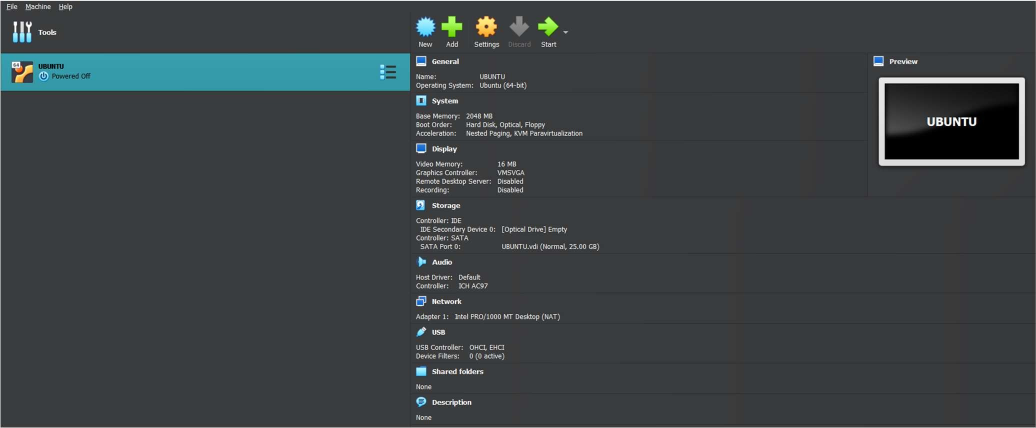
Now you should have everything ready to install Ubuntu. From the VirtualBox

Manager screen, double click the VM you created on the left and a prompt will

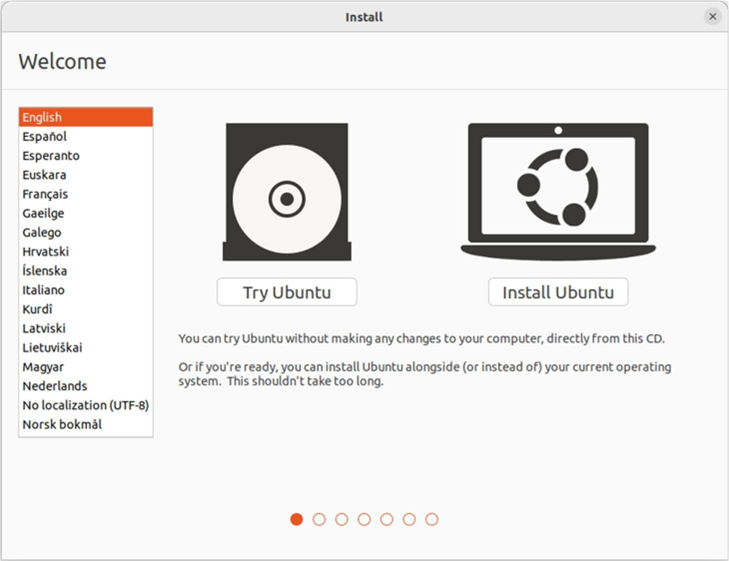
appear asking to select the start-up disk. Use the dropdown menu to select the

Ubuntu image that you downloaded in the previous step and click “Start” to boot

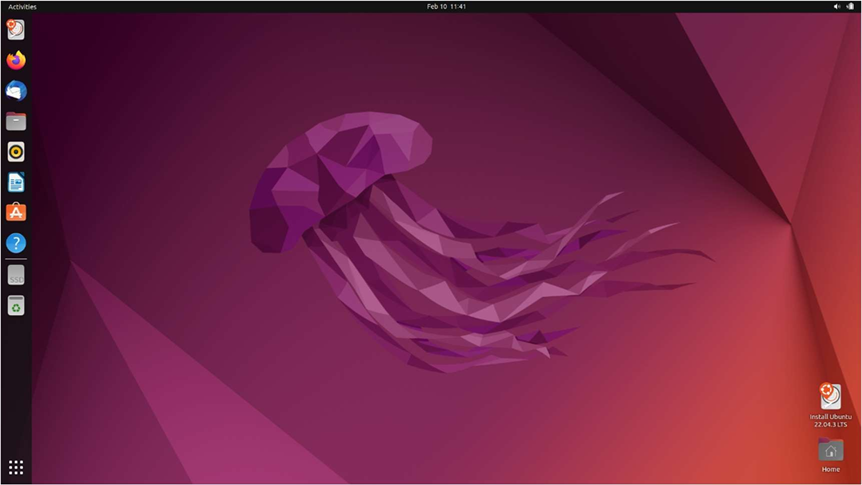
VM.



Now we will click on Start button to start the VM.



This is the first window that will appear after starting the virtual machine.



After installing the UBUNTU using default recommendations this home screen will appear which shows that we have successfully installed the UBUNTU on our virtual machine.

**Program 1(b): Introduction to GCC compiler: Basics of GCC, Compilation of program, Execution of program.**

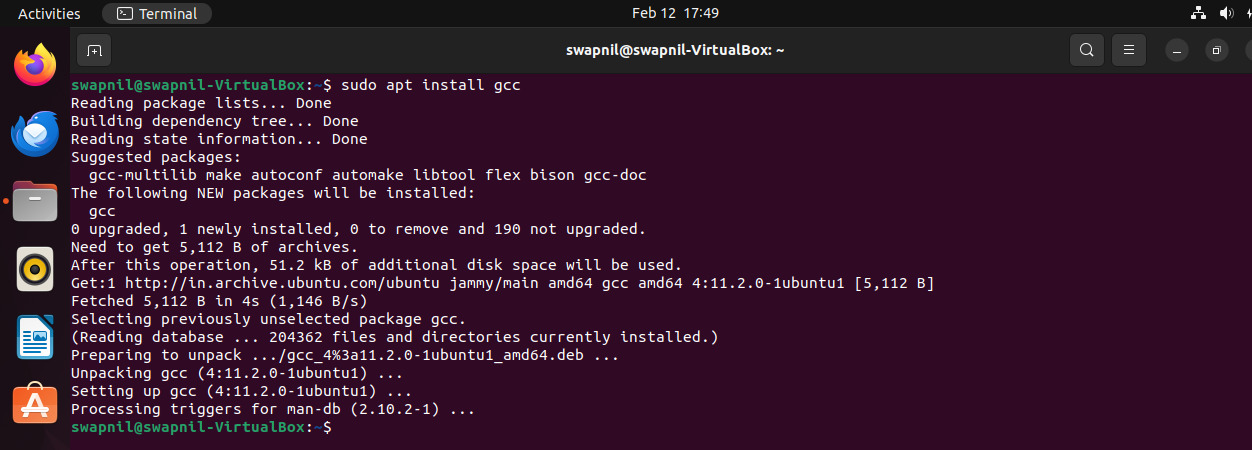
**Theory:**

**GNU Compiler Collection:**

GCC is a widely-used open-source compiler system supporting various programming languages like C, C++, and Fortran. Developed by the GNU Project, it provides tools for compiling, linking, and optimizing code on multiple platforms. Known for its portability and standards compliance, GCC offers a plethora of optimization options and supports numerous architectures. It is favored by developers for its robustness, performance, and extensive community support, making it a cornerstone of many software development projects.

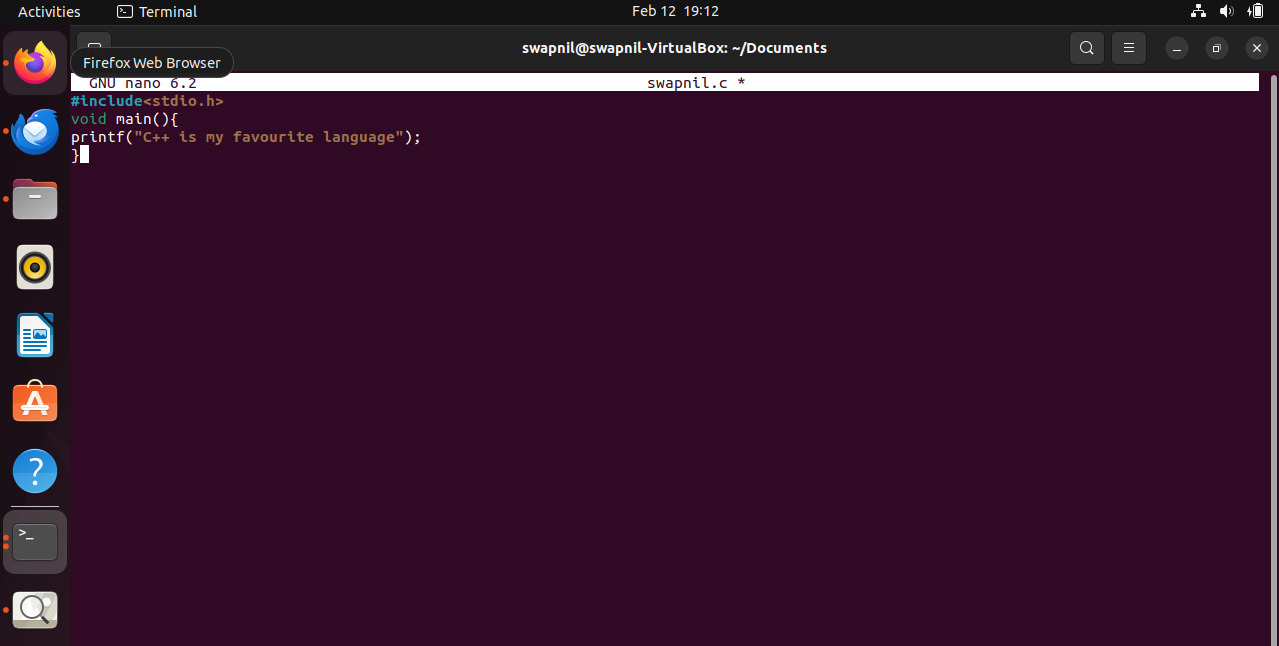
**Steps:**

1.Firstly open the VM and open Terminal in UBUNTU. Now to install GCC compiler in ubuntu we will have to write a command on terminal to install gcc compiler in it sudo apt install gcc.



2.After installing of gcc compiler, then we need to create a text file in formate of filename.c.

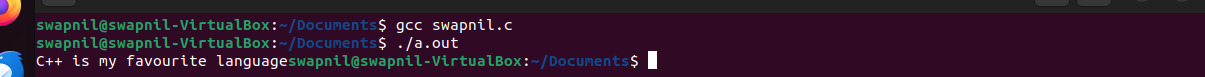
3.Type command “nano filename.c” to open text editor here we will write basic C program which we have named swapnil.c.



4.now press ctrl+x to exit the text editor.

5.now type gcc filename and press enter key. this will compile the C code which we have wrote earlier.

6.To generate the output of the compiled C program type ./a.out in the terminal “Hello there.” will be printed.

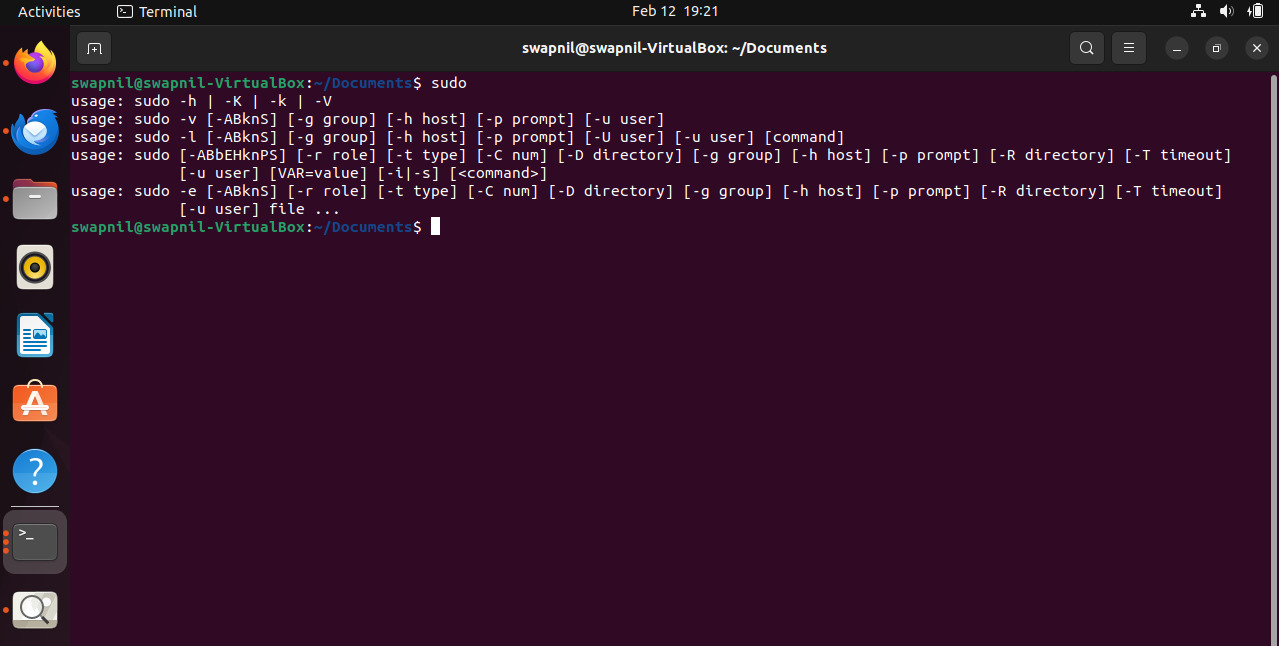


**Experiment 2: Implement the basic and user status commands like: su, sudo, man, help,** **history, who, whoami, id, uname, uptime, free, tty, cal, date, hostname, reboot, clear**

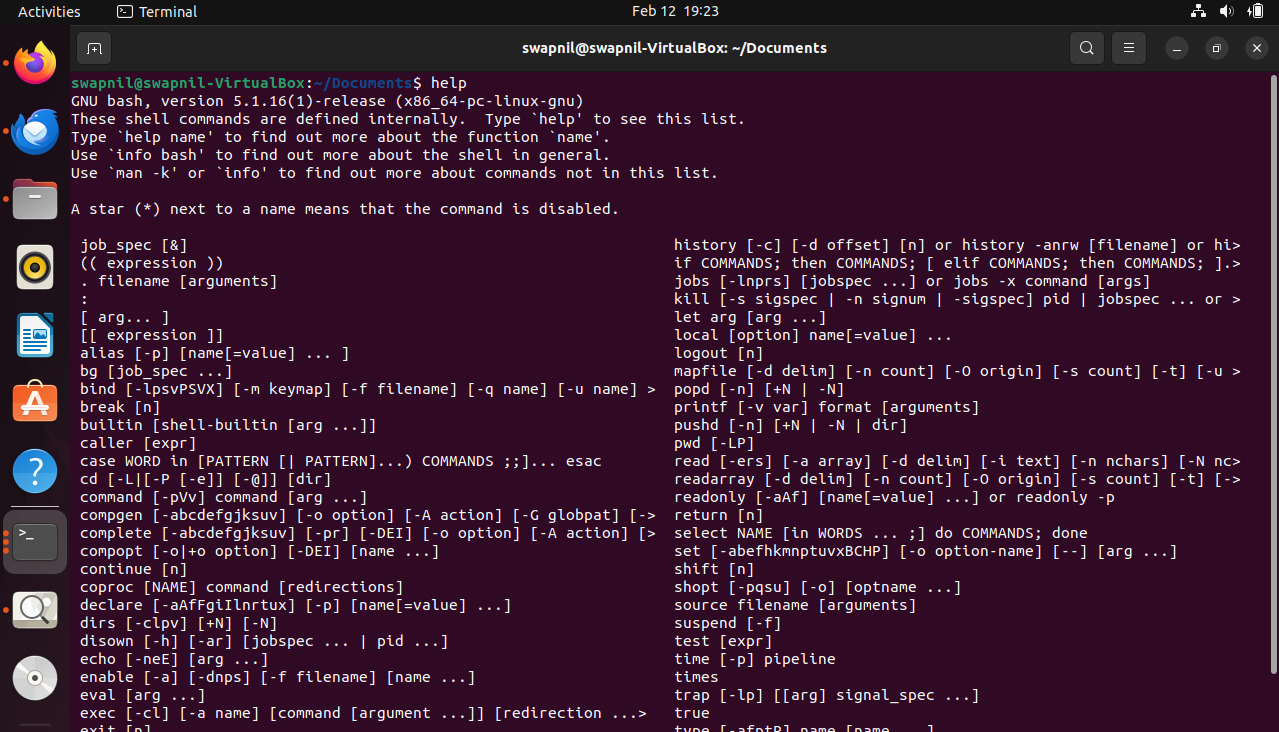
**Commands:**

**su:** The 'su' command in Linux allows users to switch to another user account, often the root account, with appropriate permissions.

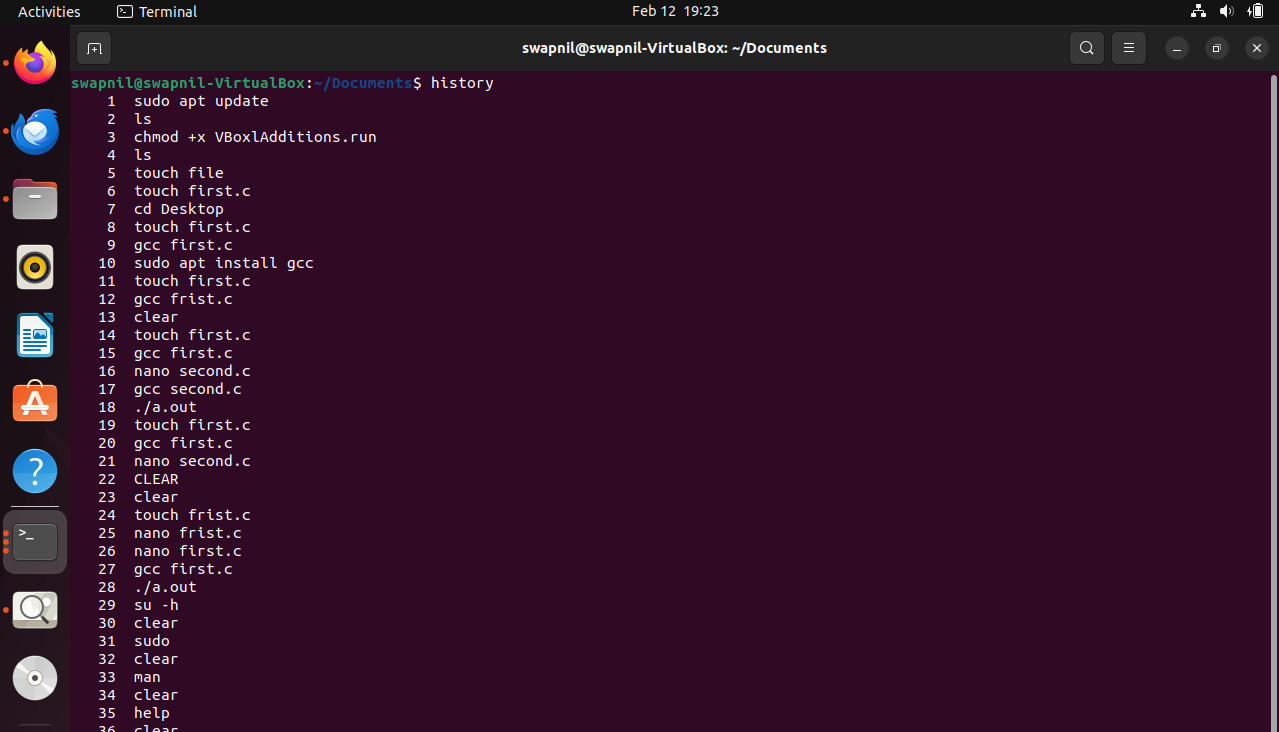
**sudo**: sudo (superuser do) is a command in Linux that allows users to run programs with the security privileges of another user.



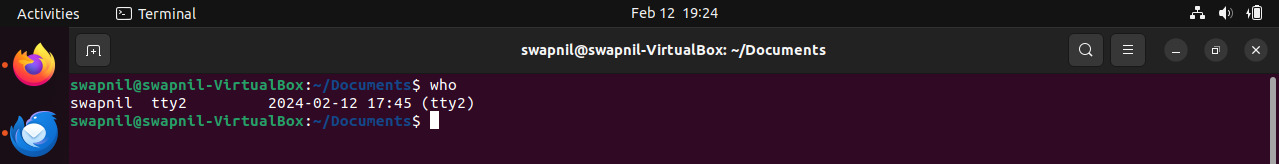
**help:** The "help" command in Linux provides information and usage instructions for built-in shell commands and utilities.



**history:** The "history" command in Linux displays a list of previously executed commands in the terminal session. It aids in recall.



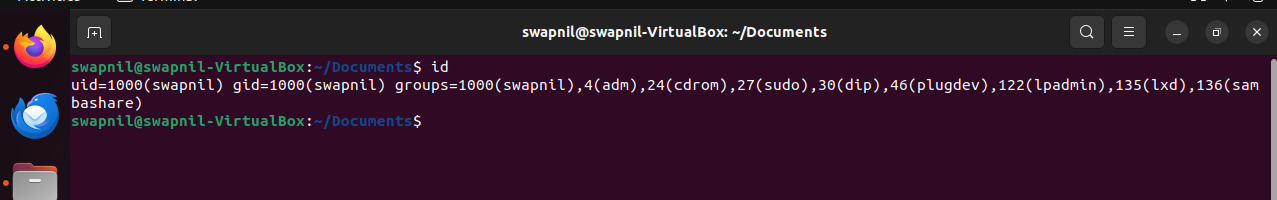
**who:** The “who” command in Linux displays information about users who are currently logged in to the system.

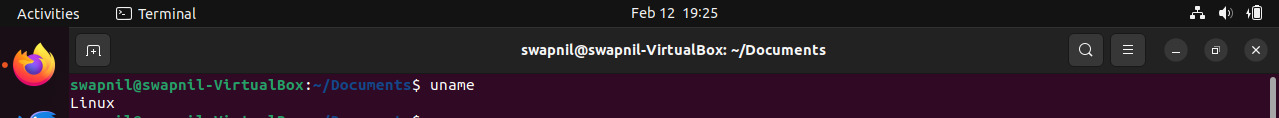


**whoami:** The "whoami" command in Linux displays the username of the current user logged in to the system.

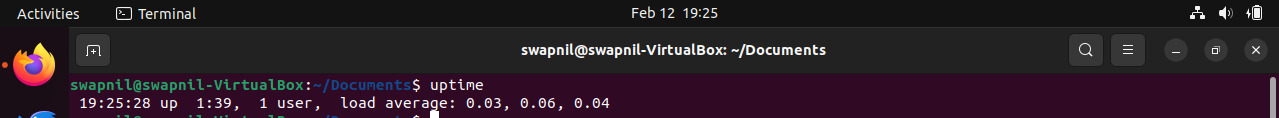


**id:** The 'id' command in Linux displays user and group information, including user ID (UID) and group ID (GID).

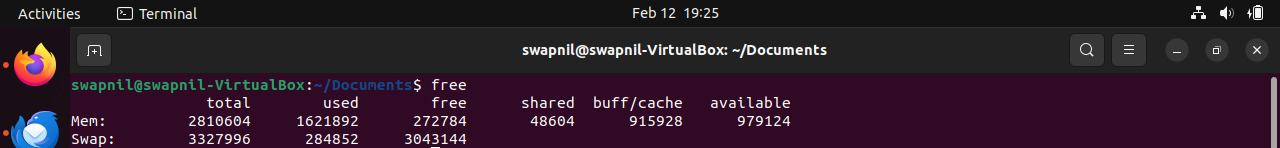
**uname:** The `uname` command in Linux displays system information like kernel version, hostname, hardware architecture, and operating system.



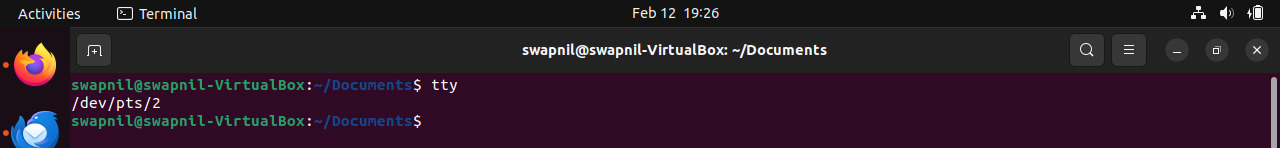
**uptime:** The `uptime` command in Linux displays system uptime, load average, and number of logged-in users in a concise format.



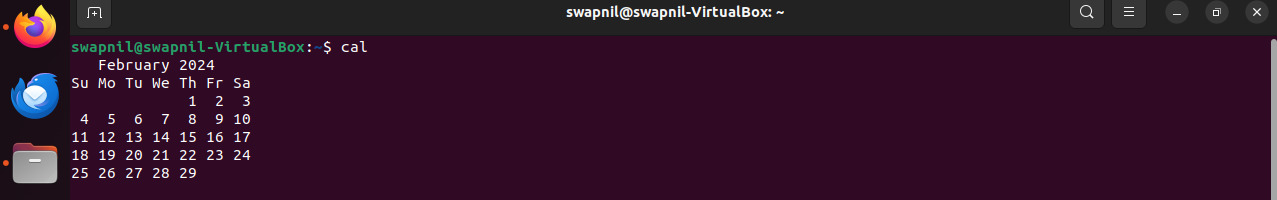
**free:** The "free" command in Linux displays system memory usage including total, used, and free memory, as well as swap usage.



**tty:** The 'tty' command in Linux prints the file name of the terminal connected to standard input.



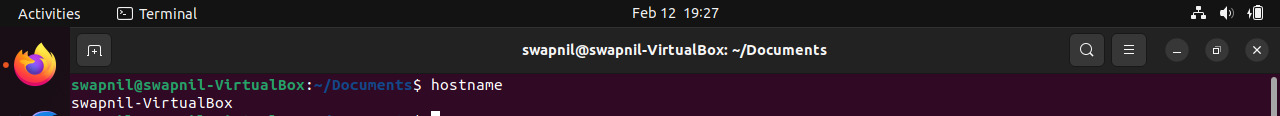
**cal:** The "cal" command in Linux displays a calendar for the current month or specified month/year. It's used for simple calendrical reference.



**date:** The `date` command in Linux displays or sets the system date and time. It's used for various time-related operations.



**hostname:** The 'hostname' command in Linux displays or sets the system's hostname, which identifies the system on a network.



**reboot:** The "reboot" command in Linux restarts the system, terminating all processes and reloading the operating system.

**clear:** The `clear` command in Linux clears the terminal screen, providing a clean workspace by removing previous output.

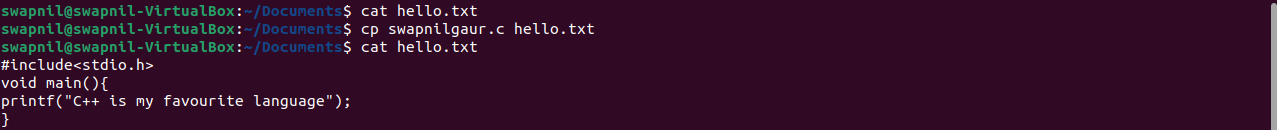
**Program 3: Implement the commands that is used for Creating and Manipulating files: cat, cp, mv, rm, ls and its options, touch and their options, which is, where is, what is**

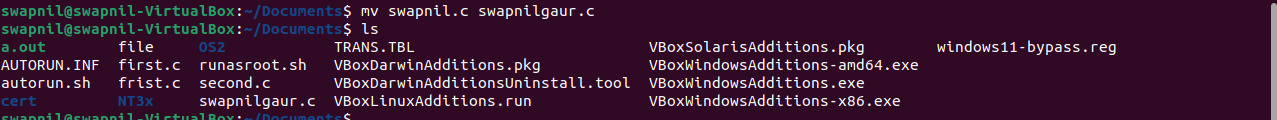
**Solution:**

**cat:** The 'cat' command in Linux concatenates files and displays their contents, commonly used to read files or combine them.

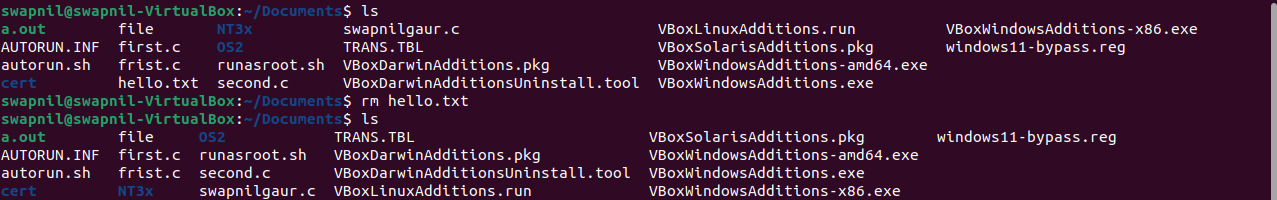


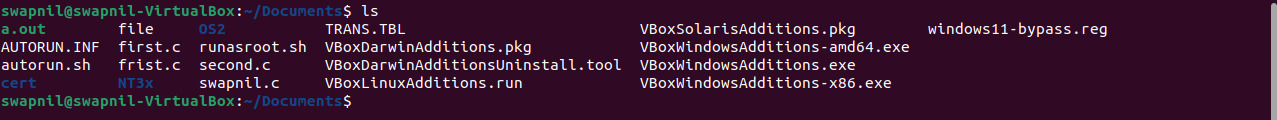
**cp:** The "cp" command in Linux is used to copy files or directories from one location to another.



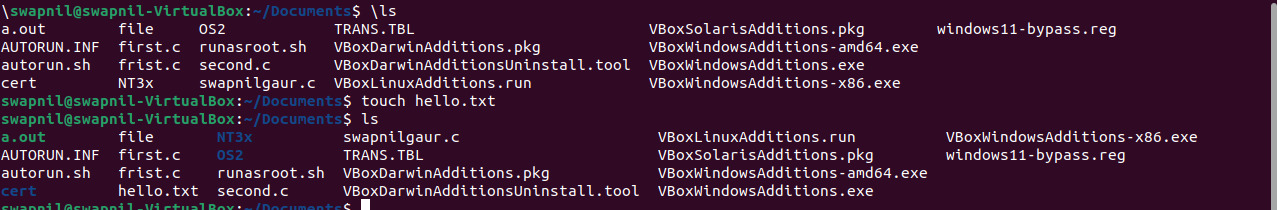
**mv:** The `mv` command in Linux is used to move files or directories from one location to another within the system. 

**rm:** The "rm" command in Linux is used to remove files or directories. Use with caution as it's irreversible.



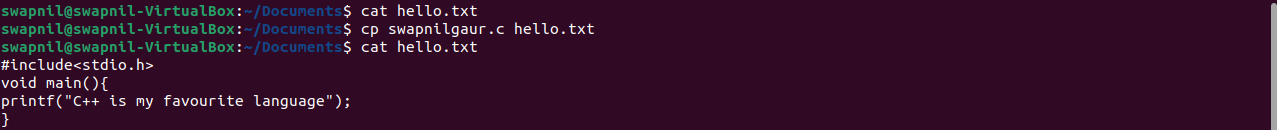
**ls:** The "ls" command in Linux lists directory contents. It displays files and directories within the specified directory. 

**touch:** The 'touch' command in Linux is used to create empty files or update file timestamps. It's versatile and handy.

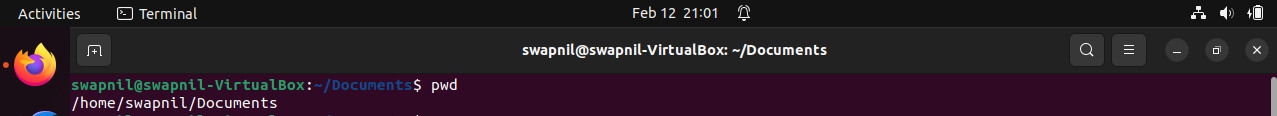


**Program 4: Implement Directory oriented commands: cd, pwd, mkdir, rmdir, Comparing Files using diff, cmp, comm.**

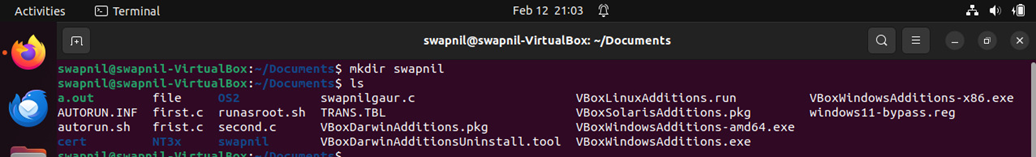
**cd:** The "cd" command in Linux is used to change the current working directory within the file system.



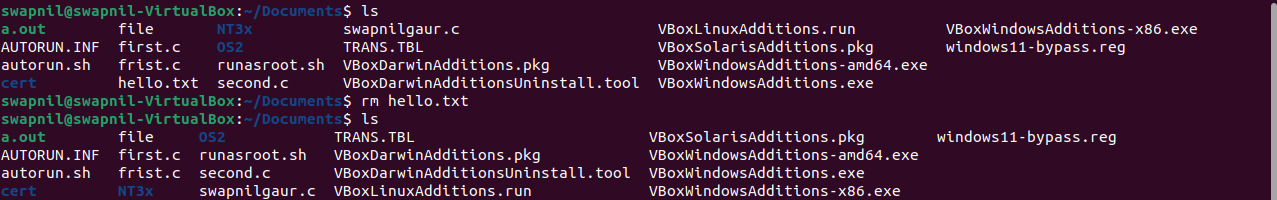
**pwd:** The “pwd” command in Linux prints the current working directory, showing the path of the directory you're currently in.



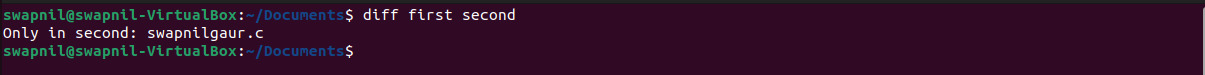
**mkdir:** “mkdir” creates directories in Linux. Usage: “mkdir directory\_name” to create a directory named `directory\_name` in the current location.

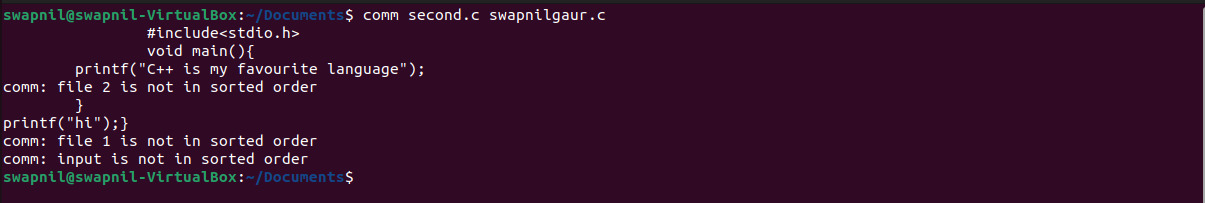


**rmdir:** The “rmdir” command in Linux is used to remove empty directories from the filesystem. It cannot delete directories with content.



**diff:** The `diff` command in Linux compares files line by line and displays the differences between them in a human-readable format.



**comm:** “comm” command in Linux is used to compare sorted files line by line and display lines unique to each file. 

**Program 5: Write a program to create and execute process using fork() system calls.**

**Solution1:**

#include<stdio.h>

#include<unistd.h>

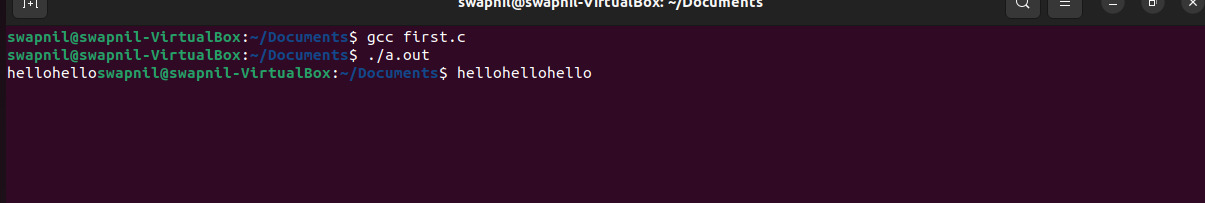
void main(){

if (fork()||fork()){

fork();}

printf("hello");}

**Output:**



**Solution2:**

#include<stdio.h>

#include<unistd.h>

void main(){

if (fork()&&fork()){

fork();}

printf("hello");}

**Output:**

