Degree Engineering

A Laboratory Manual for

Operating Systems
(3140702)

[B.E. (Computer Engineering) : Semester - 4]

Enrolment No	220170107141
Name	Thakar Atri Kamleshkumar
Branch	Computer Engineering
Academic Term	4th
Institute Name	VGEC

Directorate of Technical Education, Gandhinagar, Gujarat

Vishwakarma Government Engineering College, Chandkheda, Ahmedabad

Department of Computer Engineering

CERTIFICATE

*This is to certify that Mr./Ms._____Thakar Atri Kamleshkumar_____ Enrollment No. 220170107141 of B.E. Semester - **IV** from **Computer Engineering Department** of this Institute (GTU Code: 017) has satisfactorily completed the Practical / Tutorial work for the subject **Operating System (3140702)** for the academic year 2024-25.*

Place: _____

Date: _____

Signature of Course Faculty **Head of the Department**

Preface

Main motto of any laboratory/practical/field work is for enhancing required skills as well as creating ability amongst students to solve real time problem by developing relevant competencies in psychomotor domain. By keeping in view, GTU has designed competency focused outcome-based curriculum for engineering degree programs where sufficient weightage is given to practical work. It shows importance of enhancement of skills amongst the students and it pays attention to utilize every second of time allotted for practical amongst students, instructors and faculty members to achieve relevant outcomes by performing the experiments rather than having merely study type experiments. It is must for effective implementation of competency focused outcome-based curriculum that every practical is keenly designed to serve as a tool to develop and enhance relevant competency required by the various industry among every student. These psychomotor skills are very difficult to develop through traditional chalk and board content delivery method in the classroom. Accordingly, this lab manual is designed to focus on the industry defined relevant outcomes, rather than old practice of conducting practical to prove concept and theory.

By using this lab manual students can go through the relevant theory and procedure in advance before the actual performance which creates an interest and students can have basic idea prior to performance. This in turn enhances pre-determined outcomes amongst students. Each experiment in this manual begins with competency, industry relevant skills, course outcomes as well as practical outcomes (objectives). The students will also achieve safety and necessary precautions to be taken while performing practical.

This manual also provides guidelines to faculty members to facilitate student centric lab activities through each experiment by arranging and managing necessary resources in order that the students follow the procedures with required safety and necessary precautions to achieve the outcomes. It also gives an idea that how students will be assessed by providing rubrics.

Operating System is one of the core courses in Computer Engineering discipline. It includes basic working and application of Operating System. Process and Thread management with different Process scheduling Algorithms like FCFS, SJF, RR, Priority, etc. Concurrency control mechanisms of processes. To understand Inter Process Communication: Race Conditions, Critical Section, And Mutual Exclusion concepts are needed. Deadlock and its solutions for uninterrupted execution of processes. Memory management in Operating system. Input output and Disk scheduling algorithms. Virtualization concepts, Development of basic code of operating system using Shell Scripts.

Utmost care has been taken while preparing this lab manual however always there is chances of improvement. Therefore, we welcome constructive suggestions for improvement and removal of errors if any.

DTE's Vision

- To provide globally competitive technical education
- Remove geographical imbalances and inconsistencies
- Develop student friendly resources with a special focus on girls' education and support to weaker sections
- Develop programs relevant to industry and create a vibrant pool of technical professionals

Institute's Vision

- To create an ecosystem for proliferation of socially responsible and technically sound engineers, innovators and entrepreneurs.

Institute's Mission

- To develop state-of-the-art laboratories and well-equipped academic infrastructure.
- To motivate faculty and staff for qualification up-gradation, and enhancement of subject knowledge.
- To promote research, innovation and real life problem solving skills.
- To strengthen linkages with industries, academic and research organizations.
- To reinforce concern for sustainability, natural resource conservation and social responsibility.

Department's Vision

- To create an environment for providing value based education in Computer Engineering through innovation, team work and ethical practices.

Department's Mission

- To produce computer engineering graduates according to the needs of industry, government, society and scientific community.
- To develop state of the art computing facilities and academic infrastructure.
- To develop partnership with industries, government agencies and R & D organizations for knowledge sharing and overall development of faculties and students.
- To solve industrial, governance and societal issues by applying computing techniques.
- To create environment for research and entrepreneurship.

Programme Outcomes (POs)

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change

Operating System (3140702) **220170107141**

Program Specific Outcomes (PSOs)

- Sound knowledge of fundamentals of computer science and engineering including software and hardware.
- Develop the software using sound software engineering principles having web based/mobile based interface.
- Use various tools and technology supporting modern software frameworks for solving problems having large volume of data in the domain of data science and machine learning.

Program Educational Objectives (PEOs)

- Possess technical competence in solving real life problems related to Computing.
- Acquire good analysis, design, development, implementation and testing skills to formulate simple computing solutions to the business and societal needs.
- Provide requisite skills to pursue entrepreneurship, higher studies, research and development and imbibe high degree of professionalism in the fields of computing.
- Embrace life-long learning and remain continuously employable.
- Work and excel in a highly competence supportive, multicultural and professional environment which abiding to the legal and ethical responsibilities.

Practical – Course Outcome matrix

Course Outcomes (COs)	
CO_3140702.1	Analyze the structure of OS and basic architectural components involved in OS design
CO_3140702.2	Compare and contrast various CPU scheduling algorithms.
CO_3140702.3	Evaluate the requirements for the process synchronization and co-ordination in contemporary operating system.
CO_3140702.4	Analyze various algorithms for memory management, I/O management and security aspects of operating system.
CO_3140702.5	Write shell scripts in Unix/Linux O.S and write simple programs using kernel system calls. Also understand virtualization concept.

Sr. No.	**Practical Outcome/Title of experiment**	**CO 1**	**CO 2**	**CO 3**	**CO 4**	**CO 5**
1	Study of Linux/Windows system Architecture , Installation and MS DOS Commands.	√				√
2	Study and execute Basic and directory manipulation commands of LINUX/UNIX.					√
3	Study and execute Basic File manipulation commands.					√
4	Study and Execute Advance Filter Commands.					√
5	Write a shell script program using Loop/ control structure.					√
6	Loop/ control structure using shell script(Using while loop)					√
7	Command execution via Shell script.					√
8	Process Scheduling Algorithm and Comparison.		√			
9	Process creation and Thread Scheduling			√		
10	Page replacement and Disk Scheduling algorithm				√	

Industry Relevant Skills

Operating system specialists are professionals who design, install, configure, maintain, and troubleshoot various operating systems, one need to have a solid foundation of technical skills and knowledge, as well as soft skills to work on Operating system.

Guidelines for Faculty members

1. Teacher should provide the guideline with demonstration of practical to the students with all features.
2. Teacher shall explain basic concepts/theory related to the experiment to the students before starting of each practical
3. Involve all the students in performance of each experiment.
4. Teacher is expected to share the skills and competencies to be developed in the students and ensure that the respective skills and competencies are developed in the students after the completion of the experimentation.
5. Teachers should give opportunity to students for hands-on experience after the demonstration.
6. Teacher may provide additional knowledge and skills to the students even though not covered in the manual but are expected from the students by concerned industry.
7. Give practical assignment and assess the performance of students based on task assigned to check whether it is as per the instructions or not.
8. Teacher is expected to refer complete curriculum of the course and follow the guidelines for implementation.

Instructions for Students

1. Students are expected to carefully listen to all the theory classes delivered by the faculty members and understand the COs, content of the course, teaching and examination scheme, skill set to be developed etc.
2. Students will have to perform experiments on computer system on which UNIX/Linux is installed to execute programs of Operating System.
3. Students should develop programs and execute all the programs using UNIX/Linux OS. Students have to show output of each program in their practical file.
4. Students are instructed to submit practical list as per given sample list shown on next page.
5. Student should develop a habit of submitting the experimentation work as per the schedule and she/he should be well prepared for the same.

Common Safety Instructions

Students are expected to

1. switch on the PC carefully (not to use wet hands)
2. shutdown the PC properly at the end of your Lab
3. carefully handle the peripherals (Mouse, Keyboard, Network cable etc)
4. use Laptop in lab after getting permission from Teacher
5. carefully handle all lab resources

Index
(Progressive Assessment Sheet)

Sr. No.	Objective(s) of Experiment	Pg. No.	Date of performance	Date of submission	Assessment Marks	Sign. of Teacher with date	Remarks
1.	Study of Linux/Windows system Architecture , Installation and MS DOS Commands.						
	1.1 Study of LINUX/UNIX Architecture AND Installation of Ubuntu using Virtual Box. 1.2 Give the advantages of UNIX over Windows. 1.3 Execute the DOS Commands						
2.	Study and execute Basic and directory manipulation commands of LINUX/UNIX.						
	2.1 Study of Unix Shell and Environment Variables. 2.2 man, cal, date, echo, bc, who, uname 2.3 Using commands : pwd, mkdir, cd, rmdir ,ls generate given tree.						
3.	Study and execute Basic File manipulation commands.						
	Explore following commands: 1. cat 2.wc 3. cp 4. mv 5.rm 6. File 7. cmp 8. comm 9. diff 10. chmod 11. sort						
4.	Study and Execute Advance Filter Commands.						
	Explore following commands: 1.head 2. tail 3. paste 4. cut(-f) 5. cut(-c) 6. Grep						
5.	Write a shell script program using Loop/ control structure.						
	5.1 Write a shell script to find factorial of given number n. 5.2 Write a shell script which will generate first n Fibonacci numbers like: 1, 1, 2, 3, 5, 13, …"						
6.	Write a shell script program using Loop/ control structure.(Using while loop)						
	6.1 Write a shell script to read n numbers as command arguments and sort them in descending order.						

	6.2 Write a shell script to generate mark sheet of a student. Take 3 subjects, calculate and display total marks, percentage and Class obtained by the student."						
7.	Command execution via Shell script.						
	7.1 Write a shell script to display all executable files, directories and zero sized files from current directory 7.2 Write a menu driven shell script which will print the following menu and execute the given task. MENU • Display calendar of current month • Display today's date and time • Display usernames those are currently logged in the system • Display your name at given x, y position • Display your terminal number Exit						
8.	Process Scheduling Algorithm and Comparison.						
	Write a C Program to Implement Following CPU Scheduling algorithms. • FCFS • Round Robin						
9.	Process creation and Thread Scheduling						
	9.1 Implement Producer consumer problem using thread using C/JAVA programming Language. 9.2 Create new thread using fork() system call using C programming Language.						
10.	Page replacement and Disk Scheduling algorithm						
	10.1 Implement FIFO Page replacement Algorithm using C/Java. 10.2 Implement C-SCAN Disk Scheduling Algorithm using C/Java.						
Total							

Experiment No – 1

AIM : Study of Linux/Windows system Architecture , Installation and MS DOS Commands.

1.1 Study of LINUX/UNIX Architecture AND Installation of Ubuntu using Virtual Box.

1.2 Give the advantages of UNIX over Windows.

1.3 Execute the DOS Commands

Date: ***// Write date of experiment here***

Competency and Practical Skills: Logic building and programming

Relevant CO: CO1, CO5

Objectives:

a. To analyze various Operating Systems structure

b. To use different Commands.

c. To differentiate working of types of OS.

Equipment/Instruments: Computer System with Winows/Linux

Safety and necessary Precautions:

✓ Operate computer system carefully and responsibly.

✓ Use required lab resources cautiously

Theory:

Computer system can be divided roughly into four components: the hardware, the operating system, the application programs, and the users).The hardware like the central processing unit (CPU), the memory, and the input/output (I/O) devices provides the basic computing resources for the system. The application programs such as word processors, spreadsheets, compilers, and Web browsers—define the ways in which these resources are used to solve users' computing problems. The operating system controls the hardware and coordinates its use among the various application programs for the various users. In short, operating system provides the means for proper use of these resources in the operation of the computer system. It also provides an environment within which other programs can do useful work.

1.1 Study of LINUX/UNIX Architecture AND Installation of Ubuntu using Virtual Box.

- **Draw LINUX/UNIX Architecture Explain significance of each component.**

//Student has to draw Linux/Unix Architecture and also Write the Working/Significance about each Component of it.

- **Write a step to install Ubuntu On Virtual Box / Computer System.**

  (Faculty Needs to Explain How VirtualBox/Virtual machine is helpful)

Steps are Given for the Installation of Ubuntu on Virtual Machine. Students need to Follow steps to Install Linux OS on the System.

Step-1 Installing Linux using Virtual Machine

This is a popular method to install a Linux operating system. The virtual installation offers you the freedom of running Linux on an existing OS already installed on your computer. This means if you have Windows running, then you can just run Linux with a click of a button.

Step-2 Download and Install Virtual Box

Download Virtual box depending on your processor and OS, select the appropriate package. In our case, we have selected Windows with AMD. Once Download Complete open the source and run it and install VirtualBox on your System.Once it installed Open Virtualbox for further process.(snapshots are given below)

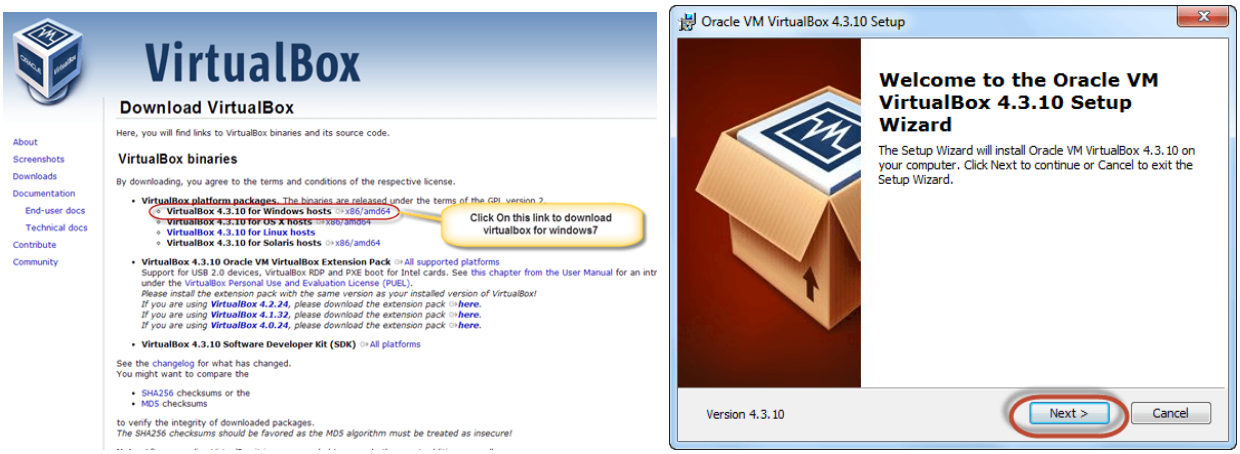


Fig.1.1 Downloading Installing Virtual Box

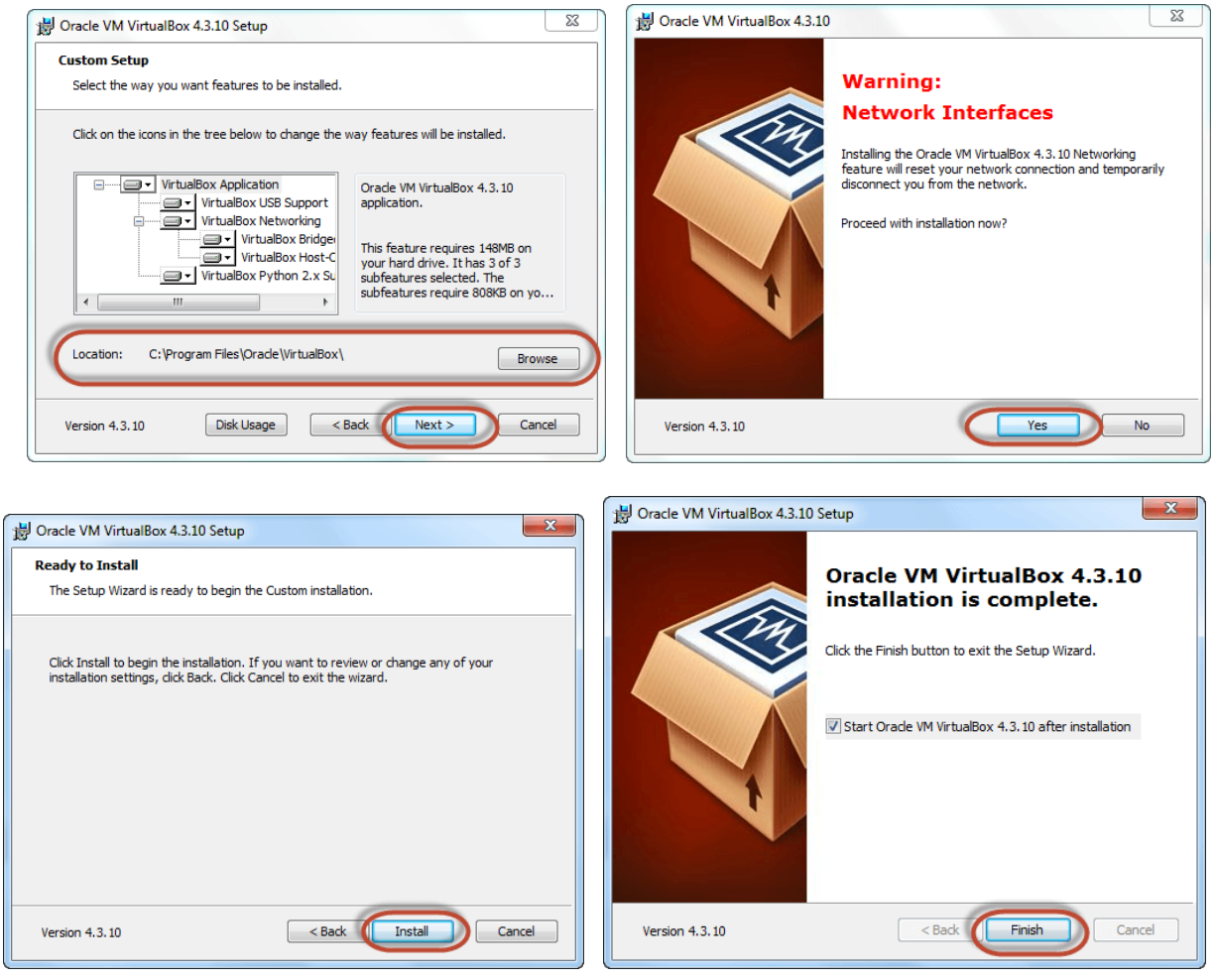


Fig.1.2 Steps to follow while installing Virtual Box

The virtual box dashboard looks like this-

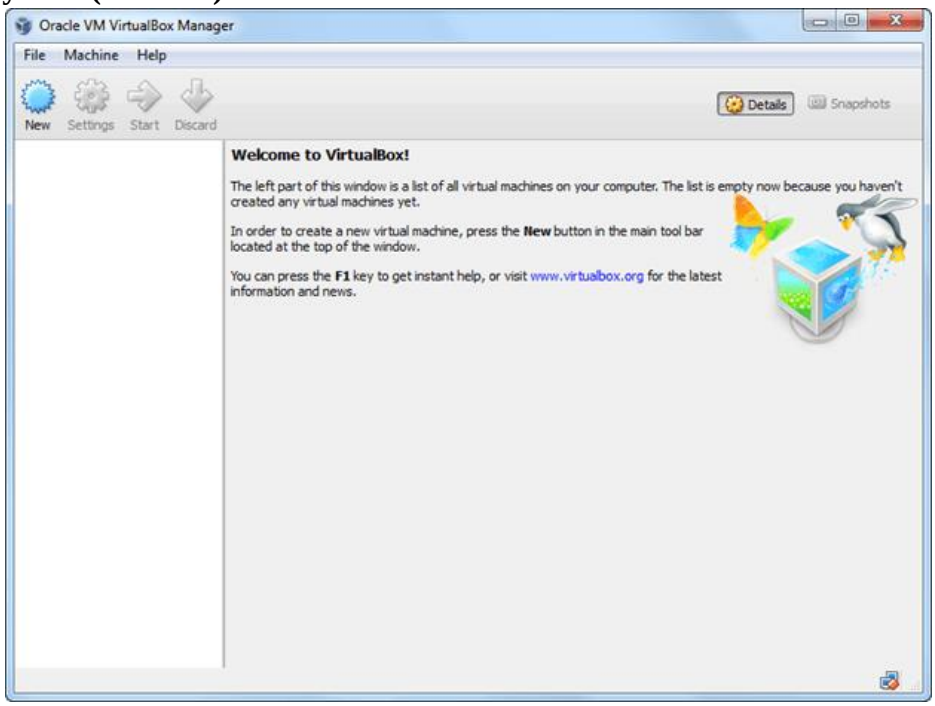


Fig.1.3 Virtual Box dashboard after installation

Step-3 Download Ubuntu AND INSTALL.For download **You can select 32/64-bit versions as per your choice.**

Fig.1.4 Downloading Ubuntu

Step-4 Create a Machine in Virtual Box. Open Virtual box and click on new button

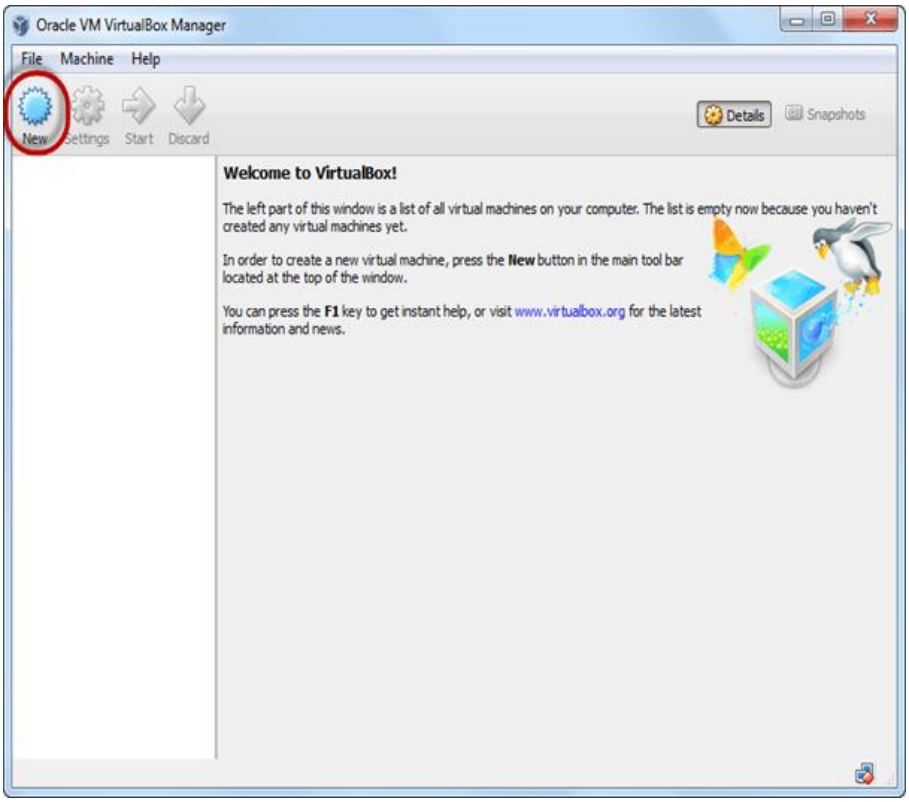


Fig.1.5 instant starting of virtual box for ubuntu installation virtual Box

Step-5 In next window**,** give the name of your OS which you are installing in virtual box. And select OS like Linux and version as Ubuntu 32 bit. And click on next

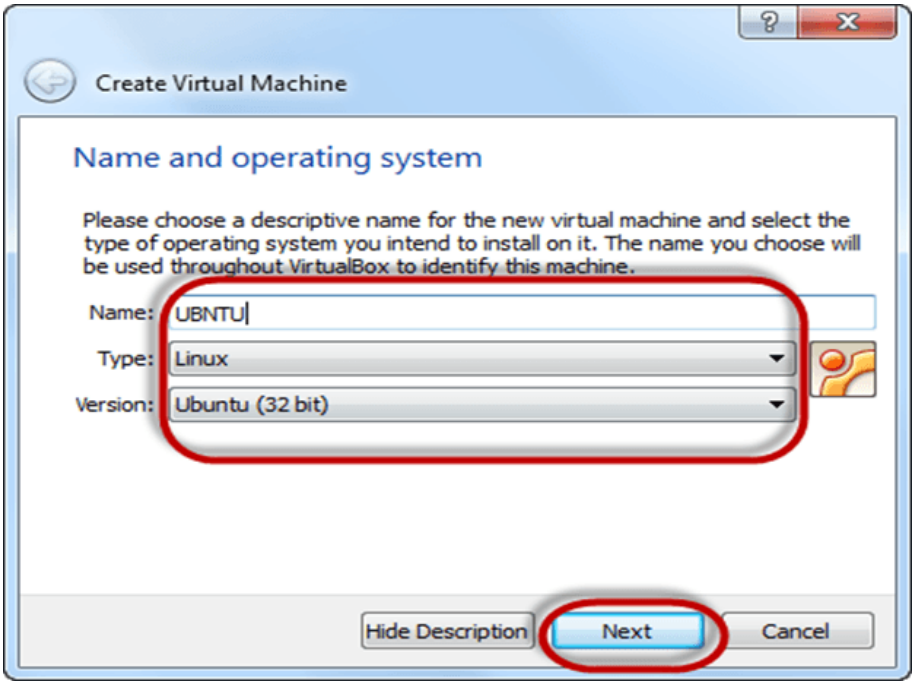


Fig.1.6 providing basic system details for ubuntu installation

Step-6 Now Allocate RAM Size To your Virtual OS. It recommended keeping 1024MB (1 GB) RAM to run Ubuntu better. And click on next.

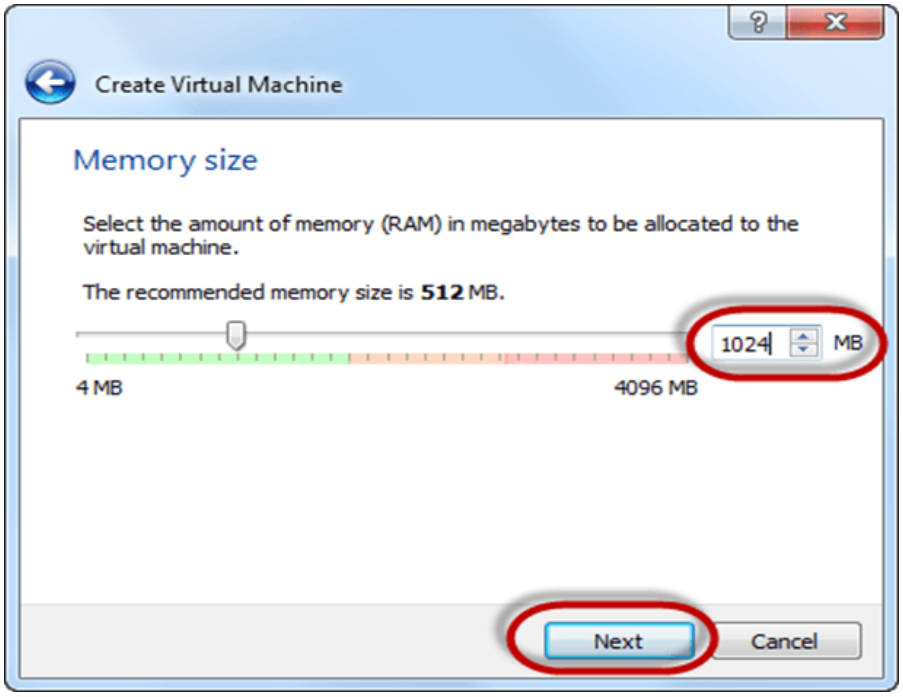


Fig.1.7 providing memory requirement for ubuntu installation

Step-7 Now To run OS in virtual box we have to create virtual hard disk, click on create a virtual hard drive now and click on create button. The virtual hard disk is where the OS installation files and data/applications you create/install in this Ubuntu machine will reside

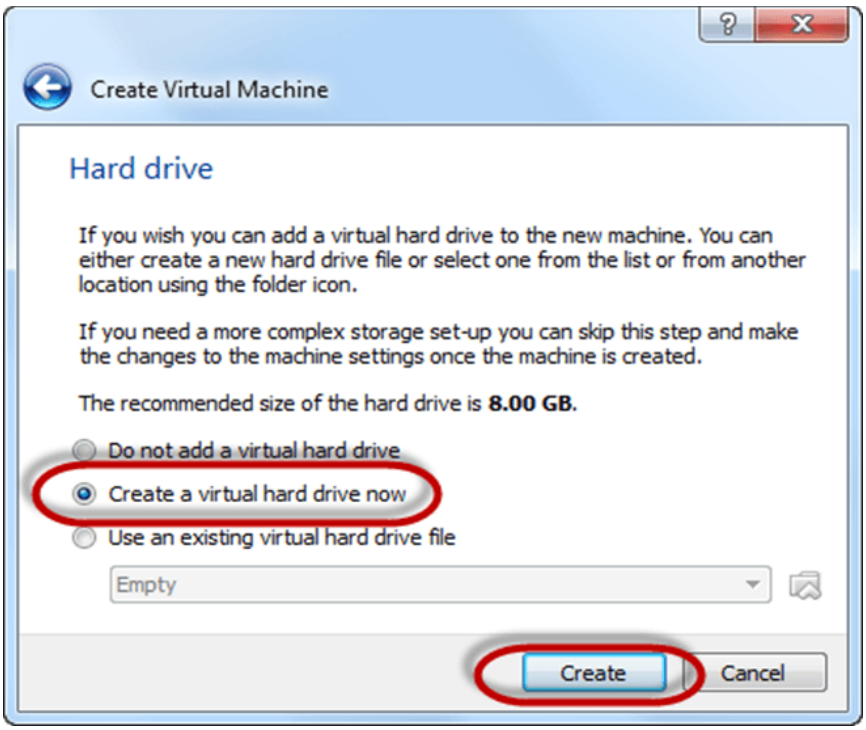


Fig.1.8 creating virtual disk drive for ubuntu installation

Step-8 select VHD (virtual hard disk) option and click on next.

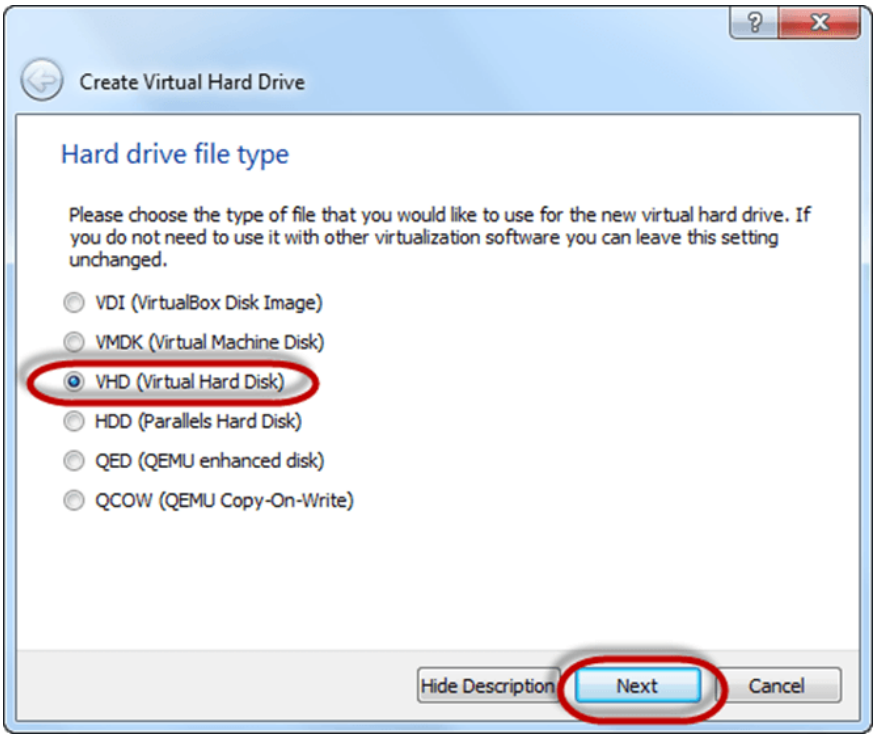


Fig.1.9 creating virtual disk drive for ubuntu installation

Step-9 Click on dynamic allocated and click on next. This means that the size of the disk will increase dynamically as per requirement.

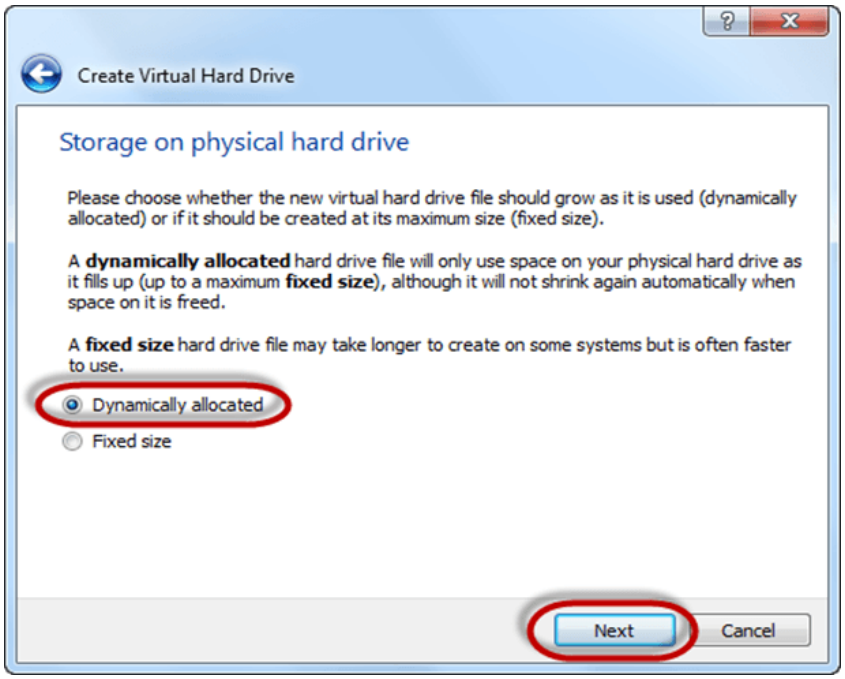


Fig.1.10 creating virtual disk drive for ubuntu installation

Step-10 Allocate memory to your virtual hard drive .8GB recommended. Click on create button.

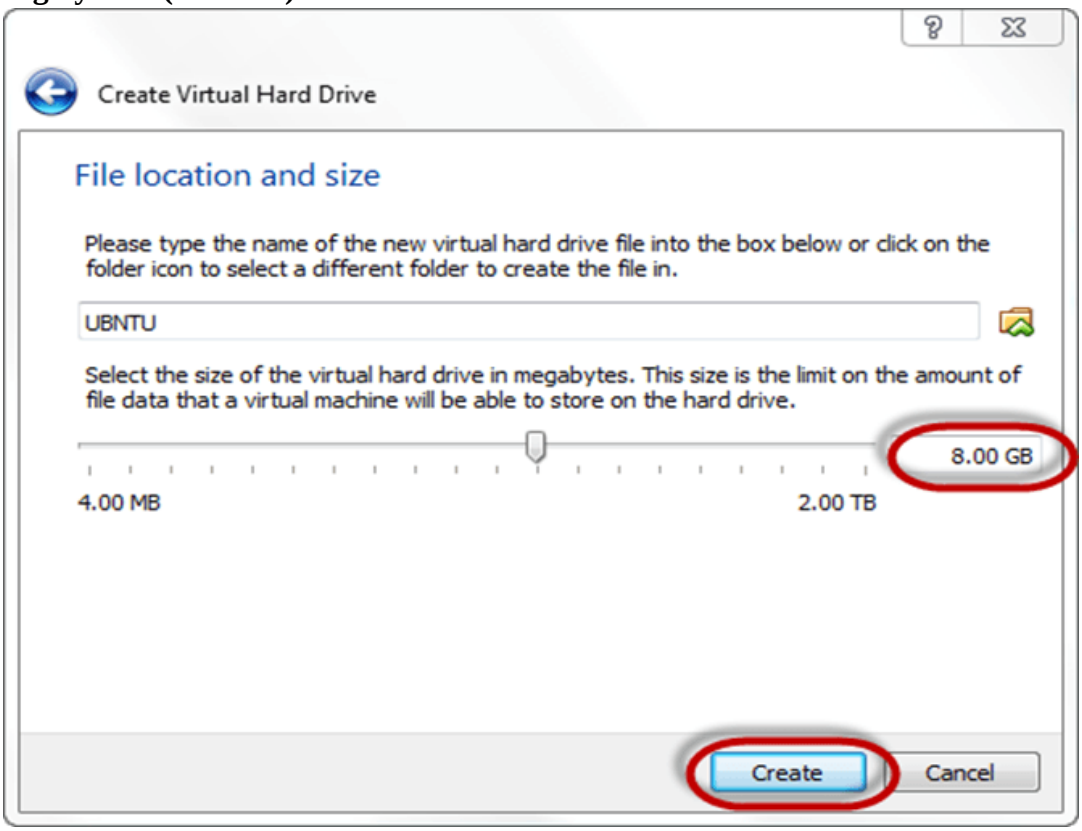


Fig.1.11 Memory allocation for ubuntu installation

Step-11 Now you can see the machine name in left panel

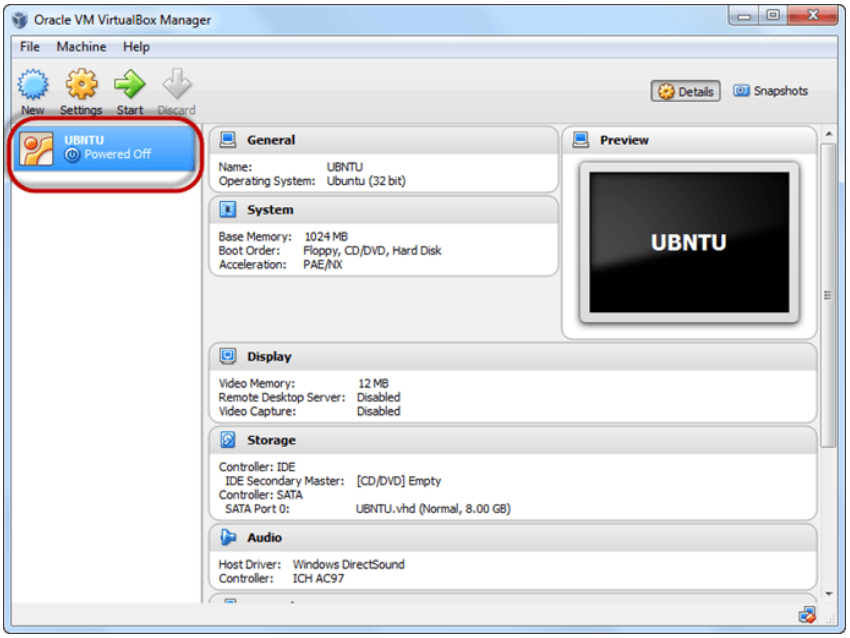


Fig.1.12 machine configuration ready for ubuntu installation

So a Machine (PC) with 8GB Hardisk, 1GB RAM is ready.

Step 12 Select the Machine and Click on Start

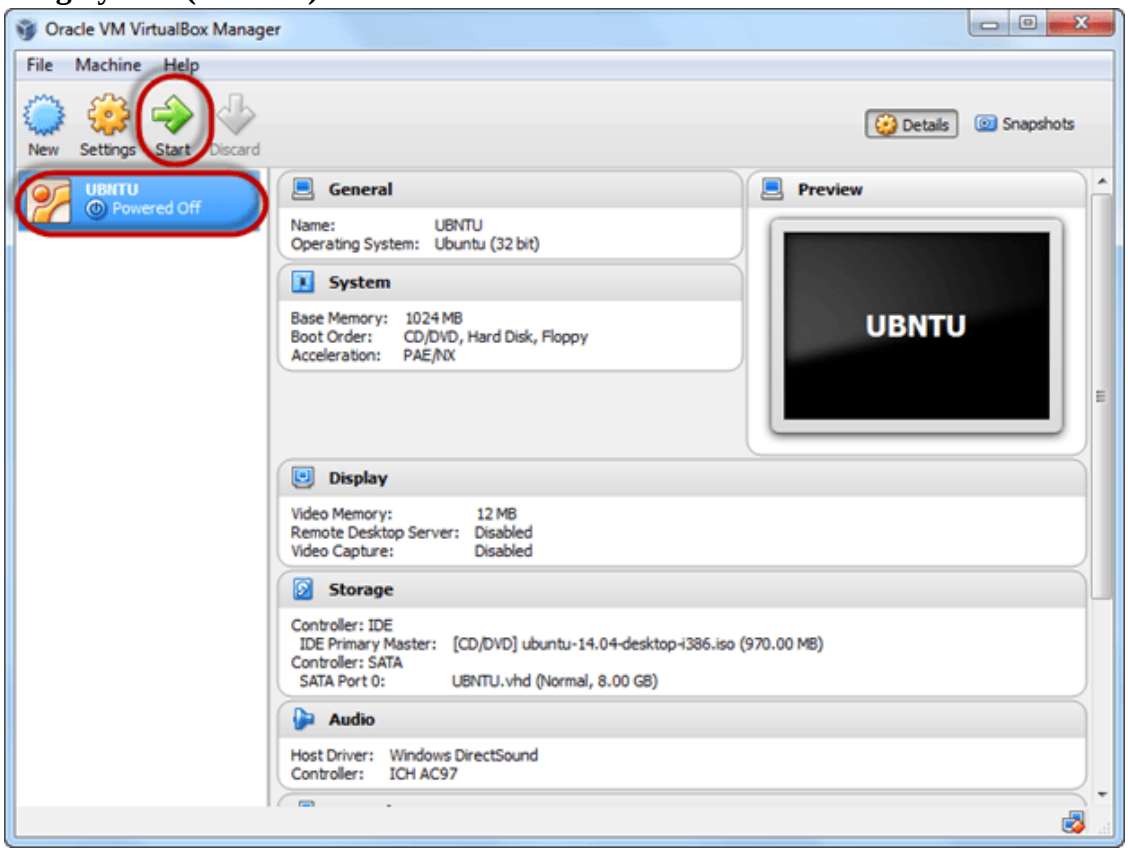


Fig.1.13 start installing ubuntu

Step 13 Select the Folder Option

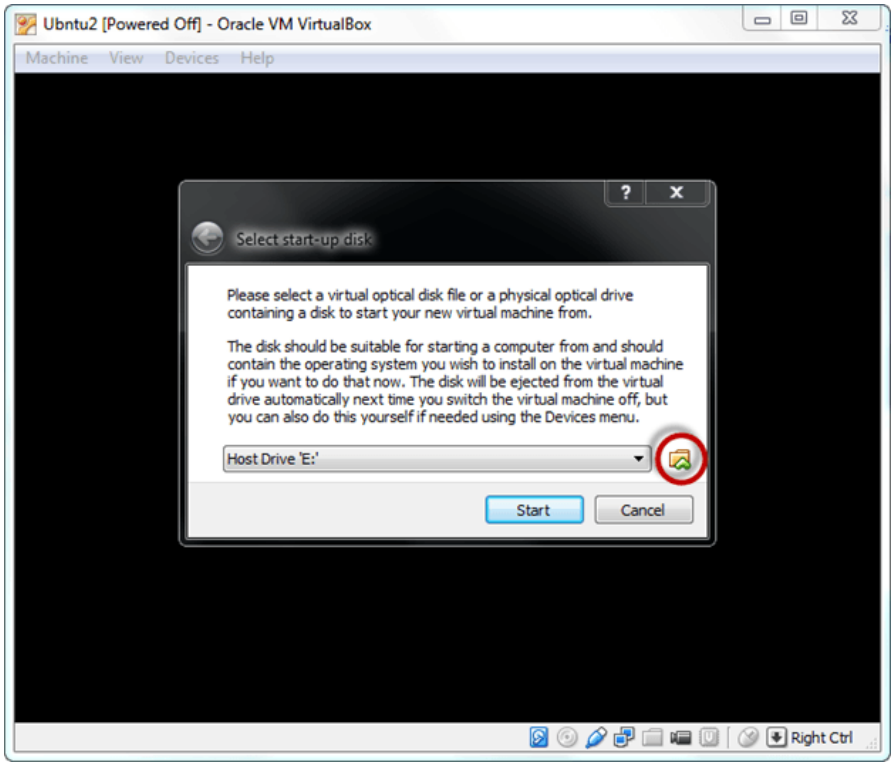


Fig.1.14 selecting source for ubuntu installation

Step 14 Select the Ubuntu iso file

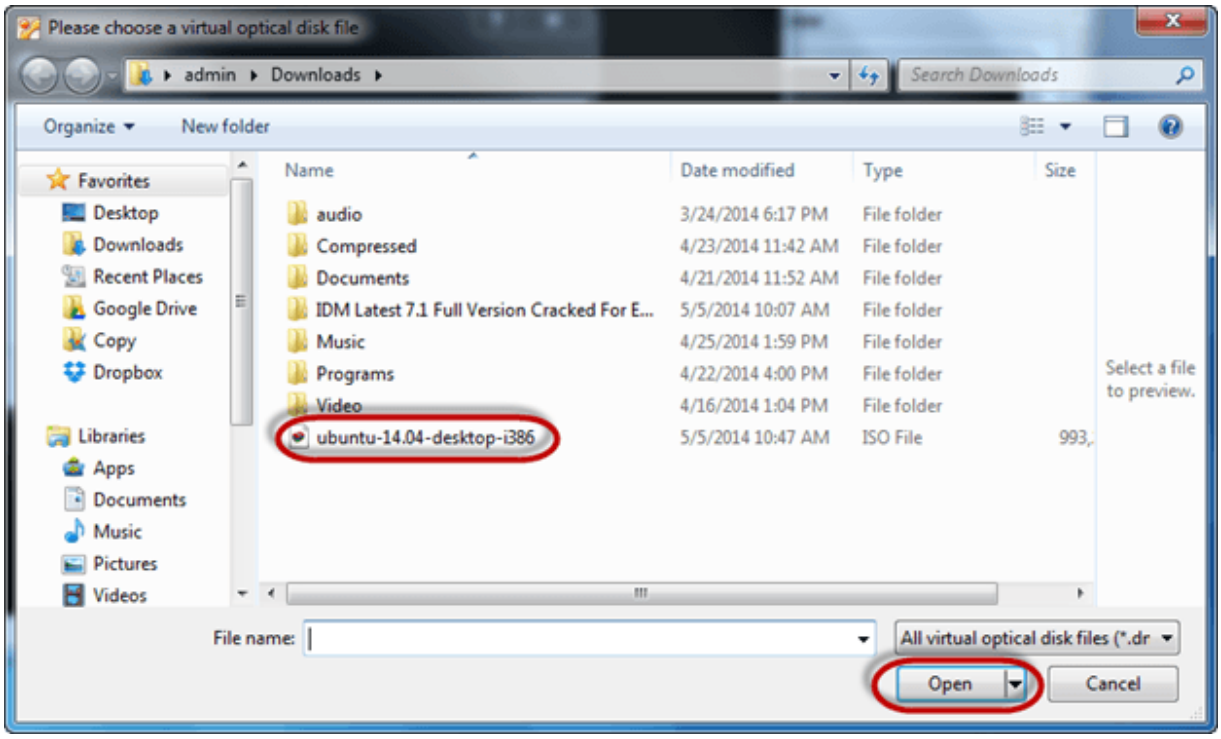


Fig.1.15 selecting source for ubuntu installation

Step 15 Click Start

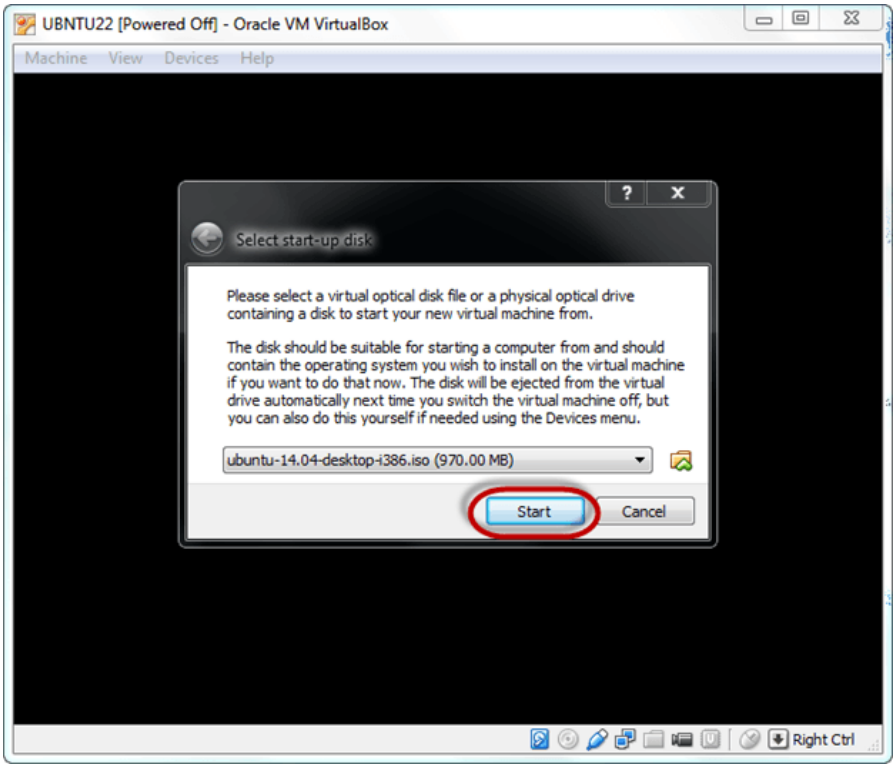


Fig.1.16 selecting source for ubuntu installation

Step-16 You have an option to Run Ubuntu WITHOUT installing. In this tutorial will install Ubuntu

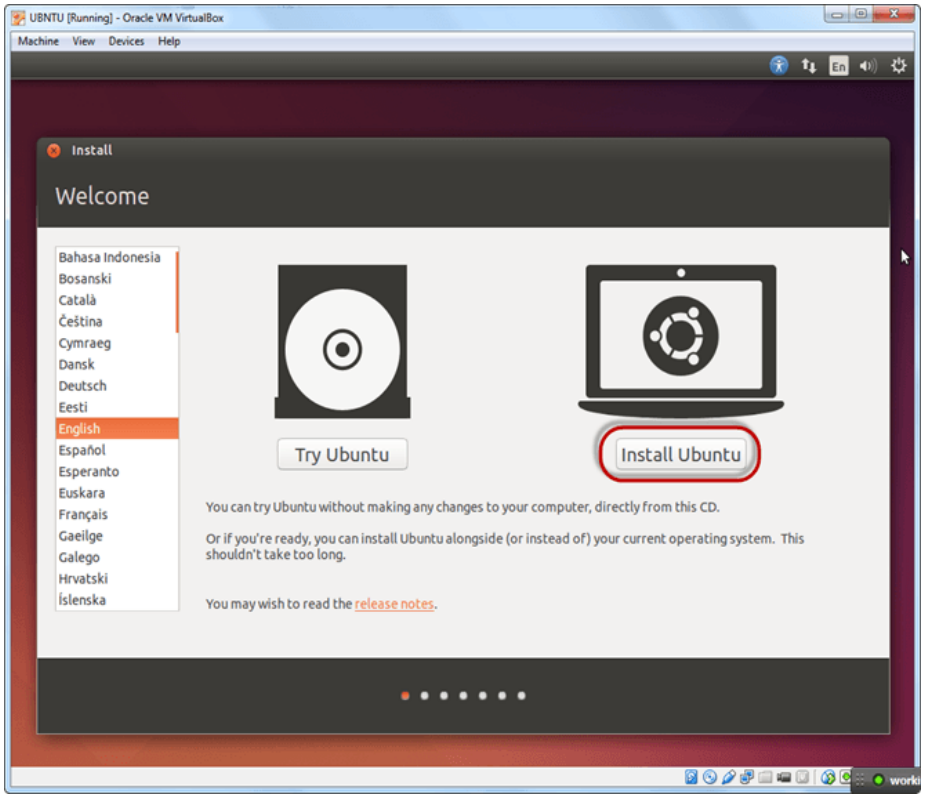


Fig.1.17 start installation

Step-17 Click continue.

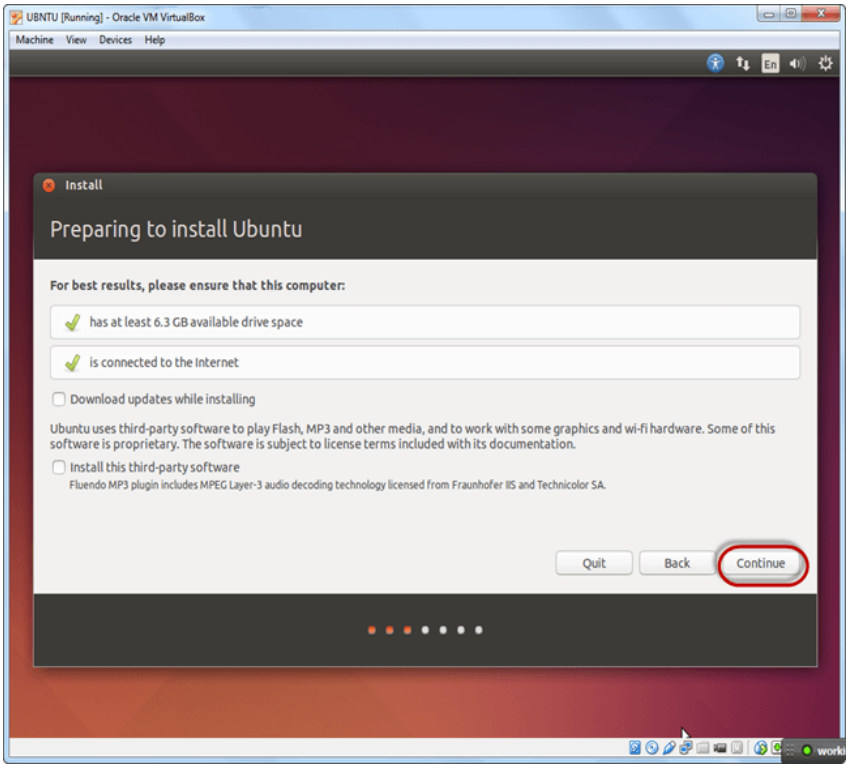


Fig.1.18 Installation started

Step-18 Select option to erase the disk and install Ubuntu and click on install now. This option installs Ubuntu into our virtual hard drive which is we made earlier. It will not harm your PC or Windows installation

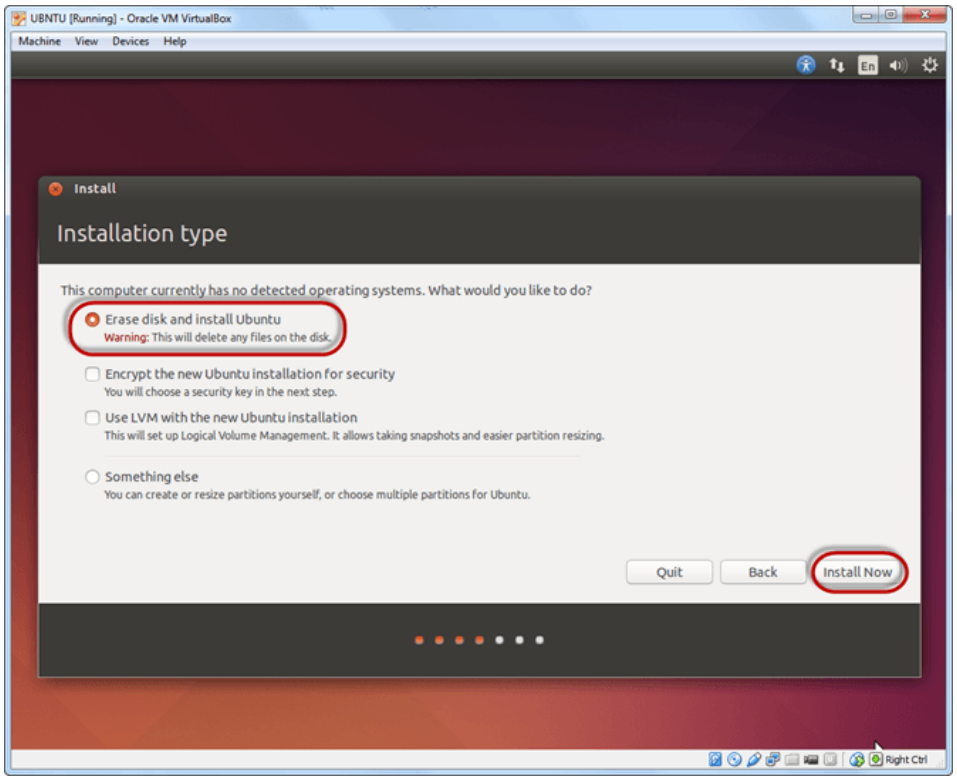


Fig.1.19 Installation started

Step-19 Select your location for setting up time zone, and click on continue

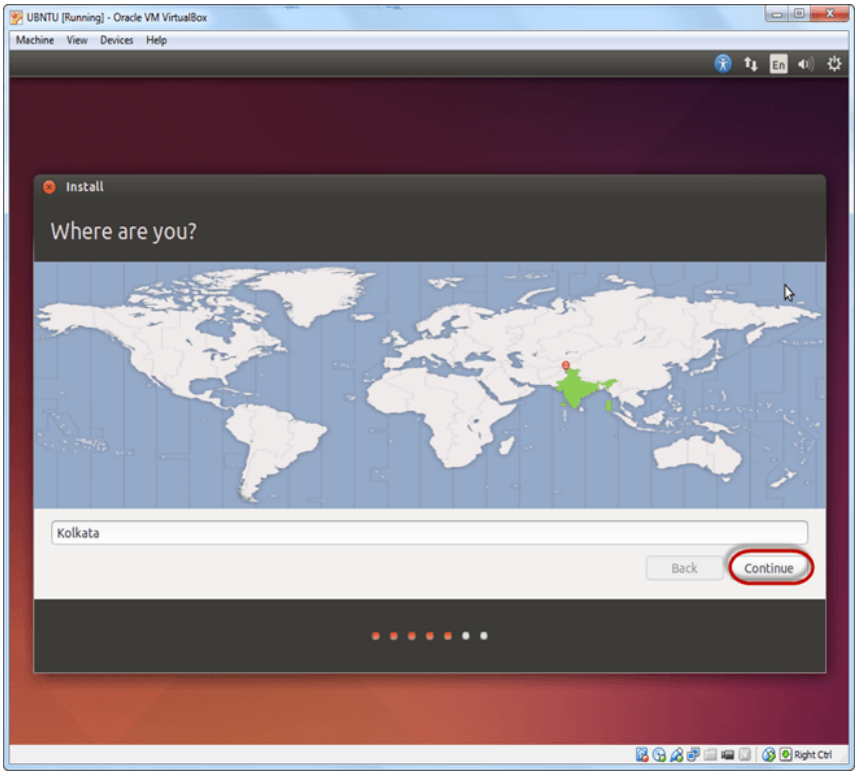


Fig.1.20 setting up timezone

Step-20 Select your keyboard layout, by default English (US) is selected but if you want to change then, you can select in the list. And click on continue

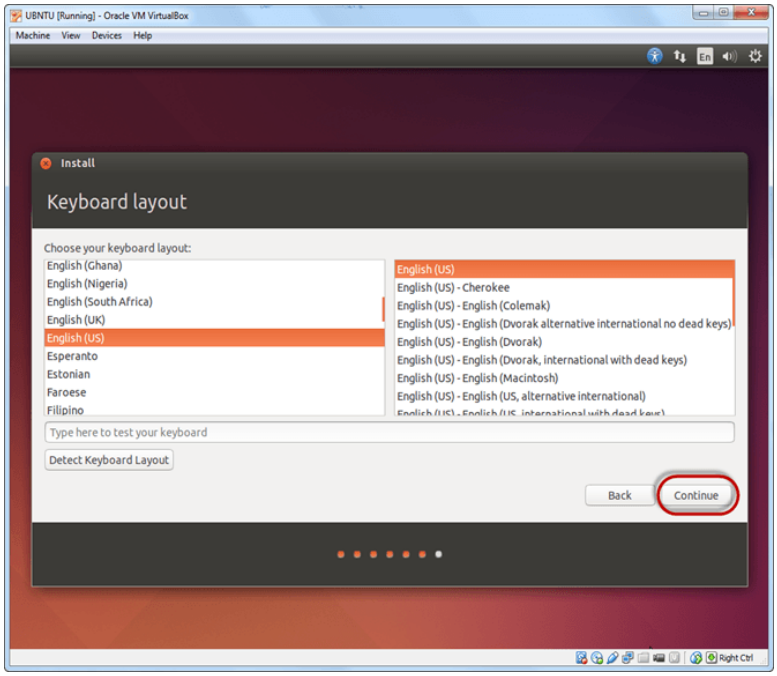


Fig.1.21 setting up preferred Language

Step-21 Select your username and password for your Ubuntu admin account. This information has been needed for installing any software package into Ubuntu and also for login to your OS. Fill up your details and tick on login automatically to ignore login attempt and click on continue

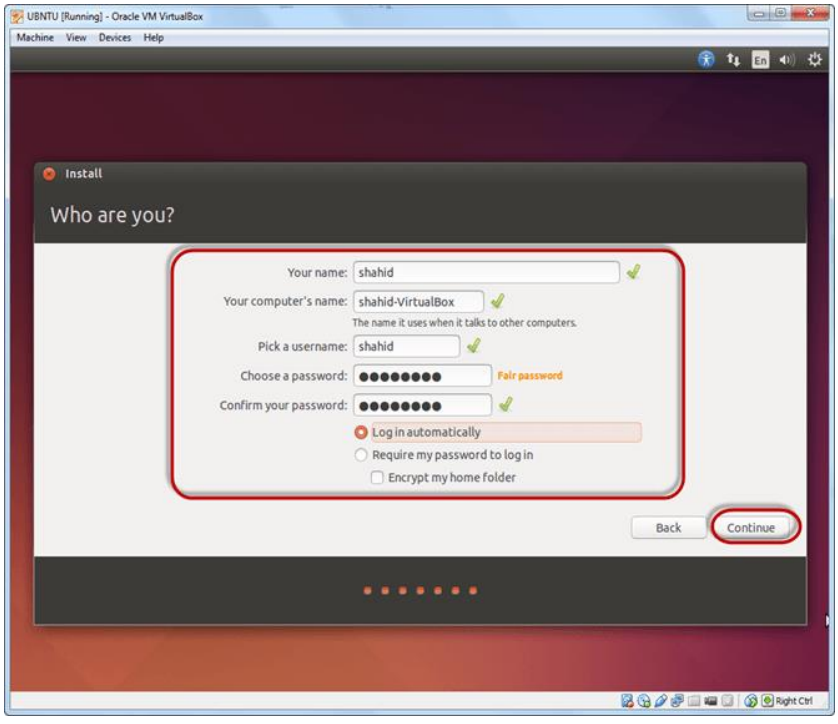


Fig.1.22 setting up User credentials

Step-22 Installation process starts. May take up to 30 minutes. Please wait until installation process completes. After finishing the installation, you will see Ubuntu Desktop.

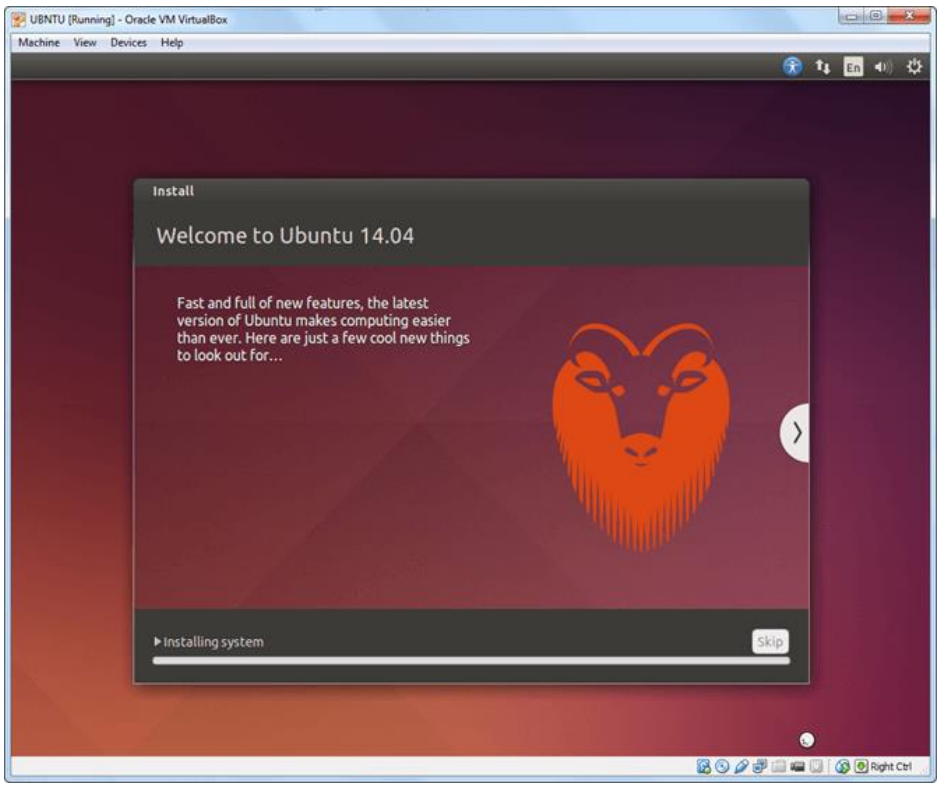


Fig.1.23 Installation in Progess

- **Faculty Member can Assign task to write Installation steps of Windows OS System.**

1.2 Give the advantage of Linux over Windows OS.

//Space for student to write

1.3 Executes Following DOS Commands.

1.ATTRIB 2.CD 3.CHDIR 4.CLS 5.COPY	6. DATE 7. DEL 8. DIR 9. ECHO 10. EXIT	11. FC 12. FIND 13. FINDSTR 14. MKDIR 15. MOVE	16. RENAME 17. REPLACE 18. RMDIR 19. TREE 20. SORT	21. TITLE 22. PRINT 23. HELP 24. TIME 25. VER

Student has to perform above mentioned command on DOS Prompt. And also need to write each command as per given example.

1. **ATTRIB: -** Using the **ATTRIB command**, you can change a file`s read/write attribute or set the archive attribute.

   Syntax:

   ATTRIB [d:][path]filename [/S]

   ATTRIB [+ R|-R] [+A|-A] [+ H|-H] [+ S|-S] [d:][path]filename [/S]

   +R - option to make a file read-only.

   -R - option to change the file protection attribute back to normal (so it can be read, changed, or deleted).

   +A - option to set the ARCHIVE attribute of a file. When the +A option is used, this flags the file as available for archiving when using the BACKUP or XCOPY commands.

   -A - Use the -A option to turn off the ARCHIVE attribute.

   +H - to set the HIDDEN attribute of a file so that it will not appear in a directory listing.

   -H - Use the -H option to turn off the HIDDEN attribute.

   +S - With to set the SYSTEM attribute of a file. When the +S option is used, this flags the file as a command file used only by DOS. The file will not appear in a directory listing. This attribute is generally reserved for programmers.

   -S - Use the -S option to turn off the SYSTEM attribute.

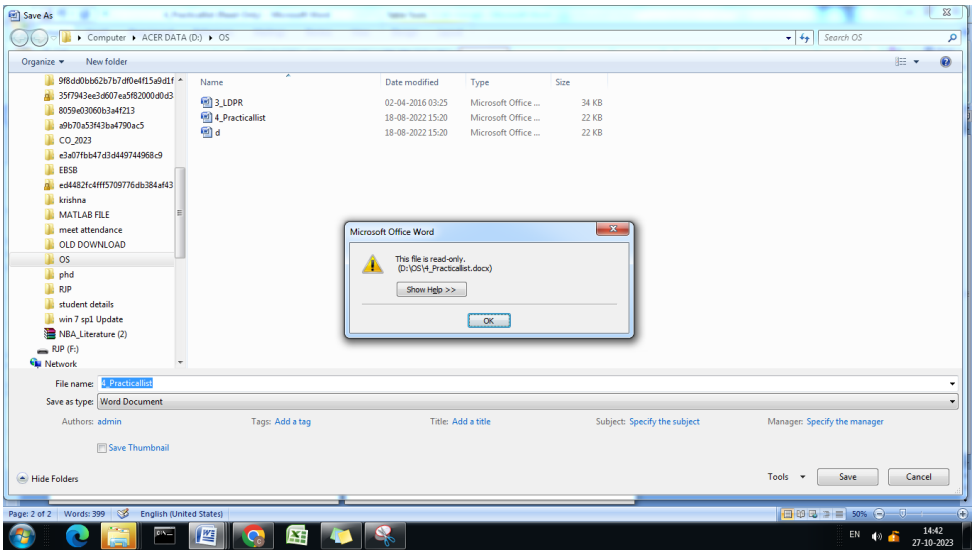
   /S - Use the /S switch to set attributes on subdirectories found within the specified path.

Execution of Command:

```
D:\OS>attrib +R 4_Practicallist.docx

D:\OS>
```

While trying to edit File It Give message “File is Read Only”



Observations:

In this practical we have learned about windows and unbuntu operating systems and their installations. We also learned how to install ubuntu in a virtual box.

Conclusion:

Windows and ubuntu both are different operating systems slightly different than each other and have different target audience.

Quiz:

1. Identify types of OS based on its working.

Ans. Operating systems (OS) can be categorized based on their working and functionality.

Here are the main types of operating systems:

1. Single-User, Single-Tasking OS:

- Designed to support one user and allow them to perform only one task at a time. Classic examples include early versions of MS-DOS.

2. Single-User, Multi-Tasking OS:

Allows a single user to run multiple applications simultaneously. Most modern desktop and laptop operating systems fall into this category, such as Windows, macOS, and Linux.

3. Multi-User OS:

Supports multiple users simultaneously, each with their own independent sessions. Mainframes and some server operating systems, like UNIX and Linux distributions, are examples of multi-user systems.

4. Real-Time OS (RTOS):

Designed to process data and events as they occur in real-time, with strict timing constraints. Commonly used in embedded systems, industrial control systems, and critical applications like medical devices.

5. Multi-Tasking OS:

- Allows multiple tasks or processes to run concurrently on a single system, sharing the CPU's processing time. Most modern operating systems, including Windows, macOS, and Linux, support multitasking.

6. Multi-Processing OS:

Supports the execution of processes on multiple processors or CPU cores. This enhances performance and is common in server and high-performance computing environments.

7. Distributed OS:

Manages a network of independent computers as if they were a single system. It enables distributed computing and is often used in cloud computing environments. Examples include Google's Chrome OS and some versions of Linux.

8. Embedded OS:

Tailored for specific embedded systems and devices, such as smartphones, IoT devices, and embedded control systems. Examples include Android, iOS, and real-time operating systems like FreeRTOS.

9. Network OS:

Primarily designed to manage network resources and facilitate communication between computers. Novell NetWare is an example of a historical network operating system.

10. Mobile OS:

Optimized for mobile devices like smartphones and tablets. Examples include Android, iOS, and Windows Mobile.

11. Desktop OS:

Designed for personal computers and workstations. Examples include Windows, macOS, and various Linux distributions.

12. Server OS:

Optimized for server hardware, focusing on managing and serving resources to other computers on a network. Examples include Windows Server, Linux server distributions, and macOS Server (historical).

These categories highlight the diverse functionalities and use cases of operating systems, catering to the specific needs of different computing environments and devices.

2. Write use of Shell in Linux.

Ans. In Linux, the shell is like a command center where you can give instructions to the computer by typing commands. Here are some practical uses:

1. Running Commands:

You can make the computer do things by typing commands, like opening programs or performing tasks.

2. Scripting:

It lets you automate repetitive tasks by creating scripts, which are sets of commands saved in a file.

3. Managing Files:

You can organize, copy, move, and delete files using commands like `cd`, `ls`, `cp`, `mv`, and `rm`.

4. Redirection:

You can direct the output of a command to a file or use a file's contents as input for a command.

5. Combining Commands:

By using pipes (`|`), you can link commands together to process data in a flexible way.

6. Variables and Environment:

You can use variables to store information and set environment variables that affect how commands behave.

7. User and Permission Control:

You can create and manage user accounts, change file permissions, and control who can

access what.

8. Process and Job Control:

It helps you start, stop, and monitor processes, and manage tasks running in the background.

9. System Configuration:

Administrators use it to adjust system settings, install software, and perform maintenance.

10. Text Manipulation:

You can search for specific text in files (`grep`), transform text (`sed` and `awk`), and manipulate data.

11. Network Tasks:

You can connect to remote servers, transfer files, and check network status using shell commands.

The shell is a powerful tool that lets you control and customize your Linux system using simple text commands.

3. Give the benefits of using Virtualbox?

Ans. VirtualBox is a popular open-source virtualization platform that allows users to run multiple operating systems on a single physical machine. Here are some benefits of using VirtualBox:

1. Multi-Platform Support:

VirtualBox is compatible with various host operating systems, including Windows, macOS, Linux, and others. It also supports a wide range of guest operating systems, making it versatile for different use cases.

2. Cost-Efficiency:

Virtualization enables running multiple virtual machines (VMs) on a single physical machine, optimizing hardware resources and reducing the need for additional physical hardware.

3. Isolation:

Virtual machines are isolated from each other and from the host system. This isolation helps prevent conflicts between different operating systems and applications running on the same physical hardware.

4. Snapshot and Cloning:

VirtualBox allows users to take snapshots of a virtual machine at a specific point in time. This snapshot feature makes it easy to revert to a previous state if something goes wrong. Cloning enables the quick duplication of VMs for testing or backup purposes.

5. Portability:

- Virtual machines can be easily moved and transferred between different VirtualBox installations. This makes it convenient for developers and system administrators to share VM configurations or move VMs between host machines.

6. Resource Allocation:

VirtualBox provides control over resource allocation, allowing users to allocate specific amounts of CPU, memory, and storage to each virtual machine. This flexibility helps optimize performance for different workloads.

7. Networking Features:

VirtualBox offers various networking options, including NAT, Bridged, and Host-Only networking. This allows users to set up different network configurations for virtual machines based on their requirements.

8. USB Device Support:

VirtualBox supports the sharing of USB devices between the host and guest operating systems. This is useful for connecting and using USB peripherals within virtual machines.

9. Wide Range of Guest OS Support:

VirtualBox supports a broad range of guest operating systems, including various versions of Windows, Linux distributions, macOS, and more. This flexibility makes it suitable for testing and development across different platforms.

10. Open Source and Community Support:

VirtualBox is open-source software, and it benefits from an active community of users and developers. This community support ensures regular updates, bug fixes, and a wealth of documentation and resources.

11. Performance Monitoring and Tuning:

Users can monitor the performance of virtual machines and tune settings to optimize resource usage. This allows for better performance and responsiveness in virtualized environments.

VirtualBox's combination of features, flexibility, and ease of use makes it a popular choice for various virtualization needs, including development, testing, and running multiple operating systems on a single physical machine.

Suggested Reference:

1. Operating Systems: Internals & Design Principles, 9th Edition, William Stallings, Pearson Education India
2. Operating System Concepts, 9th edition Peter B. Galvin, Greg Gagne, Abraham Silberschatz, John Wiley & Sons, Inc.
3. Modern Operating Systems-By Andrew S. Tanenbaum (PHI)
4. UNIX : Concepts and Applications | 4th Edition by Sumitabha Das ,McGrawHill

5. https://ubuntu.com/tutorials/how-to-run-ubuntu-desktop-on-a-virtual-machine-using-virtualbox#1-overview

References used by the students:

// Write references used by you here

Rubric-wise marks obtained:

Rubrics	**Understanding Related to OS/Computer System (4)**			**Command Execution (4)**			**Documentation &Timely Submission (2)**			**Total (10)**
	Good (4)	**Avg. (3-2)**	**Poor (1-0)**	**Good (4)**	**Avg. (3-2)**	**Poor (1-0)**	**Good (2)**	**Avg. (1)**	**Poor (0)**	
Marks										

Experiment No: 2

AIM : Study and execute Basic and directory manipulation commands of LINUX/UNIX.

2.1. Study of Unix Shell and Environment Variables.

2.2. man, cal, date, echo, bc, who, uname

2.3. Using commands : pwd, mkdir, cd, rmdir ,ls generate given tree.

Date: ***// Write date of experiment here***

Competency and Practical Skills: Basic Skills to work with Computer System/ Linux Terminal

Relevant CO: , CO5

Objectives:

a. To understand the importance of Shell/Envirnment Variable

b. To Work with basic Commands.

c. To work/Access the Directory commands.

Equipment/Instruments: Computer System with Linux OS.

Safety and necessary Precautions:

✓ Operate computer system carefully and responsibly.

✓ Use required lab resources cautiously

Theory:

A Unix shell is a command-line interpreter or shell that provides a command line user interface for Unix-like operating systems. The shell is both an interactive command language and a scripting language, and is used by the operating system to control the execution of the system using shell scripts.

Environment variables basically define the behavior of the environment. They can affect the running processes or programs executed in the environment. Every Linux process has an associated set of environment environment variables are typically accessed through the shell. The shell is a command-line interface that interprets and executes commands entered by the user. It provides a way to set, modify, and retrieve environment variables.just like programming language variables The scope of any variable is the region from which it can be accessed or over which it is defined. An environment variable in Linux can have global or local scope.

There are Large set of Commands supported by Linux/Unix. In this Practical we will study Basic

Basic Linux and Directory commands

Command Name	Description
man	It is used to open help manual for any linux command
cal	It displays the current month calendar (System).
date	It displays current date (System).
echo	It prints the string as provided by user.
bc	It is a command line calculator used to do basic mathematical calculations.
who	It prints information about users who are currently logged in.
uname	It displays the information about the system.
pwd	It displays name of present working directory.
mkdir	It creates new directory.
cd	It is used to move from one directory to another.
rmdir	It is used remove the directory.
ls	Listing the files and directory.

For any command, help manual is available in Linux System. We can use **man** Command to open help manual for any command using following command:

$man command name

- **Based on Questions asked, student has to execute command and need to write the answer.**
- **Before using the command to solve the questions, student has to write name of the command, Syntax, options available to use the command and description(as per requirement).**

Example:

Command Name: date (print or set the system date and time**)**

Syntax: date [OPTION]... [+FORMAT]

date [-u|--utc|--universal] [MMDDhhmm[[CC]YY][.ss]]

OPTIONS

%a locale's abbreviated weekday name (e.g., Sun)

%A locale's full weekday name (e.g., Sunday)

%b locale's abbreviated month name (e.g., Jan)

%B locale's full month name (e.g., January)

%c locale's date and time (e.g., Thu Mar 3 23:05:25 2005)

%C century; like %Y, except omit last two digits (e.g., 20)

%d day of month (e.g., 01)

%D date; same as %m/%d/%y

%e day of month, space padded; same as %_d

%F full date; same as %Y-%m-%d

%g last two digits of year of ISO week number (see %G)

%G year of ISO week number (see %V); normally useful only with %V

%h same as %b

%H hour (00..23)

Note: There are number of option available to work with specific command student can write description of any option as per requirement of practical.

- **To Write the Answer of Question Example is given below.**

Example.

Command Name: Shell

Syntax: $Shell

Output:

```
$ echo $SHELL
/bin/bash
```

2.1. Study of Unix Shell and Environment Variables.

2.1.1. Display The name of the current shell of your System.

```
admin@Atri:~$ echo $SHELL
/bin/bash
admin@Atri:~$
```

2.1.2. Write the different shell names available in Linux.

```
admin@Atri:~$ cat /etc/shells
# /etc/shells: valid login shells
/bin/sh
/bin/bash
/usr/bin/bash
/bin/rbash
/usr/bin/rbash
/usr/bin/sh
/bin/dash
/usr/bin/dash
admin@Atri:~$
```

2.1.3. Display the name of Enviornment Variable.

```
admin@Atri:~$ env
SHELL=/bin/bash
SESSION_MANAGER=local/Atri:@/tmp/.ICE-unix/1407,unix/Atri:/tmp/.ICE-unix/1407
QT_ACCESSIBILITY=1
COLORTERM=truecolor
XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:/etc/xdg
SSH_AGENT_LAUNCHER=gnome-keyring
XDG_MENU_PREFIX=gnome-
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
LANGUAGE=en_IN:en
GNOME_SHELL_SESSION_MODE=ubuntu
SSH_AUTH_SOCK=/run/user/1001/keyring/ssh
XMODIFIERS=@im=ibus
DESKTOP_SESSION=ubuntu
GTK_MODULES=gail:atk-bridge
PWD=/home/admin
XDG_SESSION_DESKTOP=ubuntu
LOGNAME=admin
XDG_SESSION_TYPE=wayland
SYSTEMD_EXEC_PID=1434
XAUTHORITY=/run/user/1001/.mutter-Xwaylandauth.EMAQK2
HOME=/home/admin
USERNAME=admin
IM_CONFIG_PHASE=1
```

2.2. cal, date, echo, bc, who, uname

2.2.1. Display current month Calendar

```
admin@Atri:~$ cal
     March 2024
Su Mo Tu We Th Fr Sa
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31
```

2.2.2. Display the Calendar of given month and year value provided by user.

```
admin@Atri:~$ cal 6 2016
      June 2016
Su Mo Tu We Th Fr Sa
            1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30
```

2.2.3. Display the calendar of current, previous and next month.

```
admin@Atri:~$ cal -3
   February 2024          March 2024           April 2024
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
                1  2  3                  1  2      1  2  3  4  5  6
 4  5  6  7  8  9 10   3  4  5  6  7  8  9   7  8  9 10 11 12 13
11 12 13 14 15 16 17  10 11 12 13 14 15 16  14 15 16 17 18 19 20
18 19 20 21 22 23 24  17 18 19 20 21 22 23  21 22 23 24 25 26 27
25 26 27 28 29        24 25 26 27 28 29 30  28 29 30
                      31
```

2.2.4. Display a calendar in which start day of week is Monday.

```
admin@Atri:~$ ncal -M -b
     March 2024
Mo Tu We Th Fr Sa Su
                1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31
```

2.2.5. Display a calendar in which start day of week is Sunday.

```
admin@Atri:~$ ncal -S -b
     March 2024
Su Mo Tu We Th Fr Sa
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31
```

2.2.6. Display a calendar in which current date is not highlighted.

```
admin@Atri:~$ cal
     March 2024
Su Mo Tu We Th Fr Sa
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31
```

2.2.7. Display current month number using DATE command.

```
admin@Atri:~$ date +%-m
3
```

2.2.8. Display current year number in two digit form using DATE command.

```
admin@Atri:~$ date +'%y'
24
```

2.2.9. Display the day of current date in short as well as full name using DATE command.

```
admin@Atri:~$ date +'%a'
Wed
admin@Atri:~$ date +'%A'
Wednesday
```

2.2.10. Display the month of current date in short as well as full name using DATE command.

```
admin@Atri:~$ date +'%b
Mar
admin@Atri:~$ date +'%B
March
```

2.2.11. Display date in mm/dd/yyyy format.

```
admin@Atri:~$ date +'%m/%d/%Y'
03/13/2024
```

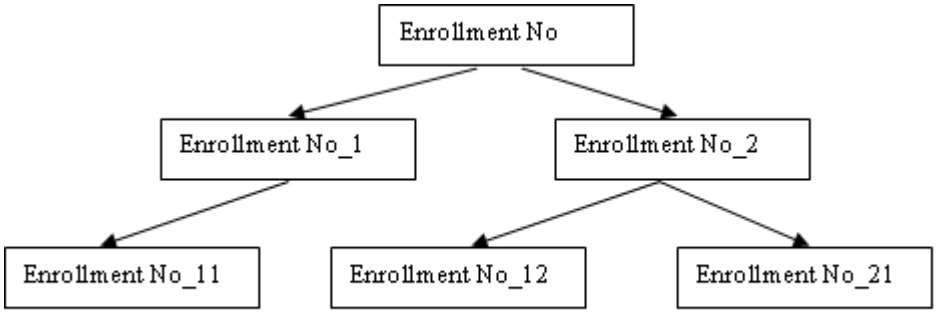
2.2.12. Display date in yyyy-mm-dd format.

```
admin@Atri:~$ date +'%Y/%d/%m'
2024/13/03
```

2.2.13. Display the century of current year.

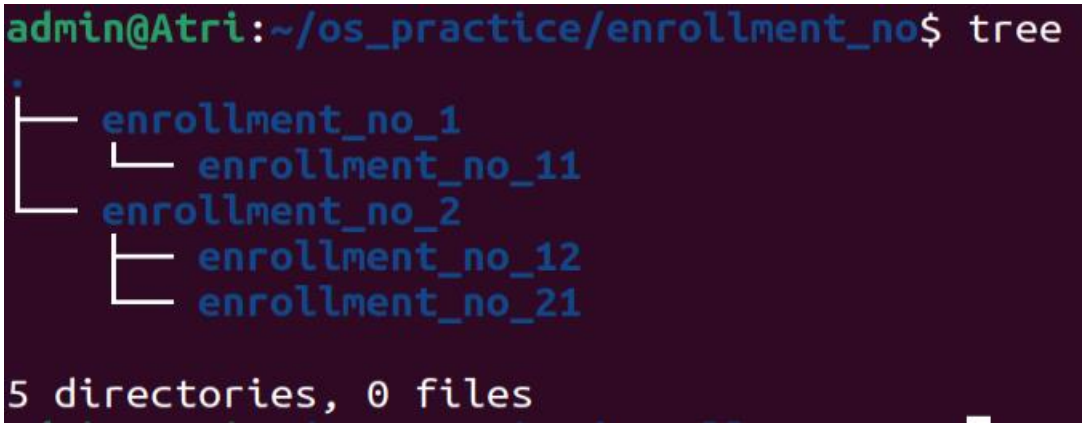
```
admin@Atri:~$ date +'%C'
20
```

2.3. Using pwd, mkdir, cd, rmdir,ls commands generate given tree.



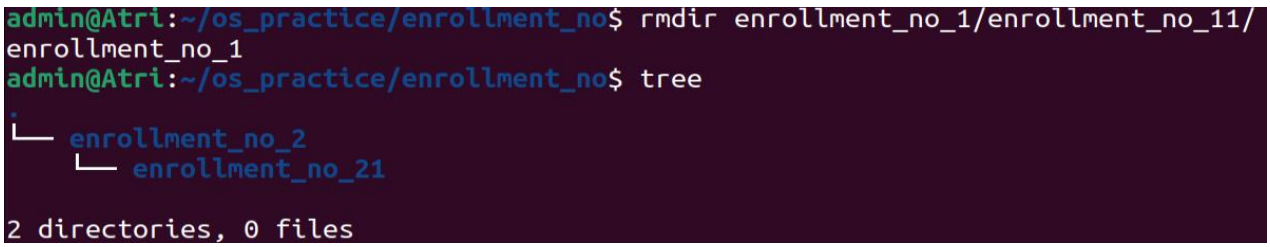
Student has to create Each directory with his/her enrollment number.

1.1.1. Display the path from root directory to the last level directory of the tree.(Consider Root directory of given tree as root)



1.1.2. Remove Directory Enrollment_12.

1.1.3. Remove Directore Enrollment_1.



Observations:

// Write your observation here

Conclusion:

// Write conclusion here

Quiz:

1. **Give Significance of Shell in Linus OS.**

2. **How to move in/out from directory using cd in single step.**

3. **Write Use of bc Command.**

Suggested Reference:

1. Operating Systems: Internals & Design Principles, 9th Edition, William Stallings, Pearson Education India

2. Operating System Concepts, 9th edition Peter B. Galvin, Greg Gagne, Abraham Silberschatz, John Wiley & Sons, Inc.
3. Modern Operating Systems-By Andrew S. Tanenbaum (PHI)
4. UNIX : Concepts and Applications | 4th Edition by Sumitabha Das ,McGrawHill

References used by the students:

// Write references used by you here

Rubric-wise marks obtained:

Rubrics	**Understanding of commands (4)**			**Ability to use Command for question solving (4)**			**Documentation &Timely Submission (2)**			**Total (10)**
	Good (4)	**Avg. (3-2)**	**Poor (1-0)**	**Good (4)**	**Avg. (3-2)**	**Poor (1-0)**	**Good (2)**	**Avg. (1)**	**Poor (0)**	
Marks										

Experiment No: 3

AIM : Study and execute Basic File manipulation commands.
1. cat 2.wc 3. cp 4. mv 5.rm 6. File 7. cmp 8. comm 9. diff 10. chmod 11. sort

Date: ***// Write date of experiment here***

Competency and Practical Skills: Basic Skills to work with Computer System/ Linux Terminal

Relevant CO: , CO5

Objectives:
a. To understand the importance of Shell/Envirnment Variable
b. To work with basic File Operations.
c. To work/Access the Directory commands.

Equipment/Instruments: Computer System with Linux OS.

Safety and necessary Precautions:

- ✓ Operate computer system carefully and responsibly.
- ✓ Use required lab resources cautiously

Theory:
File manipulation commands are mainly used for Operations like create File, Copy File, Delete File, Rejname file, Searching No. of counts from file, finding Common values from two file, Comparing to files, finding difference between two file to make them identical. There is also File Permission command chmod to change the fillle permission.

Basic File commands

Command Name	Description
cat	It redirects standard output to/from the file.
wc	It counts word, characters and lines from the file.
cp	Copy one file to another.
mv	Rename or move file from one directory to another.
rm	Remove the file from directory.

file	Displays file types and other details.
cmp	Compare the content of two files.
comm.	Find common between two files.
diff	Find the difference between two files and give way to make them identical.
chmod	It changes file permission.
sort	It sorts the file content.

3.1. Create 4 Files using cat Command.

- **Create 4 Files F1.txt ,F2.txt ,F3.txt ,F4.txt with some content.**
- **Preferable to put paragraph in one file.**
- **In File F2 and F3 write content in form of words in alphabetical sorted order**
  Also put some common value in both file (like student name or engineering branch names, fruits, vegetables, etc)
- **In fourth File write decimal numbers separated by newline.**

3.2. Copy Content of File F1 to F1_c.

3.3. Rename File F1_c to F5.

3.4. Compare File F1 and F5

3.5. Move file F5 to another Directory of your Choice.

3.6. Count no.of characters,words and Lines of F1 file.

3.7. Find Common Value between File F2 and F3.

3.8. Find the Difference Between File F2 and F3.

3.9. Change the F1 file permission to read Only.

3.10. Change F2 permission to Read and Write Only.

3.11. Change File F4 permission to Read,Write and Execute .

3.12. Perform sort command on F4.

3.13. Perform numeric Sort on File F4.

Observations:

// Write your observation for comm., cmp, diff, sort, chmod command.

Conclusion:

// Write conclusion here

Quiz:

1. Why there is need to change file permission.

2. How to Change File permission in different way using chmod command.

3. Write the use of chown command.

4. Write use of " |" character in Linux.

Suggested Reference:

1. Operating Systems: Internals & Design Principles, 9th Edition, William Stallings, Pearson Education India
2. Operating System Concepts, 9th edition Peter B. Galvin, Greg Gagne, Abraham Silberschatz, John Wiley & Sons, Inc.
3. Modern Operating Systems-By Andrew S. Tanenbaum (PHI)
4. UNIX : Concepts and Applications | 4th Edition by Sumitabha Das ,McGrawHill

References used by the students:

// Write references used by you here

Rubric-wise marks obtained:

Rubrics	**Understanding of commands (4)**			**Ability to use Command for question solving (4)**			**Documentation &Timely Submission (2)**			**Total (10)**
	Good (4)	**Avg. (3-2)**	**Poor (1-0)**	**Good (4)**	**Avg. (3-2)**	**Poor (1-0)**	**Good (2)**	**Avg. (1)**	**Poor (0)**	
Marks										

Experiment No: 4

AIM : Study and Execute Advance Filter Commands.

1.head 2. tail 3. paste 4. cut(-f) 5. cut(-c) 6. grep

Date: ***// Write date of experiment here***

Competency and Practical Skills: Basic Skills to work with Computer System/ Linux Terminal

Relevant CO: , CO5

Objectives:

a. To understand the importance of File filter commands.

b. To merge, split and Search in different way from the file.

Equipment/Instruments: Computer System with Linux OS.

Safety and necessary Precautions:

✓ Operate computer system carefully and responsibly.

✓ Use required lab resources cautiously

Theory:

Filter commands accept input data from standard input and produce output standard output. It transforms plain-text data into a meaningful way and can be used merge with other output or file. These filters are very small programs that are designed for a specific function which can be used as building blocks.There are number of commands we already have covered I n previous practicals like sort,comm.,cat and others are cut , paste, head,tail, grep, tee ,uniq, grep. Using these commands one can search and display specif content from file also filter some specific data.

For more description about the command we can read a Linux Help manual using command :

$man Commandname.

Command Name	Description
head	It displays first 10 lines from the input file.
tail	It displays last 10 lines from the file.
cut	It cut the file content Vertically(-f) as well as Horizontally(-c).
paste	It pastes content of different files and displays the output.
grep	Search the line with specified pattern present in the file(s).

grep command searches the simple pattern as well pattern specified using Regular expression.To create a pattern which matches different types of string we can use different wildcard characters to create a new pattern. The basic set of wildcards in are:

***** – This wildcard represents all the characters. Also represent one or more occurrence of preceded character.

+ - represent one or more occurrence of preceded character.

? – This wildcard represents a single character

[ ] – This wildcard represents a range of characters.

To solve the question given student has to create a text file with Student details like:

Enrollment number, Student name, Birth-date, Semester, Gender , Email Address, SPI ,where each field is separated by delimiter character "|" or any character as per wish.

```
140**0107048 |khushboo singh        | 4th | 12/6/199     | khushi1245@gmail.com      | Female | 7.73
140**0107049 |nimisha  sinh         | 2nd | 12/10/1996   | nimesh@gmail.com          | Female | 6.75
140**0107050 |urvish  patil         | 4th | 7/8/1996     | urvi@gmail.com            | Male   | 8.3
140**0107043 |daksh                 | 6th | 4/7/1996     | daksh_softskill@gmail.com | Male   | 5.65
140**0107046 |John Desoza           | 4th | 25/6/1996    | john_d@gamil.com          | Male   | 7.5
140**0107054 |Pooja Patil           | 4th | 17/9/1996    | puja_1867@gmail.com       | Female | 7.9
140**0107052 |niti  Patel           | 4th | 1/3/1996     | niti_p@gmail.com          | Female | 7.12
140**0107042 |dhara  Parekh         | 6th | 16/7/1996    | dhara@gmail.com           | Female | 7.5
140**0107052 |Mohan  Bharadwaj      | 4th | 12/6/1996    | a_mohan@gmail.com         | Male   | 7.5
150**3131002 |Raj  agrawal          | 2th | 24/12/1994   | raja@gmail.com            | Male   | 8.2
150**3131004 |vipa  agarwal         | 4th | 1/8/1997     | vipa@gmail.com            | Female | 7.34
140**0107041 |falgun  patel         | 4th | 3/4/1996     | falgun@gmail.com          | Male   | 6.87
140**0107047 |mahim  mishra         | 4th | 16/4/1996    | mahim@gmail.com           | Male   | 7.80
140**0107045 |vijeta  aggrawal      | 4th | 4/5/1996     | vijeta@1976.com           | Female | 7.12
140**0107055 |chiranjita  Chaturvedi| 4th | 23/3/1996    | chiru@gmail.com           | Female | 6.34
140**0107101 |mahesh  dwivedi       | 6th | 16/4/1996    | Mahesh_d@gmail.com        | Male   | 7.54
140**0107110 |vijay  patel          | 4th | 4/5/1996     | vijey_p@1996.com          | Male   | 7.67
140**0107148 |chintal parikh        | 2th | 23/3/1996    |chinu@gmail.com            | Female | 6.68
```

4.1. Display first 7 lines of the file.

4.2. Display Last 4 Lines of the File.

4.3. Cut the file column wise and display Student Enrollment number, Gender, Email Address.

4.4. Cut the file column wise and display Student name , Branch,semester.

4.5. Merge the result of Question 4.3 and 4.4 using delimiter $ and store it File named result1 using tee command.

4.6. Cut the File Fieldwise (vertically) to display second, third and fourth field of the file.

4.7. Display the First three student details having highest SPI in decreasing order.

4.8. Change file delimiter "|" with another "* ".

4.9. Display student details studying in same branch.

4.10. Display student details studying in same branch and same semester.

4.11. Display students have SPI greater than 6.

4.12. Display student details studying Whose surname starts will “p” and ends with “l”.

4.13. Display the name of student whose surname is agarwal (surname may be in any form).

4.14. Display details of all girls students.

Observations:

// Write your observation here

Conclusion:

// Write conclusion here

Quiz:

1. What is the use of tee and cat command.
2. By default, how many lines are displayed using the head command? which option used to display specific lines from the file?
3. Which option is used to display file content in reverse order?

Suggested Reference:

1. Operating Systems: Internals & Design Principles, 9th Edition, William Stallings, Pearson Education India
2. Operating System Concepts, 9th edition Peter B. Galvin, Greg Gagne, Abraham Silberschatz, John Wiley & Sons, Inc.
3. Modern Operating Systems-By Andrew S. Tanenbaum (PHI)
4. UNIX : Concepts and Applications | 4th Edition by Sumitabha Das ,McGrawHill

References used by the students:

// Write references used by you here

Rubric-wise marks obtained:

Rubrics	**Understanding of commands (4)**			**Ability to use Command for question solving (4)**			**Documentation &Timely Submission (2)**			**Total (10)**
	Good (4)	**Avg. (3-2)**	**Poor (1-0)**	**Good (4)**	**Avg. (3-2)**	**Poor (1-0)**	**Good (2)**	**Avg. (1)**	**Poor (0)**	
Marks										

Experiment No: 5

AIM : Write a shell script program using Loop/ control structure.

5.1. Write a shell script to find factorial of given number n

5.2. Write a shell script which will generate first n fibonnacci numbers like: 1, 1, 2, 3, 5, 13,…

Date: ***// Write date of experiment here***

Competency and Practical Skills: Logic building and programming

Relevant CO: CO5

Objectives:

a. To understand and use the loop and control structure to sove problem using shell script.

Equipment/Instruments: Computer System with Linux OS.

Safety and necessary Precautions:

✓ Operate computer system carefully and responsibly.

✓ Use required lab resources cautiously

Theory:

Linux/Unix shells are interactive, by means they accept commands as input from users and execute them and display the output accordingly. Normally we are executing command independently one by one. Sometimes it is require to execute same no.of commands repeatedly which is time consuming task but linux system also has solution for that called shell programming. For that we can put all commands together in single file and execute them in shell to avoid repetitive work. These files are called **Shell Scripts** or **Shell Programs**. Shell scripts are similar to the batch file in MS-DOS. The shell script file is saved with **".sh"** extension e.g., **First_program.sh.**

A shell script has syntax like other programming language. If you have any prior experience of programming language like Python, C/C++ etc. It would be very easy to understand shell programming. It has it's Shell Keywords, Control flow statements, Loop statements, Shell (linux command we had used in previous experiments)and Functions.

To run the shell script program, file must be executable. To set execute permission of file we can use chmod command.e.g**. chmod 777 file.sh** command allows file.sh to execute.to run the file one can write the command **./file.sh.**

- **There are 3 types loop statements supported by shell programming:**

1. **while statement**

Syntax:

```
while <condition>
do
   <command statement 1>
   <command statement 2>
    .
    .
   <command statement n>
done
```

2. **for statement**

The for loop operates on lists . It repeats a set of commands for every item as per list value. var is the variablename and var takes value from the list value1, value2, … value n on each iteration, respectively,

Syntax:

```
for <var> in <value 1 value 2 ... value n>
do
   <command statement 1>
   <command statement 2>
    .
    .
   <command statement n>
done
```

3. **do…until statement**

The do… until loop is executed as many times as th condition/command evaluates too false. The loop terminates when the condition/command becomes true.

Syntax:

```
until <condition>
do
   <command statement 1>
   <command statement 2>
.
.
done
```

To change the flow of loop statements, two commands are used they are,

1. break
2. continue

- **There are basically 2 types of contro statement supported in shell programming:**

1. **if –else statement (different versions)**

   i. **Simple if statement**

   Syntax:

```
if [ expression ]
then
        <command statement 1>
fi
```

   ii. **if-else statement**

   Syntax:

```
if [ expression ]
then
   <command statement 1>
else
        <command statement 2>
fi
```

iii. if..elif..else..fi statement (Else If ladder)

Syntax:

```
if [ expression1 ]
then
   <command statement 1>
   <command statement 2>
elif [ expression2 ]
then
   <command statement 3>
   <command statement 4>
  .
else
   <command statement 5>
fi
```

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

Syntax:

2. switch statement

Syntax:

```
case “expression” in
        Pattern 1) < command Statement 1> ;;
        Pattern 2) < command Statement 2> ;;
        .
        .
        .
        Pattern n) < command Statement n> ;;
        esac
```

Example1 : "Checking whether two numbers are equal or not?"

```
a=20
b=21
if [ $a -gt $b ]
then
                #If they are equal then print this
                echo "a is greater than  b"
else
                #else print this
                echo "a is less than  b"
fi
```

OUTPUT:

```
a is less than  b
```

Example 2: "shell script to Display branchcode of respective branch using case control statement"

```
echo "Enter department name"
read DEPARTMENT
case $DEPARTMENT in
  "CE")  echo  " branch code:07" ;;
  "EC")  echo  " branch code:11" ;;
  "CIVIL")  echo  " branch code:06" ;;
  "IT")  echo  " branch code:16" ;;
  "MECH")  echo  " branch code:19" ;;
  *)   echo -n "Invalid"
    ;;
esac
```

OUTPUT:

```
Enter department name
CE
branch code:07
```

2.1. Write a shell script to find factorial of given number n.

Program:

```
#!/bin/bash

factorial() {
   if [ $1 -eq 0 ] || [ $1 -eq 1 ]; then
      echo 1
   else
      echo $(( $1 * $(factorial $(( $1 - 1 ))) ))
   fi
}

if [ $# -eq 0 ]; then
   echo "Usage: $0 <number>"
   exit 1
fi

number=$1

if ! [[ "$number" =~ ^[0-9]+$ ]]; then
   echo "Error: Please provide a non-negative integer."
   exit 1
fi

result=$(factorial $number)
echo "Factorial of $number is: $result"
```

Output:

```
Factorial of 6 is: 720
```

2.2. Write a shell script which will generate first n fibonnacci numbers like: 1, 1, 2, 3, 5, 13,…

Program:

```
#!/bin/bash
```

```
generate_fibonacci() {
    n=$1
    a=1
    b=1

    echo "Fibonacci series for the first $n numbers:"

    for ((i = 1; i <= n; i++)); do
        echo -n "$a "

        # Calculate the next Fibonacci number
        next=$((a + b))
        a=$b
        b=$next
    done

    echo
}

if [ $# -eq 0 ]; then
    echo "Usage: $0 <number>"
    exit 1
fi

number=$1

if ! [[ "$number" =~ ^[1-9][0-9]*$ ]]; then
    echo "Error: Please provide a positive integer."
    exit 1
fi
```

Output:

```
Fibonacci series of first 8 numbers:
1 1 2 3 5 8 13 21
```

Observations:

// Write your observation here

Conclusion:

// Write conclusion here

Quiz:

1. How to read and print the value of variable in shell script?

2. Write syntax to retrieve variable value using shell script.

3. Write a step to run shell script.

Suggested Reference:

1. Operating Systems: Internals & Design Principles, 9th Edition, William Stallings, Pearson Education India
2. Operating System Concepts, 9th edition Peter B. Galvin, Greg Gagne, Abraham Silberschatz, John Wiley & Sons, Inc.
3. Modern Operating Systems-By Andrew S. Tanenbaum (PHI)
4. UNIX : Concepts and Applications | 4th Edition by Sumitabha Das ,McGrawHill

References used by the students:

// Write references used by you here

Rubric-wise marks obtained:

Rubrics	**Understanding of Shell programming syntax (4)**			**Ability to implement program for given problem using Shell script(4)**			**Documentation &Timely Submission (2)**			**Total (10)**
	Good (4)	**Avg. (3-2)**	**Poor (1-0)**	**Good (4)**	**Avg. (3-2)**	**Poor (1-0)**	**Good (2)**	**Avg. (1)**	**Poor (0)**	
Marks										

Experiment No – 6

AIM : Loop/ control structure using shell script(Using while loop)

6.1 Write a shell script to read n numbers as command arguments and sort them in descending order.

6.2 Write a shell script to generate mark sheet of a student. Take 3 subjects, calculate and display total marks, percentage and Class obtained by the student.

Date: ***// Write date of experiment here***

Competency and Practical Skills: Logic building and programming

Relevant CO: CO5

Objectives: explore usage of while loop in shell script

Equipment/Instruments: Computer System with Winows/Linux

Safety and necessary Precautions:

- ✓ Operate computer system carefully and responsibly.
- ✓ Use required lab resources cautiously

Theory:

Example:

Shows loop terminates as soon as a becomes 5

```
a=0
while [ $a -lt 10 ]
do
  echo $a
  if [ $a -eq 5 ]
  then
    break
  fi
  a=`expr $a + 1`
done
```

6.1 Write a shell script to read n numbers as command arguments and sort them in descending order.

Program:

Output:

6.2 Write a shell script to generate mark sheet of a student. Take 3 subjects, calculate and display total marks, percentage and Class obtained by the student.

Output:

Observations:

// Write your observation here

Conclusion:

// Write conclusion here

Suggested Reference:
https://www.tutorialspoint.com/
https://www.geeksforgeeks.org/
https://www.javatpoint.com/
https://www.tutorialspoint.com/unix/unix-loop-control.htm

References used by the students:

// Write references used by you here

Operating System (3140702) **220170107141**

Rubric-wise marks obtained:

Rubrics	Understanding of Shell programming syntax (4)			Ability to implement program for given problem using Shell script(4)			Documentation &Timely Submission (2)			Total (10)
	Good (4)	Avg. (3-2)	Poor (1-0)	Good (4)	Avg. (3-2)	Poor (1-0)	Good (2)	Avg. (1)	Poor (0)	
Marks										

Experiment No – 7

AIM: Command execution via Shell script.

7.1 Write a shell script to display all executable files, directories and zero sized files from current directory.

7.2 Write a menu driven shell script which will print the following menu and execute the given task.

MENU

- Display calendar of current month
- Display today's date and time
- Display usernames those are currently logged in the system
- Display your name at given x, y position
- Display your terminal number

Exit

Date: ***// Write date of experiment here***

Competency and Practical Skills: Logic building and programming

Relevant CO: CO5

Objectives: explore usage of various searching and date - time related commands.

Equipment/Instruments: Computer System with Winows/Linux

Safety and necessary Precautions:

✓ Operate computer system carefully and responsibly.
✓ Use required lab resources cautiously

7.1 Write a shell script to display all executable files, directories and zero sized files from current directory.

Theory:

find command

The **find** command in UNIX is a command line utility for walking a file hierarchy. It can be used to find files and directories and perform subsequent operations on them. It supports searching by

file, folder, name, creation date, modification date, owner and permissions. By using the '-exec' other UNIX commands can be executed on files or folders found.

Options:

- -exec CMD: The file being searched which meets the above criteria and returns 0 for as its exit status for successful command execution.
- -ok CMD : It works same as -exec except the user is prompted first.
- -inum N : Search for files with inode number 'N'.
- -links N : Search for files with 'N' links.
- -name demo : Search for files that are specified by 'demo'.
- -newer file : Search for files that were modified/created after 'file'.
- -perm octal : Search for the file if permission is 'octal'.
- -print : Display the path name of the files found by using the rest of the criteria.
- -empty : Search for empty files and directories.
- -size +N/-N : Search for files of 'N' blocks; 'N' followed by 'c'can be used to measure the size in characters; '+N' means size > 'N' blocks and '-N' means size < 'N' blocks.
- -user name : Search for files owned by username or ID 'name'.
- \(expr \) : True if 'expr' is true; used for grouping criteria combined with OR or AND.
- ! expr : True if 'expr' is false.

Example: Search for a file with a specific name.

1.$ find ./GFG -name sample.txt

It will search for sample.txt in GFG directory.

2. $ find ./GFG -name *.txt

It will give all files which have '.txt' at the end.

Program:

Output:

7.2 Write a menu driven shell script which will print the following menu and execute the given task.

MENU

- Display calendar of current month
- Display today's date and time
- Display usernames those are currently logged in the system
- Display your name at given x, y position
- Display your terminal number

Exit

Theory:

Case Statement

A case statement in bash scripts is used when a decision has to be made against multiple choices. In other words, it is useful when an expression has the possibility to have multiple values. This methodology can be seen as a replacement for multiple if-statements in a script. Case statements have an edge over if-statements because it improves the readability of our code and they are easier to maintain. Case statements in a Bash script are quite similar to Case statements in C language. But unlike C, the Bash Case statement stops continuing the search as soon as the match occurs. In simple words, they don't require any break statement that is mandatory to be used in C to stop searching for a pattern further.

The basic syntax of a case statement is given below,

Syntax:

```
case EXPRESSION in
  Pattern_Case_1)
    STATEMENTS
    ;;
 Pattern_Case_1)
    STATEMENTS
    ;;
 Pattern_Case_N)
    STATEMENTS
    ;;
 *)
    STATEMENTS
    ;;
esac
```

cal command

By default, the cal command shows the current month calendar as output.
cal command is a calendar command in Linux which is used to see the calendar of a specific month or a whole year.

Syntax:
cal [[month] year]

cal 08 2000 : Shows calendar of selected month and year.
cal 2018 : Shows the whole calendar of the year.
cal -3 : Shows calendar of previous, current and next month

who command

The who command is used to get information about currently logged in user on to system.

Syntax : $who [options] [filename]

Examples :
1. The who command displays the following information for each user currently logged in to the system if no option is provided :

Login name of the users
Terminal line numbers
Login time of the users in to system
Remote host name of the user

Program:

Output:

Observations:

// Write your observation here

Conclusion:

// Write conclusion here

Suggested Reference:
https://www.tutorialspoint.com/
https://www.geeksforgeeks.org/
https://www.javatpoint.com/

References used by the students:

// Write references used by you here

Rubric-wise marks obtained:

Rubrics	Understanding of Shell programming syntax (4)			Ability to implement program for given problem using Shell script(4)			Documentation &Timely Submission (2)			Total (10)
	Good (4)	Avg. (3-2)	Poor (1-0)	Good (4)	Avg. (3-2)	Poor (1-0)	Good (2)	Avg. (1)	Poor (0)	
Marks										

Experiment No – 8

AIM : Process Scheduling Algorithm and Comparison.

Date: ***// Write date of experiment here***

Competency and Practical Skills: Logic building and programming

Relevant CO: CO2

Objectives: Study and implement process scheduling algorithms.

Equipment/Instruments: Computer System with Winows/Linux

Safety and necessary Precautions:

- ✓ Operate computer systems carefully and responsibly.
- ✓ Use required lab resources cautiously

Theory:

8.1 Write a C Program to Implement Following CPU Scheduling Algorithms.

- **FCFS**
- **Round Robin**

First Come First Serve (FCFS) is an operating system scheduling algorithm that automatically executes queued requests and processes in order of their arrival. It is the easiest and simplest CPU scheduling algorithm. In this type of algorithm, processes which request the CPU first get the CPU allocation first. This is managed with a FIFO queue. The full form of FCFS is First Come First Serve.

Characteristics of FCFS CPU Scheduling Algorithm

- It supports non-preemptive and pre-emptive scheduling algorithm.
- Jobs are always executed on a first-come, first-serve basis.
- It is easy to implement and use.
- This method is poor in performance, and the general wait time is quite high.

Round Robin is a CPU scheduling algorithm where each process is assigned a fixed time slot in a cyclic way. It is basically the preemptive version of First come First Serve CPU Scheduling algorithm. Round Robin CPU Algorithm generally focuses on Time Sharing technique.

The period of time for which a process or job is allowed to run in a pre-emptive method is called time quantum.

Each process or job present in the ready queue is assigned the CPU for that time quantum, if the execution of the process is completed during that time then the process will end else the process will go back to the waiting table and wait for its next turn to complete the execution.

Characteristics of Round Robin CPU Scheduling Algorithm

It is simple, easy to implement, and starvation-free as all processes get fair share of CPU.

One of the most commonly used technique in CPU scheduling as a core.

It is preemptive as processes are assigned CPU only for a fixed slice of time at most.

The disadvantage of it is more overhead of context switching.

Program:

Output:

Observations:

// Write your observation here

Conclusion:

// Write conclusion here

Suggested Reference:

https://www.tutorialspoint.com/
https://www.geeksforgeeks.org/
https://www.javatpoint.com/

References used by the students:

// Write references used by you here

Operating System (3140702) **220170107141**

Rubric-wise marks obtained:

Rubrics	Understanding of Shell programming syntax (4)			Ability to implement program for given problem using Shell script(4)			Documentation &Timely Submission (2)			Total (10)
	Good (4)	Avg. (3-2)	Poor (1-0)	Good (4)	Avg. (3-2)	Poor (1-0)	Good (2)	Avg. (1)	Poor (0)	
Marks										

Experiment No – 9

AIM : Process creation and Thread Scheduling

9.1 Implement Producer consumer problem using thread using C/JAVA programming Language.

9.2 Create new thread using fork() system call using C programming Language.

Date: ***// Write date of experiment here***

Competency and Practical Skills: Logic building and programming

Relevant CO: CO3

Objectives: Study and implement thread management.

Equipment/Instruments: Computer System with Winows/Linux

Safety and necessary Precautions:

- ✓ Operate computer systems carefully and responsibly.
- ✓ Use required lab resources cautiously

Theory:

The producer-consumer problem (also known as the bounded-buffer problem) is a classic example of a multi-process synchronization problem. The problem describes two processes, the producer and the consumer, which share a common, fixed-size buffer used as a queue.

The producer's job is to generate data, put it into the buffer, and start again. At the same time, the consumer is consuming the data (i.e. removing it from the buffer), one piece at a time.

Problem:

To make sure that the producer won't try to add data into the buffer if it's full and that the consumer won't try to remove data from an empty buffer.

Solution:

The producer is to either go to sleep or discard data if the buffer is full. The next time the consumer removes an item from the buffer, it notifies the producer, who starts to fill the buffer again. In the same way, the consumer can go to sleep if it finds the buffer to be empty. The next time the producer puts data into the buffer, it wakes up the sleeping consumer.

An inadequate solution could result in a deadlock where both processes are waiting to be

awakened.

Multithreading is a Java feature that allows concurrent execution of two or more parts of a program for maximum utilization of CPU. Each part of such program is called a thread. So, threads are light-weight processes within a process.

Threads can be created by using two mechanisms :

- Extending the Thread class
- Implementing the Runnable Interface

The **Fork** system call is used for creating a new process in Linux, and Unix systems, which is called the child process, which runs concurrently with the process that makes the fork() call (parent process). After a new child process is created, both processes will execute the next instruction following the fork() system call.

The child process uses the same pc(program counter), same CPU registers, and same open files which use in the parent process. It takes no parameters and returns an integer value.

Below are different values returned by fork().
Negative Value: The creation of a child process was unsuccessful.
Zero: Returned to the newly created child process.
Positive value: Returned to parent or caller. The value contains the process ID of the newly created child process.
Example:

```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
int main()
{

        // make two process which run same
        // program after this instruction
        pid_t p = fork();
        if(p<0){
        perror("fork fail");
        exit(1);
        }
```

```
        printf("Hello world!, process_id(pid) = %d \n",getpid());
        return 0;
}
```

9.1 Implement Producer consumer problem using thread using C/JAVA programming Language.

Program:

Output:

9.2 Create new thread using fork() system call using C programming Language.

Program:

Output:

Observations:

// Write your observation here

Conclusion:

// Write conclusion here

Suggested Reference:

https://www.tutorialspoint.com/
https://www.geeksforgeeks.org/

https://www.javatpoint.com/
https://www.geeksforgeeks.org/producer-consumer-solution-using-threads-java/
https://www.geeksforgeeks.org/thread-functions-in-c-c/

References used by the students:

// Write references used by you here

Rubric-wise marks obtained:

Rubrics	**Understanding of Shell programming syntax (4)**			**Ability to implement program for given problem using Shell script(4)**			**Documentation &Timely Submission (2)**			**Total (10)**
	Good (4)	**Avg. (3-2)**	**Poor (1-0)**	**Good (4)**	**Avg. (3-2)**	**Poor (1-0)**	**Good (2)**	**Avg. (1)**	**Poor (0)**	
Marks										

Experiment No – 10

AIM : Page replacement and Disk Scheduling algorithm

10.1 Implement FIFO Page replacement Algorithm using C/Java.

10.2 Implement C-SCAN Disk Scheduling Algorithm using C/Java.

Date: ***// Write date of experiment here***

Competency and Practical Skills: Logic building and programming

Relevant CO: CO4

Objectives: Study and implement memory management by operating system.

Equipment/Instruments: Computer System with Winows/Linux

Safety and necessary Precautions:

- ✓ Operate computer systems carefully and responsibly.
- ✓ Use required lab resources cautiously

Theory:

In an operating system that uses paging for memory management, a page replacement algorithm is needed to decide which page needs to be replaced when a new page comes in.

Page Fault: A page fault happens when a running program accesses a memory page that is mapped into the virtual address space but not loaded in physical memory. Since actual physical memory is much smaller than virtual memory, page faults happen. In case of a page fault, Operating System might have to replace one of the existing pages with the newly needed page. Different page replacement algorithms suggest different ways to decide which page to replace. The target for all algorithms is to reduce the number of page faults.

First In First Out (FIFO): This is the simplest page replacement algorithm. In this algorithm, the operating system keeps track of all pages in the memory in a queue, the oldest page is in the front of the queue. When a page needs to be replaced page in the front of the queue is selected for

removal.

A Process makes the I/O requests to the operating system to access the disk. Disk Scheduling Algorithm manages those requests and decides the order of the disk access given to the requests.

Important Terms related to Disk Scheduling Algorithms

Seek Time - It is the time taken by the disk arm to locate the desired track.

Rotational Latency - The time taken by a desired sector of the disk to rotate itself to the position where it can access the Read/Write heads is called Rotational Latency.

Transfer Time - It is the time taken to transfer the data requested by the processes.

Disk Access Time - Disk Access time is the sum of the Seek Time, Rotational Latency, and Transfer Time.

C-SCAN

This algorithm is the same as the SCAN algorithm. The only difference between SCAN and C-SCAN is, it moves in a particular direction till the last and serves the requests in its path. Then, it returns in the opposite direction till the end and doesn't serve the request while returning. Then, again reverses the direction and serves the requests found in the path. It moves circularly.

10.1: Implement FIFO Page replacement Algorithm using C/Java.
Program:

Output:

10.2: Implement C-SCAN Disk Scheduling Algorithm using C/Java.
Program:

Output:

Observations:

// Write your observation here

Conclusion:

// Write conclusion here

Suggested Reference:
https://www.tutorialspoint.com/
https://www.geeksforgeeks.org/
https://www.javatpoint.com/
https://www.baeldung.com/cs/fifo-page-replacement

References used by the students:

// Write references used by you here

Rubric-wise marks obtained:

Rubrics	**Understanding of Shell programming syntax (4)**			**Ability to implement program for given problem using Shell script (4)**			**Documentation &Timely Submission (2)**			**Total (10)**
	Good (4)	**Avg. (3-2)**	**Poor (1-0)**	**Good (4)**	**Avg. (3-2)**	**Poor (1-0)**	**Good (2)**	**Avg. (1)**	**Poor (0)**	
Marks										