Diploma Engineering

Laboratory Manual

Basics of Operating System

(4330703)

Diploma Computer Engineering Semester 3

Enrolment No	
Name	
Branch	Computer Engineering
Academic Term	2024-25 Odd (3rd Semester)
Institute	A V Parekh Technical Institute

Directorate of Technical Education
Gandhinagar Gujarat

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 cater skilled engineers having potential to convert global challenges into opportunities through embedded values and quality technical education.

Institute's Mission

- Impart quality technical education and prepare diploma engineering professionals to meet the need of industries and society.
- Adopt latest tools and technologies for promoting systematic problem solving skills to promote innovation and entrepreneurship
- Emphasize individual development of students by inculcating moral, ethical and life skills.

Department's Vision

- Develop globally competent Computer Engineering Professionals to achieve excellence in an environment conducive for technical knowledge, skills, moral values and ethical values with a focus to serve the society.

Department's Mission

- To provide state of the art infrastructure and facilities for imparting quality education and computer engineering skills for societal benefit.
- Adopt industry-oriented curriculum with an exposure to technologies for building systems & application in computer engineering.
- To provide quality technical professional as per the industry and societal needs, encourage entrepreneurship, nurture innovation and life skills in consonance with latest interdisciplinary trends.

A.V. Parekh Technical Institute
(Department of Technical Education, Gujarat State)

Computer Engineering Department

Certificate

This is to certify that Mr./Ms…… Enrollment No. ……..……of **3rd Semester** of **Diploma in Computer Engineering Department of A V Parekh Technical Institute (GTU Code: 602)** has completed the term work satisfactorily in Subject **Basics of Operating System-4330703** for the academic year: **2024-2025** Term: **ODD** as prescribed in the curriculum.

Place: ……

Date: ……

Subject Faculty

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 -2021 (COGC-2021) for engineering diploma programmes. In that more time allotted to practical work than theory .It shows importance of enhancement of skills amongst students and it pays attention to utilize every seconds of time allotted for practical amongst students, instructors and Lecturers to achieve relevant outcomes by performing rather than writing practice in study type. It is must for effective implementation of competency focused outcome- based green curriculum-2021, every practical has been keenly designed to serve as a tool to develop & enhance relevant industry needed competency in each and every student. This psychomotor skills are very difficult to develop through traditional chalk and board content delivery method in the classroom. Accordingly, this lab manual 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 read procedure one day advance before actual performance day of practical experiment which creates interest and also they have idea of judgement of magnitude prior to performance .This in turn enhances pre-determined outcomes amongst students. Each and every experiment /Practical in this manual begins by competency, industry relevant skills, course outcomes as well as practical outcomes which serve as a key role for doing the practical.

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

Basics of Operating System course enables to understand importance of Operating System, its functionalities to manage resources of Computer and Peripherals, program development and its execution. Every student of computer science must therefore understand basic structure of an operating system. After learning this subject student will be able to discriminate between various types of operating systems, its processor, processes, and memory and file management. The subject also emphasizes on Linux utilities and scripting

Although we try our level best to design this lab manual, but always there are chances of improvement. We welcome any suggestions for improvement.

Programme Outcomes (POs) to be achieved through Practical of this Course

Following programme outcomes are expected to be achieved through the practical of the course:

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.

2. **Problem analysis**: Identify and analyze well-defined engineering problems using codified standard methods

3. **Design/ development of solutions** Design solutions for well-defined technical problems and assist with the design of systems components or processes to meet specified needs

4. **Engineering Tools, Experimentation and Testing:** Apply modern engineering tools and appropriate technique to conduct standard tests and measurements

5. **Engineering practices for society, sustainability and environment:** Apply appropriate technology in context of society, sustainability, environment and ethical practices.

6. **Project Management:** Use engineering management principles individually, as a team member or a leader to manage projects and effectively communicate about well-defined engineering activities.

7. **Life-long learning:** Ability to analyze individual needs and engage in updating in the context of technological changes.

Practical Outcome - Course Outcome matrix

Course Outcomes (COs):

a. *CO1:* Differentiate Operating Systems based on features.

b. *CO2:* Apply scheduling algorithms to calculate turnaround time and average waiting time.

c. *CO3:* Interpret various memory management techniques.

d. *CO4:* Apply file management techniques.

e. *CO5:* Execute basic Linux commands and Shell scripts

Sr. No.	Practical Outcome	CO1	CO2	CO3	CO4	CO5
1.	Compare windows and Linux OS. (latest version)	√	-	-	-	-
2.	Solve example with SJF, FCFS and Round robin algorithm. Draw Gantt chart.	-	√	-	-	-
3.	Solve example using following algorithms: 1. First fit 2. Best fit 3. Worst fit	-	-	√	-	-
4.	Page replacement algorithms 1. First in First out (FIFO) 2. Least Recently Used	-	-	√	-	-
5.	Disk Scheduling Algorithms 1. Scan 2. CScan	-	-	-	√	-
6.	Test and run basic unix commands	-	-	-	-	√
7.	Test and run Advanced unix commands.	-	-	-	-	√
8.	Test commands related with File editing with Vi, Vim, gedit, gcc.	-	-	-	-	√
9.	Create a shell script to read from command line and print “Hello”.	-	-	-	-	√
10.	Create a Shell script to read and display content of a file. And append content of one file to another	-	-	-	-	√
11.	Create a Shell script to accept a string in lower case letters from a user, & convert to upper case letters.	-	-	-	-	√
12.	Create a Shell script to add two numbers.	-	-	-	-	√

Industry Relevant Skills

The following industry relevant skills are expected to be developed in the students by performance of experiments of this course.

1. Analytical skills
2. Problem solving skills
3. Intermediate computer skills require in-depth knowledge of complex computer application and software.

Guidelines to Teachers

1. Teacher should provide the guideline with demonstration of practical to the students.
2. Teacher is expected to refer complete curriculum document and follow guidelines for implementation strategies.
3. Teacher shall explain prior concepts and industrial relevance to the students before starting of each practical
4. Involve all students in performance of each experiment and should give opportunity to students for hands on experience.
5. Teacher should ensure that the respective skills and competencies are developed in the students after the completion of the practical exercise.
6. Teacher is expected to share the skills and competencies to be developed in the students.
7. Finally give practical quiz as per the instructions. Utilise 2 hrs of lab hours effectively and ensure completion of write up with quiz also on same day.

Instructions for Students

1. Listen carefully the lecture, curriculum, learning structure, skills to be developed.
2. Organize the work in the group and make record of all observations.
3. Students shall develop maintenance skill as expected by industries.
4. Student shall attempt to develop related hand-on skills and build confidence.
5. Student shall develop the habits of evolving more ideas, innovations, skills etc.
6. Student shall refer technical magazines and data books.
7. Student should develop habit to submit the practical on date and time.
8. Student should well prepare while submitting write-up of exercise.

A. V. PAREKH TECHNICAL INSTITUTE, RAJKOT

COMPUTER ENGINEERING DEPARTMENT

ASSESSMENT RUBRICS FOR PRACTICAL COMPONENTS

- **SUBJECT & CODE: BASICS OF OPERATING SYSTEM (4330703)**
- **CONTINUOUS ASSESSMENT (25 Marks)**

- **Laboratory Work and Questionnaire Component (25 Marks):**

Component	Criteria	Percentage	Marks	Assessment
Laboratory Work and Questionnaire	**Excellent**	**91%-100%**	**5 marks**	Demonstrates exceptional proficiency in both laboratory work and questionnaire assessments, consistently applying skills and understanding effectively.
	Proficient	**71%-90%**	**4 marks**	Shows a strong command of both laboratory work and questionnaire assessments, with minor areas for improvement.
	Satisfactory	**51%-70%**	**3 marks**	Achieves a satisfactory level of performance in laboratory work and questionnaire assessments, with room for improvement in some areas.
	Needs Improvement	**31%-50%**	**2 marks**	Demonstrates limited proficiency in both laboratory work and questionnaire assessments, with significant areas for improvement.
	Inadequate	**0%-30%**	**0/1 mark**	Fails to meet acceptable standards in both laboratory work and questionnaire assessments; significant improvement is required.

A. V. PAREKH TECHNICAL INSTITUTE, RAJKOT
COMPUTER ENGINEERING DEPARTMENT
ASSESSMENT RUBRICS FOR PRACTICAL COMPONENTS

- **SUBJECT & CODE: BASICS OF OPERATING SYSTEM (4330703)**
- **END SEMESTER EXAMINATION (25 Marks):**

- **Viva Examination (25 Marks):**

Component	Criteria	Percentage	Marks	Assessment
Viva Examination	**Excellent**	**91%-100%**	**5 Marks**	Demonstrates exceptional proficiency in the viva exam, displaying an in-depth understanding and providing comprehensive and insightful answers.
	Proficient	**71%-90%**	**4 Marks**	Displays a strong grasp of the viva exam topics, providing clear and well-reasoned answers, with minor areas for improvement.
	Satisfactory	**51%-70%**	**3 Marks**	Provides satisfactory responses during the viva exam, covering the essential topics, with room for improvement in some areas.
	Needs Improvement	**31%-50%**	**2 Marks**	Demonstrates limited understanding of the viva exam topics, providing answers that may lack clarity or depth, with significant areas for improvement.
	Inadequate	**0%-30%**	**0/1 Mark**	Fails to meet acceptable standards in the viva exam, providing answers that are unclear, incorrect, or lacking substance; significant improvement is required.

***Note**: This Rubric is applied to the ESE Components of the Courses where End Semester Examination is conducted by Institute Faculty Internally. For the Final Year Courses, ESE Exam is conducted by an External Faculty appointed by university. So, for those courses the marks are converted from the GTU Grade and equally divided into all the COs.*

Continuous Assessment Sheet

Enrolment No:_____ Term: 2024-25 ODD

Name: _____

Sr. No.	Practical Outcome/Title of experiment	CO	Page	Date	Marks (25)	Sign
1.	Compare windows and Linux OS. (latest version)	CO1				
2.	Solve example with FCFS, SJF and Round robin algorithm. Draw Gantt chart.	CO2				
3.	Solve example using following algorithms: 1. First fit 2. Best fit 3. Worst fit	CO3				
4.	Page replacement algorithms 1. First in First out (FIFO) 2. Least Recently Used					
5.	Disk Scheduling Algorithms 1. Scan 2. CScan	CO4				
6.	Test and run basic unix commands	CO5				
7.	Test and run Advanced unix commands.					
8.	Test commands related to File editing with Vi, Vim, gedit, gcc.					
9.	Create a shell script to read from command line and print “Hello”.					
10.	Create a Shell script to read and display content of a file. And append content of one file to another					
11.	Create a Shell script to accept a string in lower case letters from a user, & convert to upper case letters.					
12.	Create a Shell script for adding two numbers and print table of a given number.					
	Total					

Practical No.1: Compare windows and Linux OS. (Latest version)

Objective: Each operating system has its advantages and disadvantages. Our aim in this practical is to compare the most significant aspects, so that each user can decide for themselves whether or not to switch or stick.

A. Expected Program Outcomes (POs)

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problem.

2. **Life-long learning:** Ability to analyze individual needs and engage in updating in the context of technological changes.

B. Expected Skills to be developed based on competency:

This practical is expected to develop the following skills for the industry identified competency: 'Compare windows and Linux OS.'

1. Selection of OS based on users requirements.
2. Identify pros and cons of windows and Linux OS.

C. Expected Course Outcomes(Cos)

1. Differentiate operating systems based on their features.

D. Practical Outcome(Pro)

1. Compare windows and Linux OS. (latest version)

E. Expected Affective domain Outcome(ADos)

1. Follow precautionary measures.
2. Follow ethical practice.

F. Prerequisite Theory:

What is Linux?

Linux is a free and open source operating system based on Unix standards which provides a programming interface as well as user interface compatibility. It also contains many separately developed elements, free from proprietary code.

The traditional **monolithic kernel** is employed in the Linux kernel for performance purposes. Its modular feature allows most drivers to dynamically load and unload at run time.

Linux was created by Finnish student Linus Torvalds, who wanted to create a free operating system kernel that anyone could use. It was launched much later than Windows, in 1991. Although it still was regarded as a very bare bones operating system, without a graphical interface like Windows. With just a few lines of source code in its original release to where it stands today, containing more than 23.3 million lines of the source code, Linux has surely grown considerably.

Linux was first distributed under GNU General Public License in 1992.

What is Windows?

Windows is a licensed operating system and its source code is inaccessible**.** It is designed for business owners, other commercial user and even individuals with no computer programming knowledge. It is simple and straightforward to use.

Windows offers features like,

- Multiple operating environments
- Symmetric multiprocessing
- Client-server computing
- Integrated caching
- Virtual memory
- Portability
- Extensibility
- Pre-emptive scheduling

The first version of Windows, known as Windows 1.0, revealed in 1985 following the formation of Microsoft. It was based upon the MS-DOS core. Following that initial launch, new versions of Windows were quickly rolled out. This included the first major update in 1987 and Windows 3.0 in the same year.

In 1995, perhaps the most widely used version yet, Windows 95 was born. At this point, it ran on a 16-bit DOS-based kernel and 32-bit user space to enhance the user experience.

Windows hasn't changed a whole lot in terms of core architecture since this version despite vast amounts of features that have been added to address modern computing.

Linux v/s Windows

Users

There are **3 types** of users in Linux (Regular, Administrative (root) and Service users) whereas, in Windows, there are **4 types** of user accounts (Administrator, Standard, Child and Guest).

Usage

According to the market research data, on 92.63% of the world's PCs, Windows is running, while hardly 1% of PC users use Linux. Many users say it is hard to use Linux as compared to Windows and so the appeal of Linux is very limited for household use. It is mostly meant for serious use such as server application. Hence, corporation servers are running on Linux, irrespective of the GUI.

Also, due to the licensing agreement with Microsoft, various PC vendors are entitled to bundle Windows OS with their PC, enabling Windows to gain market popularity over Linux. Though these days many PC vendors such as Dell and HP have started to give Linux as the pre-installed OS, Windows still rules the market.

Kernel

Linux uses the monolithic kernel which consumes more running space whereas Windows uses the micro-kernel which takes less space but lowers the system running efficiency than Linux.

File Systems

In Microsoft Windows, files are stored in directories/folders on different data drives like C: D: E: but, in Linux, files are ordered in a tree structure starting with the root directory, further branched out to various other sub-directories.

In Linux, everything is treated like a file. Directories are files, files are files, and externally connected devices (such as Printer, mouse, keyboard) are also files.

Security

Every Windows user has faced security and stability issues at some point in time. Since Windows is an extensively used OS, hackers, spammers target Windows frequently. Windows (consumer versions) were originally designed for ease-of-use on a single-user PC without a network connection and did not have security features built in. Microsoft often releases security patches through its Windows Update service. These go out once a month, although critical updates are made available at shorter intervals or when necessary.
Many a time, users of Windows OS face the *BLUE SCREEN OF DEATH*. This is caused due to the failure of the system to respond. Eventually, the user has to keep aside his/her frustrations and manually restart the PC.
On the other hand, Linux is based on a multi-user architecture, making it way more stable than a single-user OS like Windows. As Linux is community-driven with regular monitoring by the developers from every corner of the earth, any new problem raised can be solved within a few hours and the necessary patch can be ready for supply.

Compatibility

Windows shoots. Windows scores. This is where the Redmond offering wipes the floor with Linux. Despite recent improvements in software being ported or developed to Linux, Windows is still the king of compatibility.
Users of Windows can be certain that almost any software (even the most obscure, outdated software) will work, even when it is abandoned by developers. Windows has great legacy support. Plain and simple.
Linux, on the other hand, has been struggling with basics that Windows users take for granted.

Ease of Use

Linux, over recent years, has gone leaps and bounds when it comes to usability. Distributions like Linux Mint and Ubuntu, have even gone as far as making their ***installation and setup*** simpler for non-technical users to do carry on with day-to-day activities with the utmost ease.
Windows, due to its market proliferation, is the default OS on many devices. Buy a new laptop or PC, and there is quite the chance it comes with Windows 10 installed. Users are used to clicking the toolbar and opening their favourite programs, for so many years now that it makes it very difficult for them to make the switch.

Privacy

If you are a Linux user, you have an operating system that doesn't *spy* on you. Having Linux means the system is *yours and yours alone*. You can also add to the mix that most Linux systems come with an option of *built-in military-grade encryption.* As a user, you can be sure that device theft poses no real problem to your data.
On the contrary, Windows has gotten more advert driven over the last few years. Users are definitely given the choice to opt out but then again, who can help the clever registry hacks which are clearly a part of Redmond's plan. Windows can watch what users do, offering to sync to the Microsoft One-Drive service or to learn behaviour to make Cortana (the Microsoft personal assistant) better. To be honest, I do not favour these tools, as they're pretty intrusive. Though, some users like these features. Subjective opinion.

Source Code

Linux is an open source operating system whereas Windows OS is commercial. Linux has access to source code and alters the code as per user need whereas Windows does not have access to the source code.
In Linux, the user has access to the source code of the kernel and alter the code according to his need. It has its own advantages. Bugs in the OS will get fixed at a rapid pace but developers may take advantage of any weakness in the OS if found.
In windows only selected members to have access to the source code.

License

The Linux kernel (and the GNU utilities and libraries which accompany it) in most distributions are entirely free and open source. Companies offer paid support for their distros, but the underlying software is still free to download and install.

Reliability

Windows, as we all have come to know, becomes sluggish by the day. You have to re-install Windows after a while when you encounter crashes or slowdowns on your system.

If you are a Linux user, you will not have to re-install it just to experience a faster and a smoother system. Linux helps your system run smoothly for a longer period. Much longer! in fact.

Also, as a Windows user, you will have to develop a habit where you keep on rebooting the system for just about everything.

Just installed software – reboot!

Uninstalled software – reboot!

Got a Windows update – reboot!

System seems to slow down, you guessed it right – reboot!

However, in the case of Linux, you can comfortably continue with your work, and your OS will not bother you.

This is probably why you could observe that most of the Internet giants like Facebook and Google run on Linux. Even supercomputers run mostly on Linux.

So, why isn't Windows preferred over Linux in the industries? It is because Linux is far more reliable than Windows OS. Period.

Linux vs Windows: Distributions

Firstly, we need to address one of the more confusing aspects to the Linux platform. While Windows has maintained quite the standard version structure, with updates and versions split into tiers, Linux is far more complex.

Originally designed as an OS accessible to all, the Linux Kernel today underpins all Linux operating systems. However, as it remains open source, anyone can tweak or modify the OS for their own purposes. What we have as a result are hundreds of bespoke Linux-based OS called distributions, or 'distros'. This makes it incredibly difficult to choose between them, making it far more complicated than simply picking Windows 7, Windows 8 or Windows 10.

But this also comes with its set of advantages. Given the nature of open source software, these distributions can vary wildly in functionality and sophistication, and many are constantly evolving. The choice is almost overwhelming.

For those of you who are new to Linux, we'd recommend ***Ubuntu*** as a good starting point. It's extremely user-friendly (even compared to Windows) whilst still being versatile and feature-rich enough to satisfy experienced techies. It's the closest thing Linux has to a *default distribution* – although I'd urge everyone to explore the various distro options available and find their favourite.

Linux vs Windows:

This depends on what you need to do.

Multimedia

Both Linux and Windows OS are pretty rich in multimedia applications. Although, setting up the sound and video options in older versions of Linux can be relatively difficult for some users. A key advantage of Linux is that most of the multimedia applications are freely available. In the case of Windows, users may have to pay a hefty price to get the software although many Open Source/free versions are often available.

Gaming

If you are a gamer, need 100% compatibility with a particular software or want a user-friendly system, Hands down, Windows wins. Steam, among other clients and options, provides a huge number of games both from AAA publishers and small indie developers. Steam for Linux now allows you to install Windows games. But it is still in beta, and not all Windows games will work. It can be frustrating for Linux users, and no doubt the situation will change in the future.

But for now, in 2018, many Linux users miss out on the top games with their choice of OS. The graphics card vendors also tend to support Windows platforms rather than Linux. They provide timely updates and new features that don't always filter to other OS.

Open Source

If you are an advocate of open-source software or just plain tired of all the forced update and reboot in Windows, then Linux may be a viable option.

Programming

Linux supports almost all of the major programming languages (*Python, C/C++, Java, Ruby, Perl, etc*.). Moreover, it portrays a vast range of applications useful for programming purposes.

The Linux terminal is far more superior to use over Window's command line for developers. You would find many libraries natively developed for Linux. Also, a lot of programmers point out that they can do things easily using the package manager on Linux. The ability to script in ***different shells*** is also one of the most compelling reasons why programmers prefer using Linux OS.

Linux brings in native support for SSH, too. This would help you manage your servers quickly. You could include things like apt-get ***commands*** which further makes Linux a more popular choice of the programmers.

G. Practical related Quiz.

1. Give difference between Windows and Linux.
2. Give difference between windows 8 and windows 10
3. Which OS is faster, Windows or Linux? Give reason.
4. How Linux is more secure than Windows?

Sign

Practical No.2: Solve examples with FCFS, SJF and Round robin algorithm. Draw Gantt chart for each.

Objective: Scheduling is important in many different computer environments. One of the most important areas is scheduling which programs will work on the CPU. This task is handled by the Operating System (OS) of the computer and there are many different ways in which we can choose to configure programs.

A. Expected Program Outcomes (POs)

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.

2. **Problem analysis**: Identify and analyze well-defined engineering problems using codified standard methods

3. **Design/ development of solutions** Design solutions for well-defined technical problems and assist with the design of systems components or processes to meet specified needs

4. **Life-long learning:** Ability to analyze individual needs and engage in updating in the context of technological changes.

B. Expected Skills to be developed based on competency

This practical is expected to develop the following skills for the industry identified competency: 'Solve examples with various scheduling algorithms.'

1. Able to understand various scheduling algorithms.
2. Able to solve given example with SJF, FCFS and Round robin algorithm
3. Able to draw Gantt chart for SJF, FCFS and Round robin algorithm

C. Expected Course Outcomes(Cos)

Apply scheduling algorithms to calculate turnaround time and average waiting time.

D. Practical Outcome(Pro)

1. Solve examples with SJF, FCFS and Round robin algorithm. Draw Gantt chart for each.

E. Expected Affective domain Outcome(ADos)

1. Follow precautionary measures.
2. Follow ethical practice.

F. Prerequisite Theory:

FCFS Scheduling

First come first serve (FCFS) scheduling algorithm simply schedules the jobs according to their arrival time. The job which comes first in the ready queue will get the CPU first. The lesser the arrival time of the job, the sooner will the job get the CPU. FCFS scheduling may cause the problem of starvation if the burst time of the first process is the longest among all the jobs.

Advantages of FCFS

- Simple
- Easy
- First come, First serve

Disadvantages of FCFS

1. The scheduling method is non pre-emptive, the process will run to the completion.
2. Due to the non-preemptive nature of the algorithm, the problem of starvation may occur.
3. Although it is easy to implement, but it is poor in performance since the average waiting time is higher as compare to other scheduling algorithms.

Example

Let's take an example of The FCFS scheduling algorithm. In the Following schedule, there are 5 processes with process ID **P0, P1, P2, P3 and P4**. P0 arrives at time 0, P1 at time 1, P2 at time 2, P3 arrives at time 3 and Process P4 arrives at time 4 in the ready queue. The processes and their respective Arrival and Burst time are given in the following table.

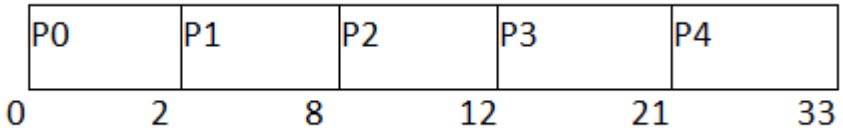
The Turnaround time and the waiting time are calculated by using the following formula.

1. Turn Around Time = Completion Time - Arrival Time
2. Waiting Time = Turnaround time - Burst Time

The average waiting Time is determined by summing the respective waiting time of all the processes and divided the sum by the total number of processes.

Process ID	Arrival Time	Burst Time	Completion Time	Turn Around Time	Waiting Time
P0	0	2	2	2	0
P1	1	6	8	7	1
P2	2	4	12	10	6
P3	3	9	21	18	9
P4	6	12	33	29	17

Average Waiting Time=31/5



Gantt chart

Shortest Job First (SJF) Scheduling

Till now, we were scheduling the processes according to their arrival time (in FCFS scheduling). However, SJF scheduling algorithm, schedules the processes according to their burst time.

In SJF scheduling, the process with the lowest burst time, among the list of available processes in the ready queue, is going to be scheduled next.

However, it is very difficult to predict the burst time needed for a process hence this algorithm is very difficult to implement in the system.

Advantages of SJF

1. Maximum throughput
2. Minimum average waiting and turnaround time

Disadvantages of SJF

1. May suffer with the problem of starvation
2. It is not implementable because the exact Burst time for a process can't be known in advance.

There are different techniques available by which, the CPU burst time of the process can be determined. We will discuss them later in detail.

Example

In the following example, there are five jobs named as P1, P2, P3, P4 and P5. Their arrival time and burst time are given in the table below.

PID	Arrival Time	Burst Time	Completion Time	Turn Around Time	Waiting Time
P1	1	7	8	7	0
P2	3	3	13	10	7
P3	6	2	10	4	2
P4	7	10	31	24	14
P5	9	8	21	12	4

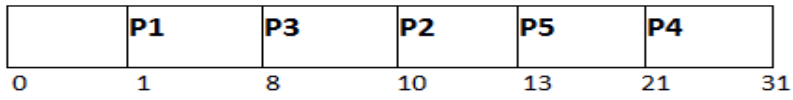
Since, No Process arrives at time 0 hence; there will be an empty slot in the **Gantt chart** from time 0 to 1 (the time at which the first process arrives).

According to the algorithm, the OS schedules the process which is having the lowest burst time among the available processes in the ready queue.

Till now, we have only one process in the ready queue hence the scheduler will schedule this to the processor no matter what is its burst time.

This will be executed till 8 units of time. Till then we have three more processes arrived in the ready queue hence the scheduler will choose the process with the lowest burst time.

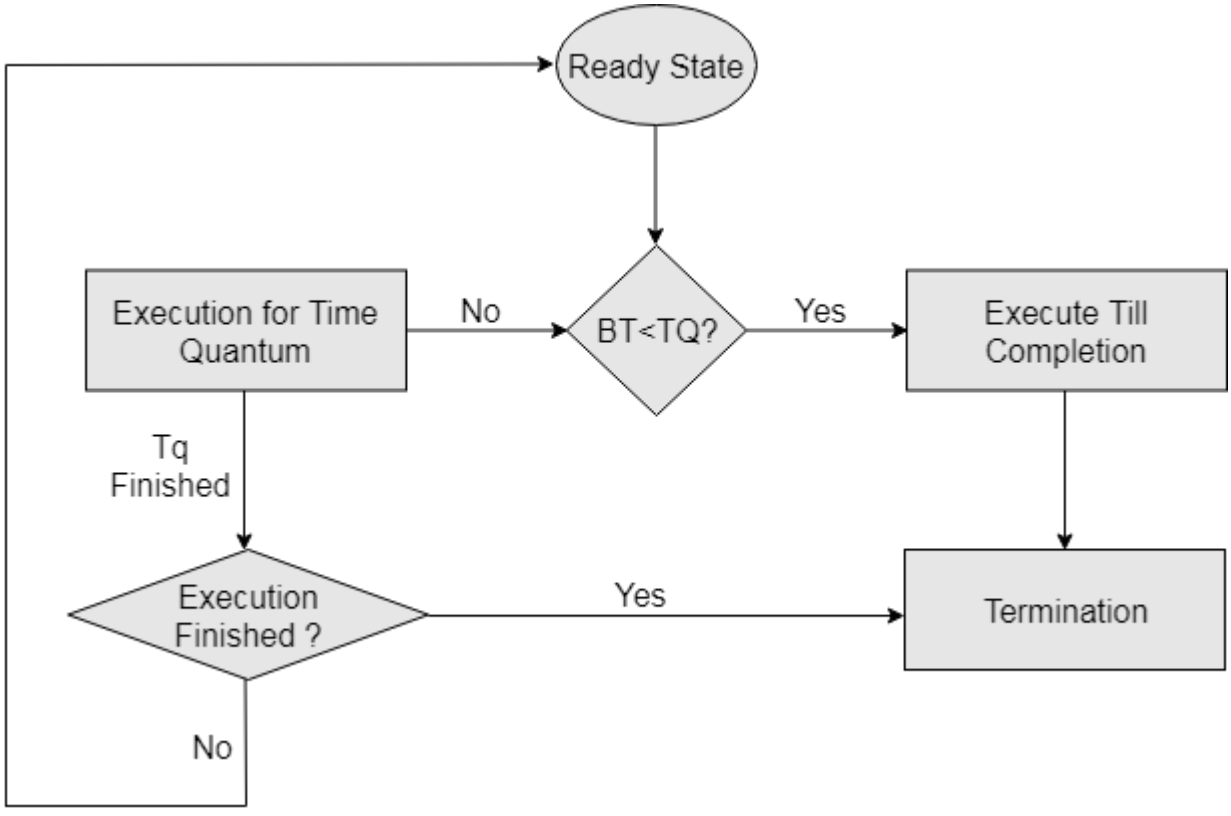
Among the processes given in the table, P3 will be executed next since it is having the lowest burst time among all the available processes. So that's how the procedure will go on in **shortest job first (SJF)** scheduling algorithm.
Avg Waiting Time = 27/5



Gantt chart

Round Robin Scheduling Algorithm

Round Robin scheduling algorithm is one of the most popular scheduling algorithm which can actually be implemented in most of the operating systems. This is the **preemptive version** of first come first serve scheduling. The Algorithm focuses on Time Sharing. In this algorithm, every process gets executed in a **cyclic way**. A certain time slice is defined in the system which is called time **quantum**. Each process 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 **terminate** else the process will go back to the **ready queue** and waits for the next turn to complete the execution.



Advantages

1. It can be actually implementable in the system because it is not depending on the burst time.
2. It doesn't suffer from the problem of starvation or convoy effect.
3. All the jobs get a fare allocation of CPU.

Disadvantages

1. The higher the time quantum, the higher the response time in the system.
2. The lower the time quantum, the higher the context switching overhead in the system.
3. Deciding a perfect time quantum is really a very difficult task in the system.

RR Scheduling Example

In the following example, there are six processes named as P1, P2, P3, P4, P5 and P6. Their arrival time and burst time are given below in the table. The time quantum of the system is 4 units.

Process ID	Arrival Time	Burst Time
P1	0	5
P2	1	6
P3	2	3
P4	3	1
P5	4	5
P6	6	4

According to the algorithm, we have to maintain the ready queue and the Gantt chart. The structure of both the data structures will be changed after every scheduling.

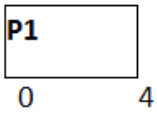
Ready Queue:

Initially, at time 0, process P1 arrives which will be scheduled for the time slice 4 units. Hence in the ready queue, there will be only one process P1 at starting with CPU burst time 5 units.

P1
5

GANTT chart

The P1 will be executed for 4 units first.



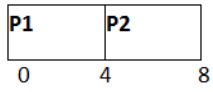
Ready Queue

Meanwhile the execution of P1, four more processes P2, P3, P4 and P5 arrives in the ready queue. P1 has not completed yet, it needs another 1 unit of time hence it will also be added back to the ready queue.

P2	P3	P4	P5	P1
6	3	1	5	1

GANTT chart

After P1, P2 will be executed for 4 units of time which is shown in the Gantt chart.



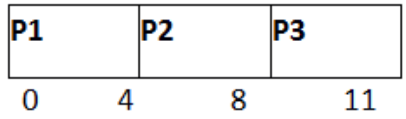
Ready Queue

During the execution of P2, one more process P6 is arrived in the ready queue. Since P2 has not completed yet hence, P2 will also be added back to the ready queue with the remaining burst time 2 units.

P3	P4	P5	P1	P6	P2
3	1	5	1	4	2

GANTT chart

After P1 and P2, P3 will get executed for 3 units of time since its CPU burst time is only 3 seconds.



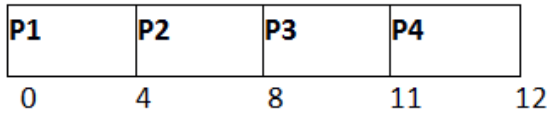
Ready Queue

Since P3 has been completed, hence it will be terminated and not be added to the ready queue. The next process will be executed is P4.

P4	P5	P1	P6	P2
1	5	1	4	2

GANTT chart

After, P1, P2 and P3, P4 will get executed. Its burst time is only 1 unit which is lesser then the time quantum hence it will be completed.



Ready Queue

The next process in the ready queue is P5 with 5 units of burst time. Since P4 is completed hence it will not be added back to the queue.

P5	P1	P6	P2
5	1	4	2

GANTT chart

P5 will be executed for the whole time slice because it requires 5 units of burst time which is higher than the time slice.

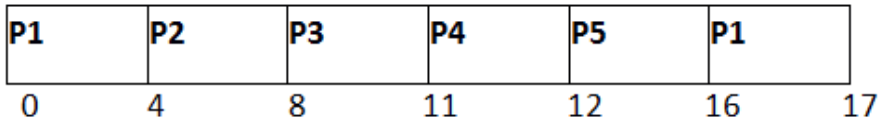
Ready Queue

P5 has not been completed yet; it will be added back to the queue with the remaining burst time of 1 unit.

P1	P6	P2	P5
1	4	2	1

GANTT Chart

The process P1 will be given the next turn to complete its execution. Since it only requires 1 unit of burst time hence it will be completed.



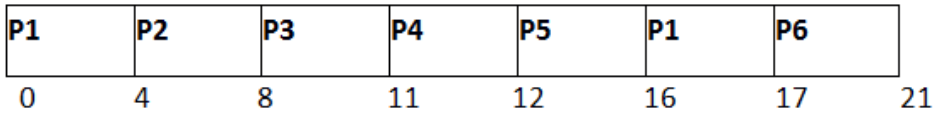
Ready Queue

P1 is completed and will not be added back to the ready queue. The next process P6 requires only 4 units of burst time and it will be executed next.

P6	P2	P5
4	2	1

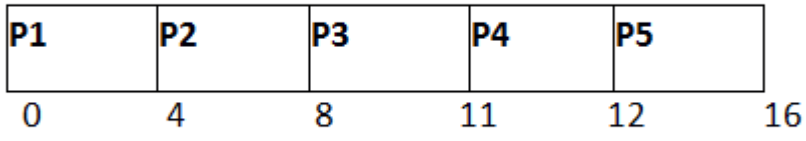
GANTT chart

P6 will be executed for 4 units of time till completion.



Ready Queue

Since P6 is completed, hence it will not be added again to the queue. There are only two processes present in the ready queue. The Next process P2 requires only 2 units of time.



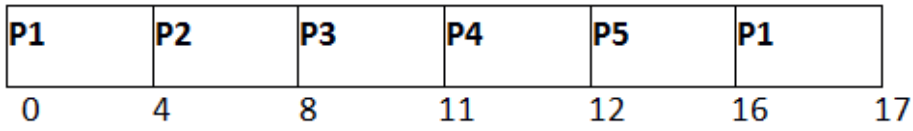
Ready Queue

P5 has not been completed yet; it will be added back to the queue with the remaining burst time of 1 unit.

P1	P6	P2	P5
1	4	2	1

GANTT Chart

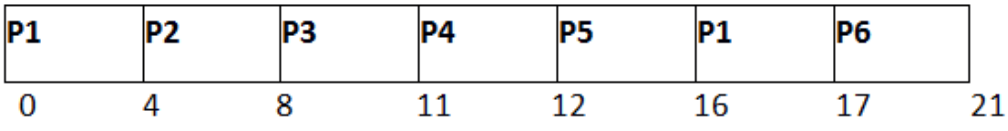
The process P1 will be given the next turn to complete its execution. Since it only requires 1 unit of burst time hence it will be completed.



Ready Queue
P1 is completed and will not be added back to the ready queue. The next process P6 requires only 4 units of burst time and it will be executed next.

P6	P2	P5
4	2	1

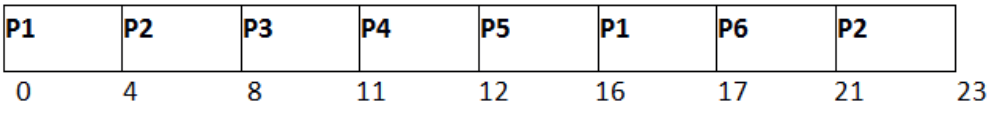
GANTT chart
P6 will be executed for 4 units of time till completion.



Ready Queue
Since P6 is completed, hence it will not be added again to the queue. There are only two processes present in the ready queue. The Next process P2 requires only 2 units of time.

P2	P5
2	1

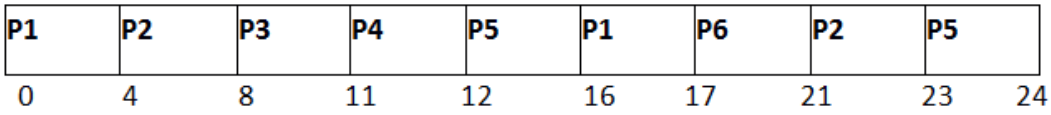
GANTT Chart
P2 will get executed again, since it only requires only 2 units of time hence this will be completed.



Ready Queue
Now, the only available process in the queue is P5 which requires 1 unit of burst time. Since the time slice is of 4 units hence it will be completed in the next burst.

P5
1

GANTT chart
P5 will get executed till completion.



The completion time, Turnaround time and waiting time will be calculated as shown in the table below.
As, we know,

1. Turn Around Time = Completion Time - Arrival Time
2. Waiting Time = Turn Around Time - Burst Time

Process ID	Arrival Time	Burst Time	Completion Time	Turn Around Time	Waiting Time
P1	0	5	17	17	12
P2	1	6	23	22	16
P3	2	3	11	9	6
P4	3	1	12	9	8
P5	4	5	24	20	15
P6	6	4	21	15	11

Average Waiting Time = (12+16+6+8+15+11)/6 = 76/6 units

G. Practical related Quiz.

1. Solve below given example with FCFS, SJF and Round robin algorithm. Draw Gantt chart and find average waiting time as well as turnaround time for the same.

Process	Arrival Time	Execution Time
P0	0	5
P1	1	3
P2	2	8
P3	3	6

2. Solve below given example with FCFS, SJF and Round robinalgorithm. Draw Gantt chart and find average waiting time as well as turnaround time for the same.

Process	Arrival Time	Burst Time
P0	3	5
P1	0	4
P2	4	2
P3	5	4

Sign

Practical No.3: Solve examples using First fit, Best fit, Worst fit Algorithms

Objective: Memory management is the process of controlling and coordinating a computer's main memory. It ensures that blocks of memory space are properly managed and allocated so the operating system (OS), applications and other running processes have the memory they need to carry out their operations.

A. Expected Program Outcomes (POs)

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.

2. **Problem analysis**: Identify and analyze well-defined engineering problems using codified standard methods

3. **Design/ development of solutions** Design solutions for well-defined technical problems and assist with the design of systems components or processes to meet specified needs

4. **Life-long learning:** Ability to analyze individual needs and engage in updating in the context of technological changes.

B. Expected skills to be developed based on competency

This practical is expected to develop the following skills for the industry identified competency:

1. Able to understand various memory management algorithms.
2. Able to solve problems related memory management using various algorithms.

C. Expected Course Outcomes(Cos)

Interpret various memory management techniques.

D. Practical Outcome(Pro)

1. Solve given examples of memory management techniques using First fit, Best fit, Worst fit Algorithms

E. Expected Affective domain Outcome(ADos)

1. Follow precautionary measures.
2. Follow ethical practice.

F. Prerequisite Theory:

Partitioning Algorithms

There are various algorithms which are implemented by the Operating System in order to find out the holes in the linked list and allocate them to the processes.

The explanation about each of the algorithm is given below.

1. First Fit Algorithm

First Fit algorithm scans the linked list and whenever it finds the first big enough hole to store a process, it stops scanning and load the process into that hole. This procedure produces two partitions. Out of them, one partition will be a hole while the other partition will store the process.

First Fit algorithm maintains the linked list according to the increasing order of starting index. This is the simplest to implement among all the algorithms and produces bigger holes as compare to the other algorithms.

2. Best Fit Algorithm

The Best Fit algorithm tries to find out the smallest hole possible in the list that can accommodate the size requirement of the process.

Using Best Fit has some disadvantages.

1. It is slower because it scans the entire list every time and tries to find out the smallest hole which can satisfy the requirement the process.
2. Due to the fact that the difference between the whole size and the process size is very small, the holes produced will be as small as it cannot be used to load any process and therefore it remains useless. Despite of the fact that the name of the algorithm is best fit, It is not the best algorithm among all.

3. Worst Fit Algorithm

The worst fit algorithm scans the entire list every time and tries to find out the biggest hole in the list which can fulfill the requirement of the process.

Despite of the fact that this algorithm produces the larger holes to load the other processes, this is not the better approach due to the fact that it is slower because it searches the entire list every time again and again.

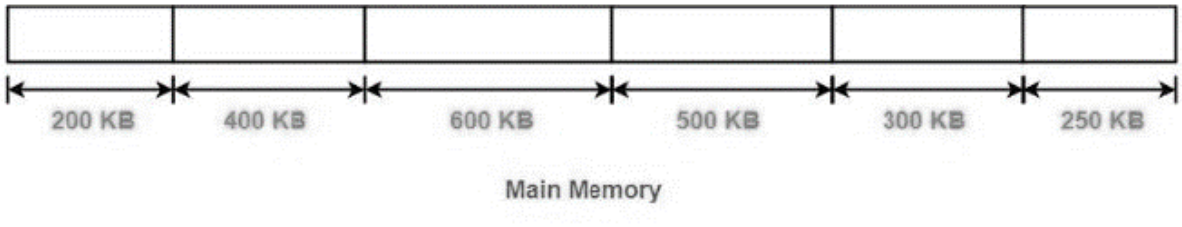
Example:

Consider six memory partitions of size 200 KB, 400 KB, 600 KB, 500 KB, 300 KB and 250 KB. These partitions need to be allocated to four processes of sizes 357 KB, 210 KB, 468 KB and 491 KB in that order.

Perform the allocation of processes using-

1. First Fit Algorithm
2. Best Fit Algorithm
3. Worst Fit Algorithm

The main memory has been divided into fixed size partitions as-



Let us say the given processes are-

- Process P1 = 357 KB
- Process P2 = 210 KB
- Process P3 = 468 KB

- Process P4 = 491 KB

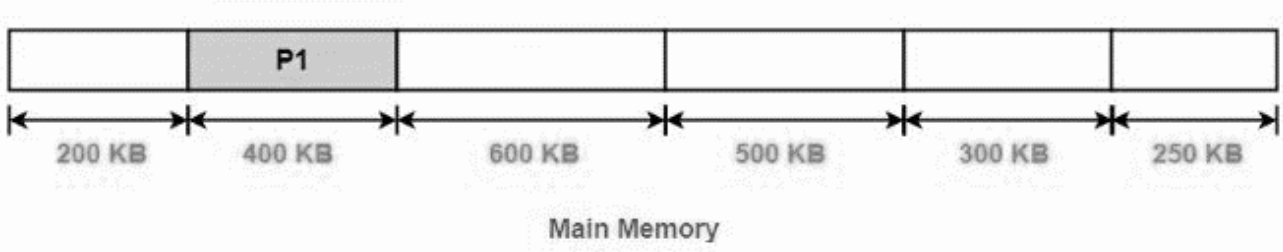
Allocation Using First Fit Algorithm-

In First Fit Algorithm,

- Algorithm starts scanning the partitions serially.
- When a partition big enough to store the process is found, it allocates that partition to the process.

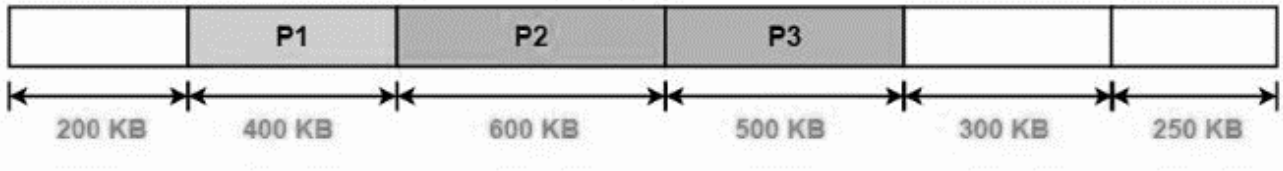
The allocation of partitions to the given processes is shown below-

Step-01:



Step-02:

Step-03:



Step-04:

- Process P4 can not be allocated the memory.
- This is because no partition of size greater than or equal to the size of process P4 is available.

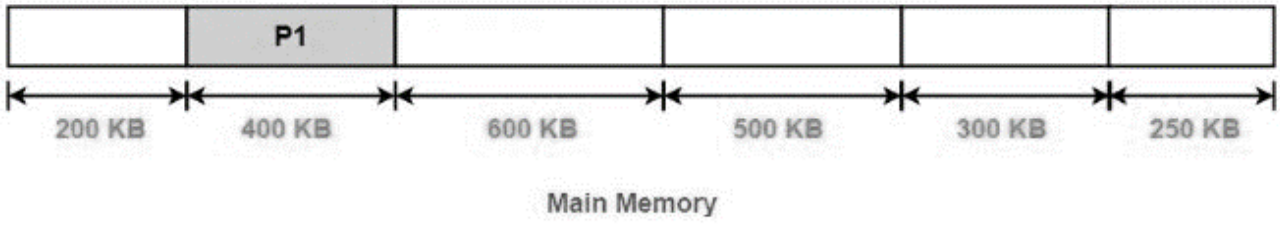
Allocation Using Best Fit Algorithm-

In Best Fit Algorithm,

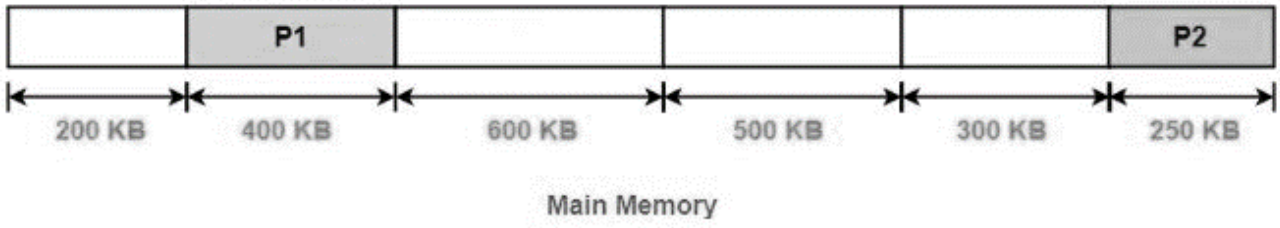
- Algorithm first scans all the partitions.
- It then allocates the partition of smallest size that can store the process.

The allocation of partitions to the given processes is shown below-

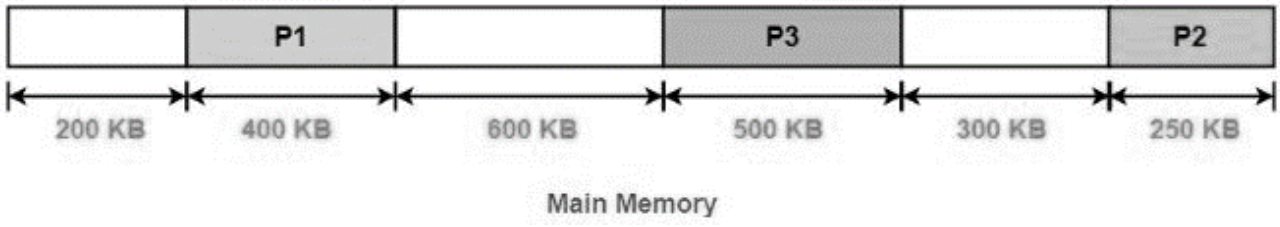
Step-01:



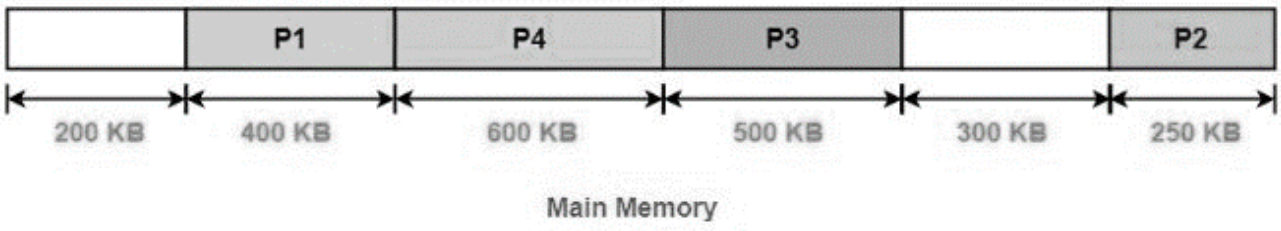
Step-02:



Step-03:



Step-04:



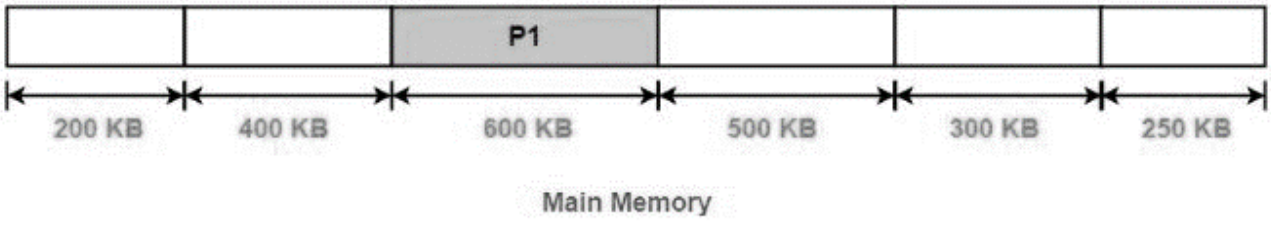
Allocation Using Worst Fit Algorithm-

In Worst Fit Algorithm,

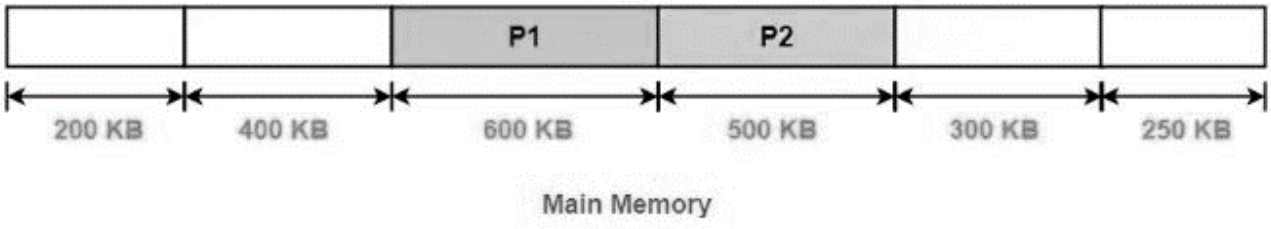
- Algorithm first scans all the partitions.
- It then allocates the partition of largest size to the process.

The allocation of partitions to the given processes is shown below-

Step-01:



Step-02:



Step-03:

- Process P3 and Process P4 can not be allocated the memory.
- This is because no partition of size greater than or equal to the size of process P3 and process P4 is available.

G. Practical related Quiz.

1. Solve examples using First fit, Best fit, Worst fit Algorithms.

Job number	Memory Request
J1	20K
J2	200K
J3	500K
J4	50K
J5	150K

Consider memory partitions of size 50 KB, 300 KB, 75 KB, 700 KB and 100 KB.

Sign

Practical No.4: Solve examples using LRU and FIFO Page replacement algorithm

Objective: In Virtual Memory Management, Page Replacement Algorithms play an important role. The main objective of all the Page replacement policies is to decrease the maximum number of page faults.

A. Expected Program Outcomes (POs)

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.
2. **Problem analysis**: Identify and analyze well-defined engineering problems using codified standard methods
3. **Design/ development of solutions** Design solutions for well-defined technical problems and assist with the design of systems components or processes to meet specified needs
4. **Life-long learning:** Ability to analyze individual needs and engage in updating in the context of technological changes.

B. Expected skills to be developed based on competency:

This practical is expected to develop the following skills for the industry identified competency:

1. Able to understand various page replacement algorithms.
2. Able to solve given examples using LRU and FIFO Page replacement algorithm.

C. Expected Course Outcomes(Cos)

Interpret various memory management techniques.

D. Practical Outcome(Pro)

1. Solve examples using LRU and FIFO Page replacement algorithm.

E. Expected Affective domain Outcome(ADos)

1. Follow precautionary measures.
2. Follow ethical practice.

F. Prerequisite Theory:

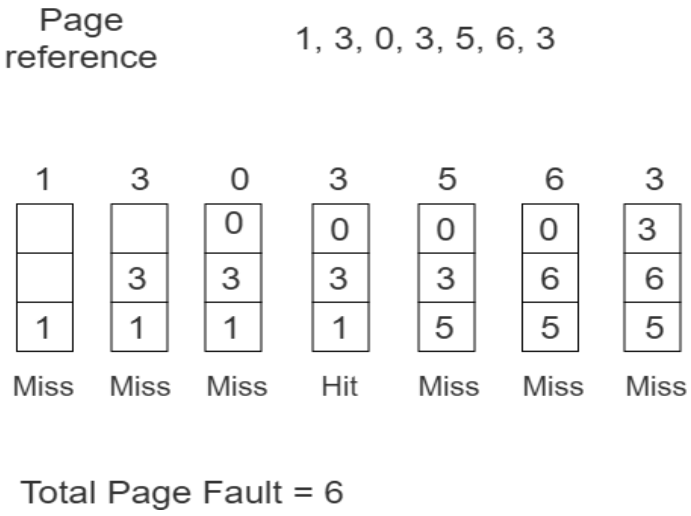
In an operating system that uses paging for memory management, a page replacement algorithm is needed to decide which page needs to be replaced when a new page comes in.

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.

Page Replacement Algorithms:

1. 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.

Example 1: Consider page reference string 1, 3, 0, 3, 5, 6, 3 with 3 page frames.Find the number of page faults.



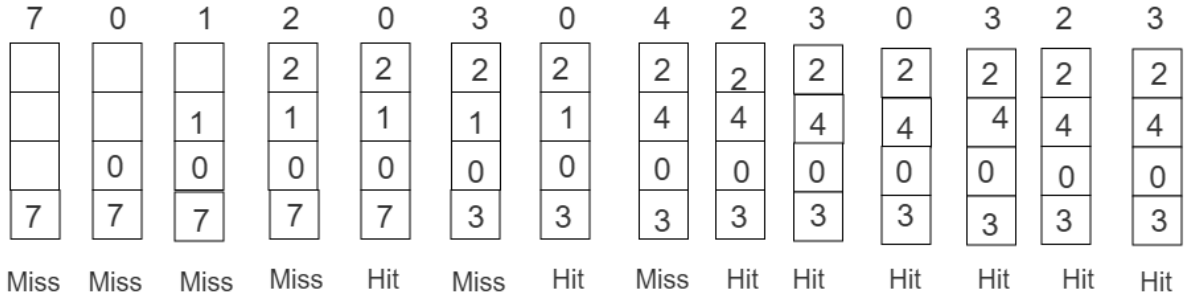
Initially, all slots are empty, so when 1, 3, 0 came they are allocated to the empty slots —> **3 Page Faults.**
when 3 comes, it is already in memory so —> **0 Page Faults.** Then 5 comes, it is not available in memory so it replaces the oldest page slot i.e 1. —>**1 Page Fault.** 6 comes, it is also not available in memory so it replaces the oldest page slot i.e 3 —>**1 Page Fault.** Finally, when 3 come it is not available so it replaces 0 **1 page fault.**

Belady's anomaly proves that it is possible to have more page faults when increasing the number of page frames while using the First in First Out (FIFO) page replacement algorithm. For example, if we consider reference strings 3, 2, 1, 0, 3, 2, 4, 3, 2, 1, 0, 4, and 3 slots, we get 9 total page faults, but if we increase slots to 4, we get 10-page faults.

2. Least Recently Used: In this algorithm, page will be replaced which is least recently used.

Example-3: Consider the page reference string 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 3 with 4 page frames. Find number of page faults.

Page reference 7,0,1,2,0,3,0,4,2,3,0,3,2,3 No. of Page frame - 4



Here LRU has same number of page fault as optimal but it may differ according to question.

Initially, all slots are empty, so when 7 0 1 2 are allocated to the empty slots —> 4 Page faults
0 is already their so —> 0 Page fault.
when 3 came it will take the place of 7 because it is least recently used —>1 Page fault
0 is already in memory so —> 0 Page fault.
4 will takes place of 1 —> 1 Page Fault

Now for the further page reference string —> 0 Page fault because they are already available in the memory.

G. Practical related Quiz.

1. Solve Using LRU and FIFO. Consider the page reference string of size 12: 1, 2, 3, 4, 5, 1, 3, 1, 6, 3, 2, 3 with frame size 4(i.e. maximum 4 pages in a frame).

Sign

Practical No.5: Solve examples using SCAN and CSCAN disc scheduling algorithm

Objective: Disc scheduling is an important process in operating systems that determines the order in which disk access requests are serviced. The objective of disc scheduling is to minimize the time it takes to access data on the disk and to minimize the time it takes to complete a disk access request.

A. Expected Program Outcomes (POs)

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.
2. **Problem analysis**: Identify and analyze well-defined engineering problems using codified standard methods
3. **Design/ development of solutions** Design solutions for well-defined technical problems and assist with the design of systems components or processes to meet specified needs
4. **Life-long learning:** Ability to analyze individual needs and engage in updating in the context of technological changes.

B. Expected skills to be developed based on competency

This practical is expected to develop the following skills for the industry identified competency:

1. Understand various disk scheduling algorithms
2. Solve given example with SCAN AND CSCAN algorithm.

C. Expected Course Outcomes(COs)

Apply File management techniques.

D. Practical Outcome(Pros)

1. Solve given example with SCAN AND CSCAN algorithm.

E. Expected Affective domain Outcome(ADos)

1. Follow precautionary measures.
2. Follow ethical practice.

F. Prerequisite Theory:

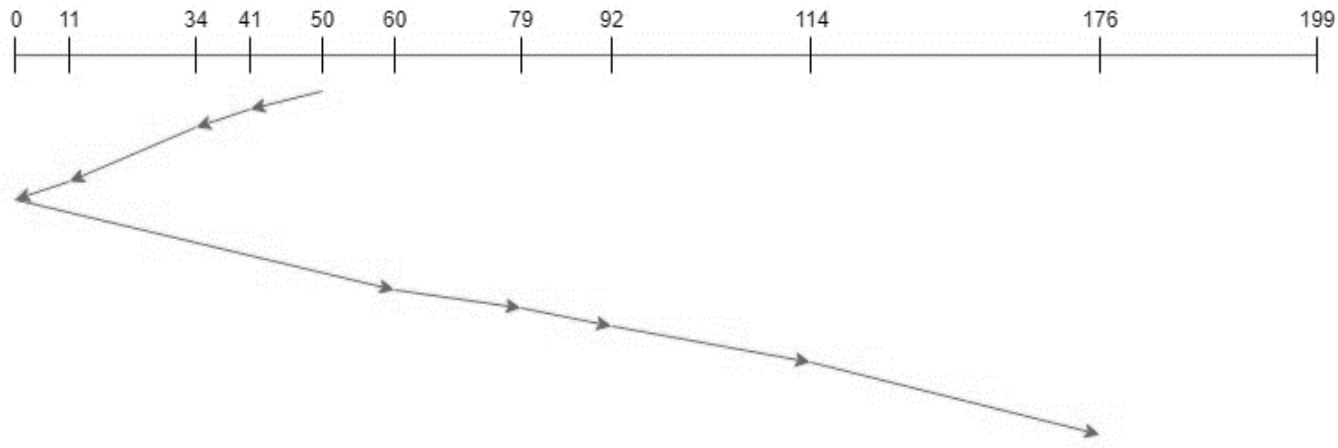
Scan Algorithm

It is also called as Elevator Algorithm. In this algorithm, the disk arm moves into a particular direction till the end, satisfying all the requests coming in its path, and then it turns back and moves in the reverse direction satisfying requests coming in its path.

It works in the way an elevator works, elevator moves in a direction completely till the last floor of that direction and then turns back.

Example
Input:
Request sequence = {176, 79, 34, 60, 92, 11, 41, 114}
Initial head position = 50
Direction = left (We are moving from right to left)

Output:
Total number of seek operations = 226
Seek Sequence is
41
34
11
0
60
79
92
114
176



Therefore, the total seek count is calculated as:

= (50-41)+(41-34)+(34-11)

+(11-0)+(60-0)+(79-60)

+(92-79)+(114-92)+(176-114)

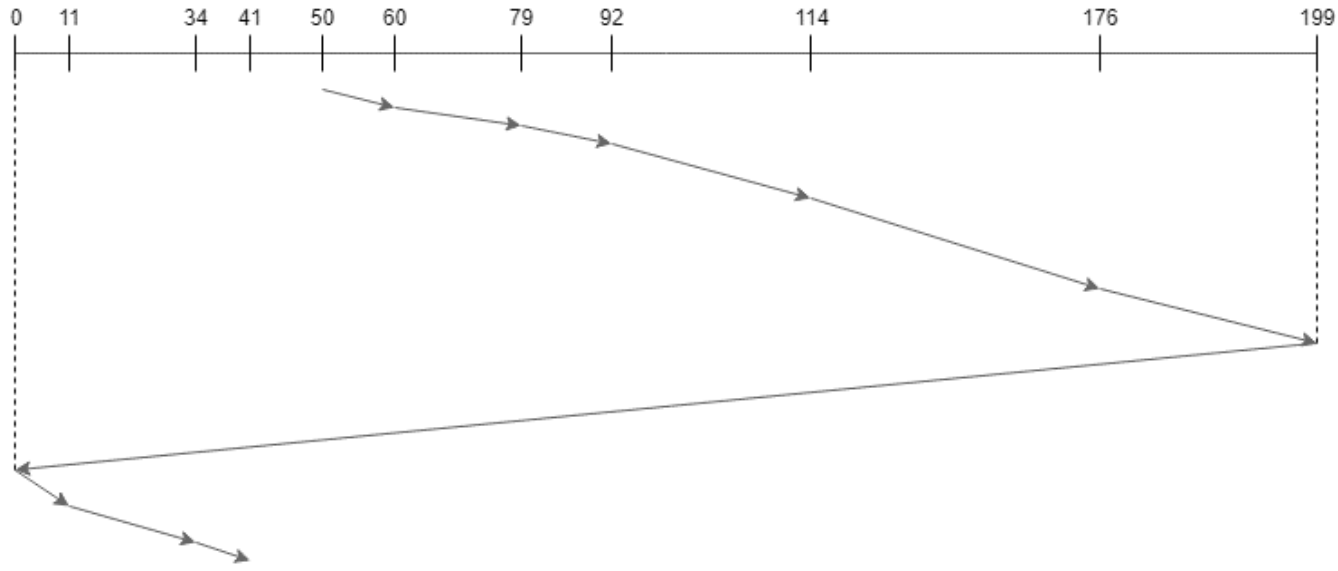
= 226

C-SCAN algorithm

In C-SCAN algorithm, the arm of the disk moves in a particular direction servicing requests until it reaches the last cylinder, then it jumps to the last cylinder of the opposite direction without servicing any request then it turns back and start moving in that direction servicing the remaining requests.

Example
Input:
Request sequence = {176, 79, 34, 60, 92, 11, 41, 114}
Initial head position = 50
Direction = right(We are moving from left to right)

Output:
Initial position of head: 50
Total number of seek operations = 389
Seek Sequence is
60
79
92
114
176
199
0
11
34
41



Therefore, the total seek count is calculated as:

= (60-50)+(79-60)+(92-79)

+(114-92)+(176-114)+(199-176)+(199-0)

+(11-0)+(34-11)+(41-34)

= 389

G. Practical related Quiz.

1. Consider a disc with 200 tracks (0-199) and a disc queue with the following input/output requests: 75, 90, 40, 135, 50, 170, 65, 10. The Read/Write head's initial position is 46,

and it will move to the left-hand side. Using the SCAN method, determine the total number of track movements of the Read/Write head.

2. Consider a disc with 200 tracks (0-199) and a disc queue with the following input/output requests: 75, 90, 40, 135, 170, 65, 10, 45. The Read/Write head's initial position is 55, and it will move to the right-hand side. Using the C-SCAN method, determine the total number of track m

Sign

Practical No.6: Test and run basic Unix commands

Objective: Unix is a multi-user operating system which allows more than one person to use the computer resources at a time. It was originally designed as a time-sharing system to serve several users simultaneous. Unix allows direct communication with the computer via a terminal, hence being very interactive and giving the user direct control over the computer resources.

A. Expected Program Outcomes (POs)

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.
2. **Problem analysis**: Identify and analyze well-defined engineering problems using codified standard methods
3. **Design/ development of solutions** Design solutions for well-defined technical problems and assist with the design of systems components or processes to meet specified needs
4. **Engineering Tools, Experimentation and Testing:** Apply modern engineering tools and appropriate technique to conduct standard tests and measurements
5. **Life-long learning:** Ability to analyze individual needs and engage in updating in the context of technological changes.

B. Expected skills to be developed based on competency

This practical is expected to develop the following skills for the industry identified competency:

1. Able to execute basic UNIX commands.

C. Expected Course Outcomes(COs)

Execute basic Linux commands and Shell scripts.

D. Practical Outcome(Pros)

1. Test and run basic Unix commands

E. Expected Affective domain Outcome(ADos)

1. Follow precautionary measures.
2. Follow ethical practice.

F. Prerequisite Theory:

1) cal: Print the calendar of any year in the range 1 to 9999

a) The system not stores these calendar in memory but generate through the calculation when we invoke command

Options

b) If we want to see the calendar of any year then

$ Cal 2006

c) If we want to see the calendar for particular month then

$ Cal 3 2006

2) who:- List all the users who are currently logged in, with their terminal and their log in time.

Option:

$ who am i:

It displays our login name, terminal name and log in time.

3) ls: directory.

It lists all the filenames in alphabetical order which are present in current

Option:

a) ls –a : display hidden files also

b) ls [aeiou]* : display all files which has first letter any from the bracket.

c) ls –l : It list the files with 7 columns having information about it's permission ,number of links, owner name, group name, size of file in bytes, date and time when file was last modified & filename.

4) cat: It is used to create a new file. Press ctrl + d to indicate EOF.

a) $ cat > test Crete a filename with test

b) $ cat test Display the content of test file

c) $ cat file1 file2 > file3

This would create a new file and content of file1 & file2 are copied into it.

5) wc: It counts the number of lines , words and character from the specified file or files

6) pipe :

a) Wc –l : display the number of lines

b) Wc-w : display the number of words

c) Wc-c : display the number of character.

We can join the commands using a pipe, means it sends the output of one command as input of another.

$ ls | wc-l

Here the output of ls becomes the input to wc which promptly counts number of lines it receives as input and display this count on the screen.

7) echo:

It is used to display output. If we use "" in echo then the shell treats the content with in " " as a single string to be displayed.

8) exit: It will stop the execution of shell script and exit form the program.

9) type: It returns the location of the given command

10) man: This provide the help manual for every commands

$ Man cd It display the help manual for cd command.

11) date: It display date on screen
. $ date
It display Wed mar 08 04:40:10 IST 2006 Where IST means Indian standard time.
The output of the date command can be modified by a variety of Switche.
For e.g. $date '+DATE:%d-%m-%y%n TIME :%H:%M:%S'
Where %d, %m, %y indicate day, month, year and %h, %m, %s indicate hour, minute and second.

12) bc: Calculator can be invoked by typing bc at shell prompt.
$ bc Sqrt(25) 5

13) script: When you write script on the prompt, all the commands which you run after it will be stored in a file named type script.
You can close the file by writing exit on the prompt.

14) ln: It is used to create another link for a same file.
$ ln file1 file2
This command establishes one more link for the file1 in the form of the name file2.

15) nl: This command displays the content of the file with the line number.
$ nl abc
This command display the content of files abc with line number before every line.

16) head: It helps in viewing lines at the beginning of file.
If we not specify anything then this command display first 10 lines For e.g. $ head abc
$ head *-15 abc
This command display first 15 lines of file abc.

17) tail: It helps in viewing lines from the end of file
For e.g. $ tail -20 abc will display the last 20 lines from the file abc.

18) grep: grep stands for "globally search a regular expression and print it."
This command search for the specified input fully for a match with the supplied pattern and display it.

G. Practical Related Quiz:

1. Execute cal, who, Is, cat, wc and pipe commands.
2. Execute echo, exit, date , time, man, ln and nl commands.
3. Execute head, tail and grep commands.

Sign

Practical No.7: Test and run Advance Unix commands

Objective: Unix is a multi-user operating system which allows more than one person to use the computer resources at a time. It was originally designed as a time-sharing system to serve several users simultaneous. Unix allows direct communication with the computer via a terminal, hence being very interactive and giving the user direct control over the computer resources.

A. Expected Program Outcomes (POs)

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.
2. **Problem analysis**: Identify and analyze well-defined engineering problems using codified standard methods
3. **Design/ development of solutions** Design solutions for well-defined technical problems and assist with the design of systems components or processes to meet specified needs
4. **Engineering Tools, Experimentation and Testing:** Apply modern engineering tools and appropriate technique to conduct standard tests and measurements
5. **Life-long learning:** Ability to analyze individual needs and engage in updating in the context of technological changes.

B. Expected skills to be developed based on competency

This practical is expected to develop the following skills for the industry identified competency:

1. Able to execute advance UNIX commands.

C. Expected Course Outcomes(COs)

Execute basic Linux commands and Shell scripts.

D. Practical Outcome(Pros)

1. Test and run Advance Unix commands

E. Expected Affective domain Outcome(ADos)

1. Follow precautionary measures.
2. Follow ethical practice.

F. Prerequisite Theory:

1) grep

   $ grep xy abc

   This would search the word xy in a file called abc

2) sort:

   $ sort abc

   It is used to sort the content of a file.

   It shows the content of file abc in sorted format

$ sort file1 file2 file3

This will sort the contents of several files at once.

3) cut:

It cuts or picks up a given number of character or fields from the specified file Suppose we have a large database of student information from that we want only specific field say name (second field) and division (fifth field) then we can write the cut command as

$ cut –f 2,7 empinfo

If we want to view the field 2 through 7 then

$ cut –f 2-7 empinfo

4) passwd:

You can change your password whenever you are logged in by using the

passwd command as below.

$ passwd

This command asks you to enter old password to prove that you are the authorized person and then after it ask for the new password.

5) pwd:

Pwd stands for 'present working directory'. When you write pwd on prompt it will display current working directory.

For e.g. $ pwd

6) cd:

This command is used to change the directory.

$ cd newdir This command would take you in new directory.

$ cd

When given without any argument is interpreted by the shell as a request to change over the current user's home directory.

7) mkdir:

This command is used to create a new directory.

8) rm:

$ mkdir xyz

This will create directory named xyz.

$ mkdir –p xyz/abc

-p option tell to create xyz directory first and then create directory abc.

This command removes the given file or files supplied to it.

$ rm -i file1

Where –i is a switch , removes file interactively; means you are asked for confirmation before deleting the file.

$ rm -r dir1

This command recursively removes all content of dir1 and also dir1 itself

9) cp:

This command is used to copy a file.

$ cp file1 file2

This will copy file1 in to file2. if file2 does not exit then it will be created. We can copy more than one file into a directory.

$ cp file1 file2 dir1

10) mv:

This command is used to renamethe file

$ mv file1 file2

This command renames the file1 to file2

$mv file1 file2 dir1

This command moves the file1 and file2 from its original location to the directory dir.

G. Practical Related Quiz:

1. Execute grep, sort, cut, passwd and pwd commands.
2. Execute cd, mkdir, rm, cp and mv commands

Sign

Practical No.8: Test commands related to File editing with Vi, Vim, gedit, gcc

Objective Unix is a multi-user operating system which allows more than one person to use the computer resources at a time. It was originally designed as a time-sharing system to serve several users simultaneous. Unix allows direct communication with the computer via a terminal, hence being very interactive and giving the user direct control over the computer resources.

A. Expected Program Outcomes (POs)

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.
2. **Problem analysis**: Identify and analyze well-defined engineering problems using codified standard methods
3. **Design/ development of solutions** Design solutions for well-defined technical problems and assist with the design of systems components or processes to meet specified needs
4. **Engineering Tools, Experimentation and Testing:** Apply modern engineering tools and appropriate technique to conduct standard tests and measurements
5. **Life-long learning:** Ability to analyze individual needs and engage in updating in the context of technological changes.

B. Expected skills to be developed based on competency

This practical is expected to develop the following skills for the industry identified competency:

1. Able to execute file related commands.

C. Expected Course Outcomes(COs)

Execute basic Linux commands and Shell scripts.

D. Practical Outcome(Pros)

1. Test commands related with File editing with Vi, Vim, gedit, gcc

E. Expected Affective domain Outcome(ADos)

1. Follow precautionary measures.
2. Follow ethical practice.

F. Prerequisite Theory:

- **VI AND VIM:**

The vi editor is available on almost all Unix systems. vi can be used from any type of terminal because it does not depend on arrow keys and function keys--it uses the standard alphabetic keys for commands.

vi (pronounced "vee-eye") is short for "vi"sual editor. It displays a window into the file being edited that shows 24 lines of text. vi is a text editor, not a "what you see is what you get" word processor. vi lets you add, change, and delete text, but does not provide such formatting capabilities as centering lines or indenting paragraphs.

This help note explains the basics of vi:

- opening and closing a file

- moving around in a file
- elementary editing

- **STARTING VI :**

You may use vi to open an already existing file by typing

vi filename

where "filename" is the name of the existing file. If the file is not in your current directory, you must use the full pathname or you may create a new file by typing

vi newname

where "newname" is the name you wish to give the new file.

- To open a new file called "testvi," enter

vi testvi

On-screen, you will see blank lines, each with a tilde (~) at the left, and a line at the bottom giving the name and status of the new file:

"testvi" [New file]

- **VI MODES**:

vi has two modes:

- command mode
- insert mode

In command mode, the letters of the keyboard perform editing functions (like moving the cursor, deleting text, etc.). To enter command mode, press the escape

<Esc> key.

In insert mode, the letters you type form words and sentences. Unlike many word processors, vi starts up in command mode.

- **Entering Text**

In order to begin entering text in this empty file, you must change from command mode to insert mode. To do this, type i

Nothing appears to change, but you are now in insert mode and can begin typing text. In general, vi's commands do not display on the screen and do not require the Return key to be pressed.

Type a few short lines and press <Return> at the end of each line. If you type a long line, you will notice the vi does not word wrap, it merely breaks the line unceremoniously at the edge of the screen.

If you make a mistake, pressing <Backspace> or <Delete> may remove the error, depending on your terminal type.

- **Moving the Cursor:**

To move the cursor to another position, you must be in command mode. If you have just finished typing text, you are still in insert mode. Go back to command mode by pressing <Esc>. If you are not sure which mode you are in, press <Esc> once or twice until you hear a beep. When you hear the beep, you are in command mode.

The cursor is controlled with four keys: h, j, k, l.

Key	Cursor Movement
h	left one space
j	down one line

k up one line

l right one space

When you have gone as far as possible in one direction, the cursor stops moving and you hear a beep. For example, you cannot use l to move right and wrap around to the next line, you must use j to move down a line. See the section entitled "Moving Around in a File" for ways to move more quickly through a file.

- **Basic Editing:**

Editing commands require that you be command mode. Many of the editing commands have a different function depending on whether they are typed as upper- or lowercase. Often, editing commands can be preceded by a number to indicate a repetition of the command.

- **Deleting Characters:**

To delete a character from a file, move the cursor until it is on the incorrect letter, then type x

The character under the cursor disappears. To remove four characters (the one under the cursor and the next three) type 4x

To delete the character before the cursor, type X (uppercase)

- **Deleting Words :**

To delete a word, move the cursor to the first letter of the word, and type Dw

This command deletes the word and the space following it. To delete three words type 3dw

- **Deleting Lines :**

To delete a whole line, type Dd

The cursor does not have to be at the beginning of the line. Typing dd deletes the entire line containing the cursor and places the cursor at the start of the next line. To delete two lines, type 2dd

To delete from the cursor position to the end of the line, type D (uppercase)

- **Replacing Characters :**

To replace one character with another:

1. Move the cursor to the character to be replaced.
2. Type r
3. Type the replacement character.

The new character will appear, and you will still be in command mode.

- **Replacing Words :**

To replace one word with another, move to the start of the incorrect word and type Cw

The last letter of the word to be replaced will turn into a $. You are now in insert mode and may type the replacement. The new text does not need to be the same length as the original. Press <Esc> to get back to command mode. To replace three words, type

3cw

- **Replacing Lines :**

To change text from the cursor position to the end of the line:

1. Type C (uppercase).

2. Type the replacement text.
3. Press <Esc>.

- **Inserting Text :**

To insert text in a line:

1. Position the cursor where the new text should go.
2. Type i
3. Enter the new text.
   The text is inserted BEFORE the cursor.
4. Press <Esc> to get back to command mode.

- **Appending Text:**

To add text to the end of a line:

1. Position the cursor on the last letter of the line.
2. Type a
3. Enter the new text.
   This adds text AFTER the cursor.
4. Press <Esc> to get back to command mode.

- **Opening a Blank Line :**

To insert a blank line below the current line, type
o (lowercase)
To insert a blank line above the current line, type
O (uppercase)

- **Joining Lines:**

To join two lines together:

1. Put the cursor on the first line to be joined.
2. Type J

To join three lines together:

1. Put the cursor on the first line to be joined.
2. Type 3J

- **Undoing :**

To undo your most recent edit, type U

To undo all the edits on a single line, type U (uppercase)

Undoing all edits on a single line only works as long as the cursor stays on that line. Once you move the cursor off a line, you cannot use U to restore the line.

- **Moving Around in a File :**

There are shortcuts to move more quickly though a file. All these work in command mode

Key	Movement
w	forward word by word
b	backward word by word
$	to end of line
0 (zero)	to beginning of line
H	to top line of screen
M	to MIDDLE line of screen
L	to last line of screen
G	to last line of file
1G	to first line of file
<Control>f	scroll forward one screen
<Control>b	scroll backward one screen
<Control>d	scroll down one-half screen
<Control>u	scroll up one-half screen

- **Moving by Searching :**

To move quickly by searching for text, while in command mode:

1. Type / (slash).
2. Enter the text to search for.
3. Press <Return>.

The cursor moves to the first occurrence of that text. To repeat the search in a forward direction, type
N

To repeat the search in a backward direction, type
N

- **Closing and Saving a File :**

With vi, you edit a copy of the file, rather than the original file. Changes are made to the original only when you save your edits.
To save the file and quit vi, type
ZZ

The vi editor editor is built on an earler Unix text editor called ex. ex commands can be used within vi. ex commands begin with a : (colon) and end with a

<Return>. The command is displayed on the status line as you type. Some ex commands are useful when saving and closing files.

To save the edits you have made, but leave vi running and your file open:

1. Press <Esc>.
2. Type :w
3. Press <Return>.

To quit vi, and discard any changes your have made since last saving:

1. Press <Esc>.
2. Type :q!
3. Press <Return>.

- **GEDIT:**

1. It is a GUI based text editor and can be used with Linux as well as otheroperating system such as Windows, Mac OS X
2. It is similar to 'notepad' editor of windows operating system. But it is far more powerful than 'notepad'.
3. It is frtee software and provides simplicity and ease of use.
4. It includes syntax highlighting for variousprogram code and texr mark-up formats.
5. It also has GUI tabs for editing multiple files. Users can work with multiple files simultaneously using these tabs.
6. It supports a full undo and redo system as well as search and replace. Other typical code oriented features include line numbering, backetmatching, textwrapping and current line highlighting, automaticindentation and automatic file backup.
7. It also supports multi-language spell checking and a flexible plug –in system allowing the addition of new features.

- **GCC:**

1. GCC refers to GNU Collection
2. Linus Thorvaldsen used gcc while developing Linux kernel to compiles his programs.
3. GCC is a compiler for c,c++,java ,fortan and other programming languages that can be used in Linux.
4. Here steps are given to compile a shell script.

[i] Develpp a simple shell script to display "hello " using ant of the above text editors and store it in a file named hello.sh.

[ii] Compile the program by following code.

gcc hello.sh –o hello.out

[iii] Run the program by following syntax

. / hello.out

G. Practical Related Quiz:

Write Unix Command For Following Questions:

To insert new text	
To save file	
To quit the vi editor	
To quit without saving	
To save and quit vi editor	
To continue with search	
To copy the line where cursor is located	
To paste the text just deleted or copied at the cursor	
To delete entire line where cursor is located	
To delete word from cursor position	
To delete 3 lines	
To delete character at cursor	
To change word	
To terminate insert mode	

Sign

Practical No.9: Create a shell script to read from command line and print "Hello".

Objective: Shell scripting is meant to be simple and efficient. It uses the same syntax in the script as it would on the shell command line, removing any interpretation issues. Writing code for a shell script is also faster and requires less of learning curve than other programming languages

A. Expected Program Outcomes (POs)

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.
2. **Problem analysis**: Identify and analyze well-defined engineering problems using codified standard methods
3. **Design/ development of solutions** Design solutions for well-defined technical problems and assist with the design of systems components or processes to meet specified needs
4. **Engineering Tools, Experimentation and Testing:** Apply modern engineering tools and appropriate technique to conduct standard tests and measurements
5. **Life-long learning:** Ability to analyze individual needs and engage in updating in the context of technological changes.

B. Expected skills to be developed based on competency

This practical is expected to develop the following skills for the industry identified competency:

1. Able to create shell script.

C. Expected Course Outcomes(COs)

Execute basic Linux commands and Shell scripts.

D. Practical Outcome(Pros)

1. Create a shell script to read from command line and print "Hello".

E. Expected Affective domain Outcome(ADos)

1. Follow precautionary measures.
2. Follow ethical practice.

F. Prerequisite Theory:

- **First Shell Script:**

Create a file (first.sh) as follows:

first.sh

```
#!/bin/sh
# This is a comment!
echo Hello World        # It prints Hello World
```

- The first line tells Unix that the file is to be executed by /bin/sh. If you're using GNU/Linux, /bin/sh is normally a symbolic link to bash (or, more recently, dash).
- The second line begins with a special symbol: **#**. This marks the line as a comment, and it is ignored completely by the shell
- The third line runs a command: **echo**, with two parameters, or arguments - the first is **"Hello"**; the second is **"World"**.
- run **chmod 755** first.sh to make the text file executable, and run **./first.sh**.

```
$ chmod 755 first.sh
$ ./first.sh
  Hello World
```

G. Practical Related Quiz:

1. Create a shell script to read from command line and print “Hello”.

Sign

Practical No.10: Create a Shell script to read and display content of a file. And append content of one file to another

Objective: Shell scripting is meant to be simple and efficient. It uses the same syntax in the script as it would on the shell command line, removing any interpretation issues. Writing code for a shell script is also faster and requires less of learning curve than other programming languages

A. Expected Program Outcomes (POs)

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.
2. **Problem analysis**: Identify and analyze well-defined engineering problems using codified standard methods
3. **Design/ development of solutions** Design solutions for well-defined technical problems and assist with the design of systems components or processes to meet specified needs
4. **Engineering Tools, Experimentation and Testing:** Apply modern engineering tools and appropriate technique to conduct standard tests and measurements
5. **Life-long learning:** Ability to analyze individual needs and engage in updating in the context of technological changes.

B. Expected skills to be developed based on competency

This practical is expected to develop the following skills for the industry identified competency:

1. Able to create a Shell script to read and display content of a file. And append content of one file to another

C. Expected Course Outcomes(COs)

Execute basic Linux commands and Shell scripts.

D. Practical Outcome(Pros)

1. Create a Shell script to read and display content of a file. And append content of one file to another

E. Expected Affective domain Outcome(ADos)

1. Follow precautionary measures.
2. Follow ethical practice.

F. Prerequisite Theory:

- **Read and display of a file:**

  Cat (concatenate) command is very frequently used in Linux. It reads data from the file and gives their content as output. It helps us to create, view, and concatenate files.

  - **To view a single file**

    $cat filename

    It will show content of given filename

- **Append content of file:**
    - You can use **cat** with redirection to append a file to another file. You do this by using the append redirection symbol, ``>>". To append one file to the end of another, type **cat**, the file you want to append, then >>, then the file you want to append to, and press <Enter>.
    - For example, to append a file called *report2* to the end of *report1*, type:

      cat report2 >> report1
      You can use the append symbol ``>>" with any command that writes output.

G. Practical Related Quiz:

1. Create a Shell script to read and display content of a file. And append content of one file to another

Sign

Practical No.11: Create a Shell script to accept a string in lower case letters from a user, & convert to upper case letters.

Objective: Shell scripting is meant to be simple and efficient. It uses the same syntax in the script as it would on the shell command line, removing any interpretation issues. Writing code for a shell script is also faster and requires less of learning curve than other programming languages

A. Expected Program Outcomes (POs)

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.
2. **Problem analysis**: Identify and analyze well-defined engineering problems using codified standard methods
3. **Design/ development of solutions** Design solutions for well-defined technical problems and assist with the design of systems components or processes to meet specified needs
4. **Engineering Tools, Experimentation and Testing:** Apply modern engineering tools and appropriate technique to conduct standard tests and measurements
5. **Life-long learning:** Ability to analyze individual needs and engage in updating in the context of technological changes.

B. Expected skills to be developed based on competency

This practical is expected to develop the following skills for the industry identified competency:

1. Able to create a Shell script to accept a string in lower case letters from a user, & convert to upper case letters.

C. Expected Course Outcomes(COs)

Execute basic Linux commands and Shell scripts.

D. Practical Outcome(Pros)

1. Create a Shell script to accept a string in lower case letters from a user, & convert to upper case letters.

E. Expected Affective domain Outcome(ADos)

1. Follow precautionary measures.
2. Follow ethical practice.

F. Prerequisite Theory:

- **tr command:**

Use the tr command to convert all incoming text / words / variable data from upper to lower case or vise versa (translate all uppercase characters to lowercase).

```
$ echo Hello There | tr [:lower:] [:upper:]
HELLO THERE
```

- **Taking value from** the user and convert it

```
#!/bin/bash

echo -n "Enter department name: "
read dept
echo $dept | tr [:lower:] [:upper:]
```

G. Practical Related Quiz:

1. Create a Shell script to accept a string in lower case letters from a user, & convert to upper case letters.

Sign

Practical No.12: Create a Shell script for adding two numbers and print table of a given number
Objective: Shell scripting is meant to be simple and efficient. It uses the same syntax in the script as it would on the shell command line, removing any interpretation issues. Writing code for a shell script is also faster and requires less of learning curve than other programming languages

A. Expected Program Outcomes (POs)

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.
2. **Problem analysis**: Identify and analyze well-defined engineering problems using codified standard methods
3. **Design/ development of solutions** Design solutions for well-defined technical problems and assist with the design of systems components or processes to meet specified needs
4. **Engineering Tools, Experimentation and Testing:** Apply modern engineering tools and appropriate technique to conduct standard tests and measurements
5. **Life-long learning:** Ability to analyze individual needs and engage in updating in the context of technological changes.

B. Expected skills to be developed based on competency

This practical is expected to develop the following skills for the industry identified competency:

1. Able to create a Shell script for adding two numbers and print table of a given number

C. Expected Course Outcomes(COs)

Execute basic Linux commands and Shell scripts.

D. Practical Outcome(Pros)

1. Create a Shell script for adding two numbers and print table of a given number

E. Expected Affective domain Outcome(ADos)

1. Follow precautionary measures.
2. Follow ethical practice.

F. Prerequisite Theory:

- **UNIX Shell - The for Loop:**

The for loop operate on lists of items. It repeats a set of commands for every item in a list.

SYNTAX:

for var in word1 word2 ... word N do

Statement(s) to be executed for every word. done

Here var is the name of a variable and word1 to word N are sequences of characters separated by spaces (words). Each time the for loop executes, the value of the variable var is set to the next word in the list of words, word1 to word N.

Example:

Here is a simple example that uses for loop to span through the given list of numbers:

```
for var in 0 1 2 3 4 5 6 7 8 9 do
echo $var done
```

- **UNIX Shell - The while Loop:**

The while loop enables you to execute a set of commands repeatedly until some condition occurs. It is usually used when you need to manipulate the value of a variable repeatedly.

SYNTAX:

while command do

Statement(s) to be executed if command is true done

Here Shell command is evaluated. If the resulting value is true, given statement(s) are executed. If command is false then no statement would be not executed and program would jump to the next line after done statement.

Example:

Here is a simple example that uses the while loop to display the numbers zero to nine:

```
a=0
while[ $a -lt 10] do
echo $a
a=`expr $a + 1` done
```

- **UNIX Shell - The Until Loop:**

The while loop is perfect for a situation where you need to execute a set of commands while some condition is true. Sometimes you need to execute a set of commands until a condition is true.

SYNTAX:

Until command Do

Statement(s) to be executed until command is true Done

Here Shell command is evaluated. If the resulting value is false, given statement(s) are executed. If command is true then no statement would be not executed and program would jump to the next line after done statement.

Example:

Here is a simple example that uses the until loop to display the numbers zero to nine:

```
a=0
until[! $a -lt 10] do
echo $a a=`expr $a + 1` done
```

G. Practical Related Quiz:

1. Create a Shell script to add two numbers.
2. Create a Shell script to print table of a number.

Sign

Basics of Operating System

4330703

Lab manual is prepared by
Mrs Avani Galathiya
Lecturer
R.C. Technical Institute, Ahmedabad

&

Mrs Soniya Dadhania
Lecturer
R.C. Technical Institute, Ahmedabad

Branch Coordinator
Shri B H Kantewala
HOD - CE
Government Polytechnic, Ahmedabad

Committee Chairman
Shri R. D. Raghani
(HOD-EC)
Principal (I/C)
Government Polytechnic, Gandhinagar