Question 1

a. Define how the operating system works.

Ans :- The operating system (OS) is a crucial software component that manages computer hardware and software resources and provides common services for computer programs. It acts as an intermediary between applications and hardware, facilitating communication and coordination between them. The OS performs various tasks such as managing memory, input/output devices, file systems, and controlling peripheral devices like printers and monitors. It also provides a user interface for interacting with the computer system.

b. List any FIVE (5) resources of networked operating system.

Ans :- Five resources of a networked operating system include:

Network Interface :- This resource enables computers to connect to networks, facilitating communication with other devices and systems. It encompasses both hardware (like network cards or Wi-Fi adapters) and software protocols (such as TCP/IP) necessary for data transmission over the network.

File System :- Responsible for managing the organization, storage, and retrieval of files on the network. It maintains a structured hierarchy of directories and files, ensuring data integrity, accessibility, and security. The file system manages file metadata, permissions, and access control to regulate user interactions with files.

Security Services :- These services provide essential mechanisms such as authentication, authorization, and encryption to safeguard data and resources from unauthorized access and malicious threats. Authentication verifies the identities of users or devices, while authorization controls their access rights based on predefined policies. Encryption ensures data confidentiality and integrity during transmission and storage, preventing unauthorized interception or tampering.

Printing Services :- Printing services handle the submission, management, and control of print jobs initiated by users on the network. They manage print queues, prioritize print jobs, and monitor printer status to ensure efficient printing operations. Additionally, printing services may support features like printer pooling, job scheduling, and accounting for better resource utilization and cost management.

Remote Access Services :- These services enable users to access network resources and applications from remote locations, extending the reach of the network beyond physical boundaries. They provide secure connectivity options such as Virtual Private Networks (VPNs), Remote Desktop Protocol (RDP), or web-based portals. Remote access services ensure data confidentiality, authentication of remote users, and efficient transmission of data over the network, thereby enhancing flexibility and convenience in network usage.

c. Explain THREE (3) tasks that performed by the operating system to support the system components.

Ans :- Three tasks performed by the operating system to support system components are:

Memory Management :- The operating system manages system memory, allocating memory resources to processes and ensuring efficient utilization of available memory. It handles memory allocation, deallocation, and protection, preventing processes from accessing unauthorized memory locations and facilitating multitasking by efficiently swapping data between main memory and secondary storage.

Process Management :- Process management involves the creation, scheduling, and termination of processes or tasks running on the system. The operating system allocates processor time to different processes, prioritizes tasks based on their importance, and manages process synchronization and communication. It also handles process states, context switching, and inter-process communication to ensure smooth execution of concurrent tasks.

Input/Output Management :- Input/output (I/O) management involves controlling and coordinating the interaction between the computer system and its peripheral devices, such as keyboards, mice, displays, disks, and network interfaces. The operating system provides device drivers to interface with hardware components, manages I/O operations, and handles data buffering, error handling, and device communication protocols. It ensures efficient data transfer between the CPU, memory, and I/O devices, optimizing system performance and responsiveness.

d. Given the model of non-networks operating system in Figure 1:

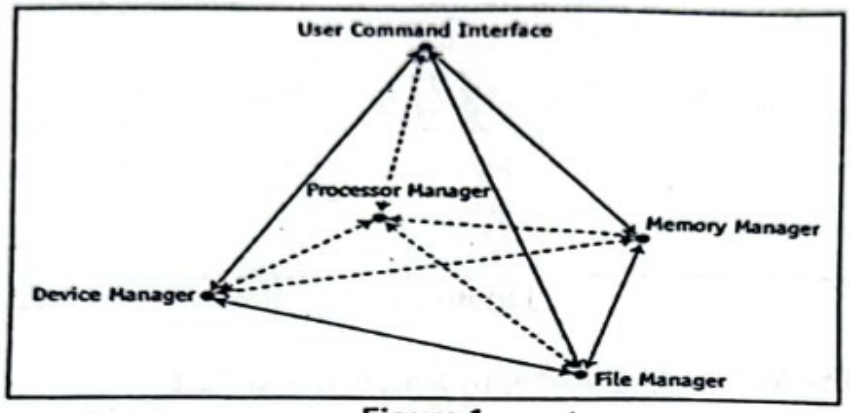


Figure 1

Describe FOUR (4) tasks that each of the sub-system has to perform.

Ans :-

1. User Command Interface

Command Interpretation :- Receives user commands and interprets them to determine the corresponding system operations to be performed.

Input Handling :- Manages input from users, including parsing command arguments, validating input data, and handling input errors or exceptions.

User Feedback :- Provides feedback to users, such as command execution status, error messages, or prompts for additional input.

Interface Customization :- Supports customization of the user interface, allowing users to configure settings, preferences, or interface themes according to their preferences.

2. Device Manager

Device Detection :- Detects and identifies connected hardware devices, such as input/output devices (e.g., keyboard, mouse), storage devices (e.g., disks, USB drives), and communication devices (e.g., network interfaces).

Device Initialization :- Initializes and configures device drivers and hardware interfaces to enable communication and interaction with hardware devices.

Device Control :- Controls device operations, including device activation, deactivation, resetting, and error handling to ensure proper device functioning and prevent system crashes or data loss.

Device Abstraction :- Provides abstraction layers for device access, allowing applications and system components to interact with devices using standardized interfaces, regardless of hardware variations.

3. Process Manager

Process Creation and Termination :- Manages the creation and termination of processes, including allocating resources, setting up process control blocks (PCBs), and handling process cleanup upon termination.

Process Scheduling :- Schedules processes for execution on the CPU, selecting the next process to run based on scheduling algorithms, priorities, and system load.

Process Communication :- Facilitates inter-process communication (IPC) mechanisms, allowing processes to exchange data, synchronize execution, or communicate events using shared memory, message passing, or synchronization primitives.

Process Monitoring and Control :- Monitors process status and performance, detecting and handling process abnormalities (e.g., deadlock, resource contention), and enforcing process policies (e.g., resource quotas, priority levels).

4. Memory Manager

Memory Allocation :- Manages the allocation of memory to processes, allocating memory blocks from the available physical memory and maintaining memory allocation tables or data structures.

Memory Protection :- Implements memory protection mechanisms to prevent unauthorized access or modification of memory regions, enforcing memory access permissions and memory isolation.

Memory Paging and Swapping :- Implements memory paging and swapping strategies to optimize memory usage and accommodate process memory requirements, swapping memory pages between main memory and secondary storage when necessary.

Memory Cleanup and Defragmentation :- Handles memory cleanup and defragmentation tasks, reclaiming memory from terminated processes, consolidating fragmented memory regions, and optimizing memory layout to improve memory utilization and system performance.

Question 2

a. Explain THERE (3) types of scheduler in operating system.

Ans :- Schedulers are responsible for managing the execution of processes on the CPU. There are several types of schedulers, each serving different purposes.

Here are three types of schedulers commonly found in operating systems:

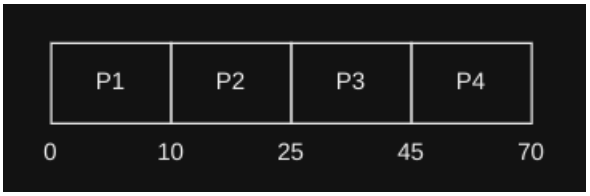
Long-Term Scheduler (Job Scheduler) :- The long-term scheduler is responsible for selecting processes from the pool of new processes and deciding which ones to admit into the system for execution. Its primary goal is to maintain an optimal balance between system throughput and resource utilization by controlling the degree of multiprogramming. Long-term scheduling decisions are based on various factors such as process priority, resource availability, and system load. This scheduler is generally invoked less frequently compared to other types of schedulers since it deals with selecting processes for admission rather than managing their execution.

Short-Term Scheduler (CPU Scheduler) :- The short-term scheduler is responsible for selecting which process from the ready queue will execute next on the CPU. It operates at a rapid pace, making scheduling decisions on a frequent basis (e.g., every few milliseconds) to ensure efficient CPU utilization and responsiveness. The primary goal of the short-term scheduler is to minimize process waiting times, maximize CPU throughput, and maintain fairness among competing processes. Short-term scheduling decisions are typically based on various factors such as process priority, CPU burst characteristics, and scheduling algorithms (e.g., round-robin, shortest job next).

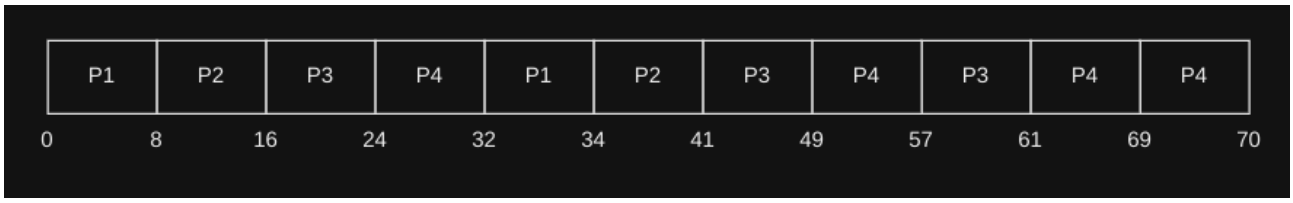
Medium-Term Scheduler :- The medium-term scheduler is an optional component found in some operating systems, particularly those that support process swapping or suspension to secondary storage. It is responsible for selecting processes from the pool of ready processes and deciding which ones to suspend and swap out of main memory, freeing up space for other processes. The medium-term scheduler helps in managing memory resources more efficiently by controlling the degree of multiprogramming and reducing memory contention. Its decisions are based on factors such as memory pressure, process priority, and memory usage patterns, aiming to optimize overall system performance and responsiveness.

b. Given the following processes and burst time in Table 1.

i. Draw a Gantt chart for Priority scheduling algorithm.



ii. Draw a Gantt chart for Round Robin scheduling algorithm with the Time Quantum = 8.



c. Write the Turnaround Time of the scheduling algorithms in Question 2(b)(i) and Question 2(b)(ii).

Process	AT	BT	CT	TAT
P1	0	10	10	10
P2	0	15	25	25
P3	0	20	45	45
P4	0	25	70	70

Process	AT	BT	CT	TAT
P1	0	10	34	34
P2	0	15	41	41
P3	0	20	61	61
P4	0	25	70	70

d. Discuss any TWO (2) benefits of multi-threading in operating system.

Ans :- Multi-threading in operating systems offers several benefits, two of which are:

1. Improved Responsiveness :- Multi-threading allows applications to remain responsive to user interactions even when performing intensive tasks. By dividing the workload into

multiple threads, an application can execute tasks concurrently, enabling it to handle user input, update the user interface, and respond to events without being blocked by long-running operations. For example, in a web browser, multi-threading enables the browser to load web pages, render content, and handle user input simultaneously, providing a smoother and more interactive browsing experience.

2. Increased Utilization of CPU Cores :- Multi-threading maximizes the utilization of modern multi-core processors by allowing multiple threads to execute simultaneously on different CPU cores. This leads to improved performance and throughput, as the operating system can schedule threads to run in parallel, exploiting the available processing power more efficiently. Applications that are multi-threaded can benefit from better scalability and performance on multi-core systems, as they can distribute their workload across multiple cores, reducing processing bottlenecks and achieving higher levels of concurrency. For example, a video encoding application can use multi-threading to divide the encoding process into parallel tasks, leveraging multiple CPU cores to encode different segments of the video concurrently, thereby reducing the overall encoding time.

e. Discuss the Starvation Philosophy shown in Figure 2.

Ans :- Starvation arises in a system when a process is unable to access essential resources due to prolonged contention with other processes. This situation often occurs when certain processes continuously hold onto critical resources, preventing others from obtaining them. For instance, if processes p1, p2, and p3 persistently monopolize resources f1, f2, and f3, respectively, processes p4 and p5 may experience starvation if they depend on these resources but cannot acquire them. Consequently, p4 and p5 remain in a blocked or waiting state indefinitely, unable to proceed with their execution tasks.

The impact of starvation can be severe, leading to degraded system performance, increased response times, and inefficient resource utilization. To mitigate starvation, operating systems implement various strategies such as priority-based resource allocation, fair scheduling policies, and resource preemption mechanisms. These approaches aim to ensure equitable access to resources among processes, prioritize critical tasks, and prevent any single process from monopolizing resources for an extended period. By addressing starvation effectively,

operating systems can maintain system stability, optimize resource utilization, and enhance overall system performance and responsiveness.

Question 3

a. Write about the concept of paging and it's implementation in the memory.

Ans :- Paging is a memory management technique used by modern operating systems to efficiently manage memory and provide virtual memory support to processes. It allows the physical memory (RAM) to be divided into fixed-size blocks called "frames" and the logical memory (address space) of each process to be divided into fixed-size blocks called "pages." Paging enables processes to access memory in smaller, more manageable units, providing several benefits such as efficient memory allocation, improved system performance, and support for virtual memory.

The concept of paging involves several key components and mechanisms:

Page Table :- Each process has its own page table, which is a data structure maintained by the operating system. The page table maps the logical addresses (virtual pages) used by the process to the corresponding physical addresses (frames) in memory. It enables the operating system to translate virtual addresses generated by the CPU into physical addresses.

Page Fault Handling :- When a process attempts to access a page that is not currently present in physical memory, a page fault occurs. The operating system handles page faults by fetching the required page from secondary storage (such as a hard disk) into an available frame in physical memory. Page faults may also occur if the accessed page is marked as invalid or protected.

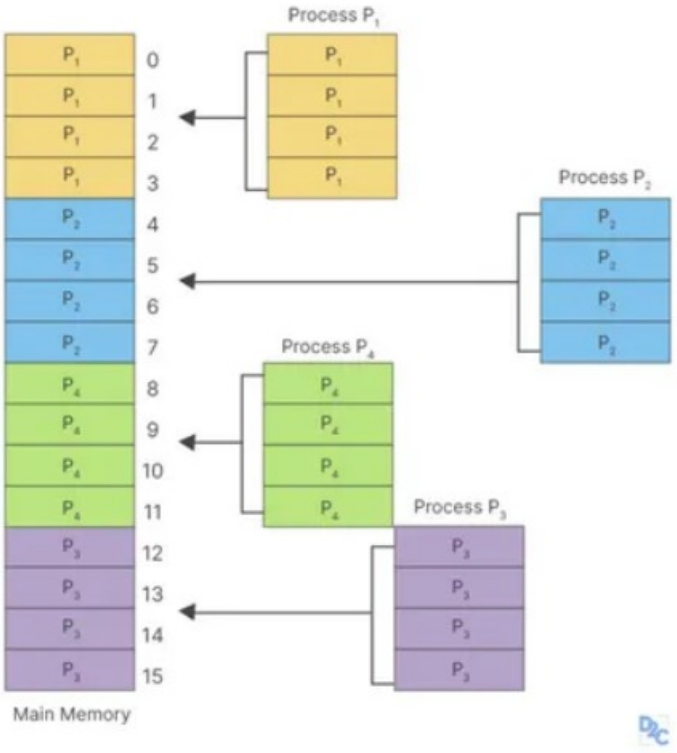
Page Replacement :- If physical memory becomes full and there are no available frames to accommodate a new page, the operating system must select a page to evict from memory to make room for the new page. Page replacement algorithms, such as Least Recently Used (LRU), First-In-First-Out (FIFO), or Clock, are used to select the victim page for replacement based on various criteria such as access history or time since last access.

Memory Management Unit (MMU) :- The MMU is a hardware component responsible for translating virtual addresses generated by the CPU into physical addresses. It uses the

page table to perform this translation and ensures that processes can access the correct physical memory locations. The MMU also enforces memory protection and access control by checking the validity and permissions of page table entries.

TLB (Translation Lookaside Buffer) :- The TLB is a hardware cache that stores recently used page table entries to accelerate address translation. It caches the mappings between virtual pages and physical frames, reducing the overhead of accessing the page table for every memory access.

b. Illustrate a figure that show the shared pages for three processes; P1, P2 and P3 in memory management.



Question 4

a. Define the function of following commands of Ubuntu Linux operating system:

- **pwd**
- **whoami**
- **man login**
- **sudo su-**
- **rm -r**

Ans :-

1. pwd :- Stands for print working directory. This command displays the current working directory in the terminal. It helps users identify their current location within the directory structure of the file system.

2. whoami :- This command simply outputs the username of the current user who is logged in to the terminal. It's useful for quickly identifying the user context within the system.

3. man login :- This command opens the manual page for the login command. The manual pages (or man pages) provide detailed documentation and usage instructions for various commands and utilities in Linux. Using man login allows users to learn about the login command, its options, and how to use it effectively.

4. sudo su- :- This command is used to switch to the root user or superuser (admin) account in Ubuntu Linux. sudo stands for superuser do, and it allows users to execute commands with administrative privileges. su stands for switch user, and the hyphen (-) at the end of su indicates that the root user's environment variables should be used. This command is commonly used when performing system administration tasks that require elevated privileges.

5. rm -r :- The rm command is used to remove or delete files and directories in Linux. The -r option is used to recursively delete directories and their contents. When combined, rm -r deletes the specified directory and all of its subdirectories and files. This command should be used with caution as it permanently deletes data and cannot be undone. It's often used when cleaning up directories or removing unwanted files.

b. Write a single Ubuntu Linux command in octal form to set the following permissions for the file name assignment.

- **rwx rwx rwx**
- **rwx r-- r--**
- **rw- rw- rw-**
- **rw- --- ---**
- **rwx r-w ---**
- **rwx --- ---**
- **rwx rw- r--**
- **rw- rw- r--**

Ans :-

- rwx rwx rwx = chmod 777
- rwx r-- r-- = chmod 744
- rw- rw- rw- = chmod 666
- rw- --- --- = chmod 600
- rwx r-w --- = chmod 760
- rwx --- --- = chmod 700
- rwx rw- r-- = chmod 764
- rw- rw- r-- = chmod 664

c. Assume that the current working directory is /home/wanashekin. Write a single Linux commands for the following statements:

i. User Wanashekin wants to remove two empty directories called EC2339 Final Examination and EC3123_Lab.

Ans :- rm -r /home/wanashekin/EC2339 Final Examination /home/wanashekin/EC3123_Lab

ii. User Wanashekin wants to create a directory called MyDiary.

Ans :- mkdir /home/wanashekin/MyDiary

iii. User Wanashekin wants to create a file called Monthly_Activity in directory MyDiary.

Ans :- touch /home/wanashekin/MyDiary/Monthly_Activity

iv. User Wanashekin wants to create a file called My Assignment with some data in directory /home/wanashekin.

Ans :- echo "Some data" > /home/wanashekin/My Assignment

v. User Wanashekin wants to view the list of files and file permissions in directory /home/wanashekin.

Ans :- ls -l /home/wanashekin

vi. User Wanashekin wants to add a new user with user's home directory called Maria in the system.

Ans :- sudo useradd -m Maria

vii. User Wanashekin wants to remove user Maria from the system.

Ans :- sudo userdel -r Maria

viii. User Wanashekin wants to add a new group called Junior with primary group ID 2100.

Ans :- sudo groupadd -g 2100 Junior

ix. User Wanashekin wants to create a text file called Event.txt.

Ans :- touch Event.txt

x. User Wanashekin wants to view the list of files in directory /tmp from home directory.

Ans :- ls -l /tmp

xi. User Wanashekin wants to move house3 and house4 to the $HOME/projects/houses/doors directory.

Ans :- mv house3 house4 $HOME/projects/houses/doors

Xii. User Wanashekin want to create nine empty files (housel to house9) in directory projects

Ans :- touch $HOME/projects/house{house1, house2, house3, house4, house5, house6, house7, house8, house9}