Q.NO. 1.

a. What is operating system?

Ans:- An operating system (OS) is a system software that acts as an intermediary between computer hardware and software applications. It manages and controls the hardware resources of a computer system, such as the central processing unit (CPU), memory, storage devices, input/output devices, and network connections. The primary functions of an operating system include managing processes, memory, file systems, and providing a user interface.

b. Discuss THREE (3) aims of operating system in providing supports for the system components.

Ans:- Three aims of an operating system in providing support for system components are:

1. **Resource Management:-** The OS aims to efficiently allocate and manage hardware resources such as CPU time, memory, and I/O devices. It ensures that multiple processes can run concurrently without interfering with each other, preventing resource conflicts and maximizing system utilization.

2. **Abstraction and Virtualization:-** Operating systems aim to abstract hardware complexities for application programs. This abstraction allows software developers to write code that is independent of specific hardware details, making it easier to develop software that runs on various types of hardware. Virtualization is an extension of this aim, where the OS creates virtual resources (e.g., virtual memory, virtual CPUs) to provide an illusion of more resources than physically available.

3. **Security and Access Control:-** Operating systems aim to provide a secure environment for executing software. They implement access control mechanisms to protect system resources from unauthorized access. This includes user authentication, file permissions, and encryption. The OS also enforces policies to ensure the integrity and confidentiality of data.

c. Describe any TWO (2) tasks that the subsystem manager must perform in operating system.

Ans:- Two tasks that the subsystem manager must perform in an operating system are:

1. Process Management:- The subsystem manager is responsible for managing processes and threads. It must create and terminate processes, allocate and deallocate resources, and schedule processes to run on the CPU efficiently. This involves context switching between processes, ensuring fair CPU time allocation, and handling process synchronization and communication.

2. Memory Management:- The subsystem manager oversees the allocation and deallocation of memory resources. It manages physical and virtual memory, including paging, swapping, and memory protection. Memory management ensures that processes have the necessary memory space to execute and prevents unauthorized access to memory areas.

d. Network manager provides a convenient way for users to share resources while controlling users' access to it. Explain with examples TWO (2) categories of network manager resources.

Ans:- Network managers in an operating system provide a convenient way for users to share resources while controlling access to them. Two categories of network manager resources are:

1. Printers:- Network managers can control access to printers connected to the network. They ensure that multiple users or devices can send print jobs to a shared printer without conflicts. Access control can be based on user permissions, print job priority, or queuing mechanisms. For example, in a corporate network, a network manager might configure access rights so that only authorized employees can print to a high-speed color printer.

2. File Shares:- Network managers enable the sharing of files and directories over a network. Users can access shared files and collaborate with others. Access control is crucial here to protect sensitive data. For instance, a network manager might set up a shared folder for a project team, granting read-only access to some members and read-write access to others, ensuring that only authorized individuals can modify the shared documents.

Q.NO.2.

a. Explain TWO (2) communication models in operating system.

Ans:- Two Communication Models in Operating Systems:

1. Message Passing Model:- In this model, processes communicate by sending and receiving messages. Each process has its own address space, and to share data, they use message-passing mechanisms provided by the operating system. This model is typically used in distributed systems and ensures better isolation between processes.

2. Shared Memory Model:- In the shared memory model, processes communicate by accessing shared regions of memory. These shared regions can be used to exchange data directly between processes. However, careful synchronization mechanisms, such as semaphores or mutexes, are required to avoid data conflicts and ensure proper communication.

b. Table 1 contains the process ID, arrival time, CPU time and priority:

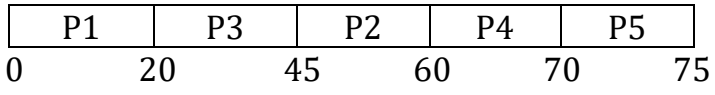
Table 1

Process ID	Arrival Time	CPU Time	Priority
P1	0	20	3
P2	1	15	2
P3	2	25	1
P4	3	10	4
P5	4	5	5

Based on the information in Table 1, draw the Gantt Chart for the following process scheduling:

i.Priority Scheduling.

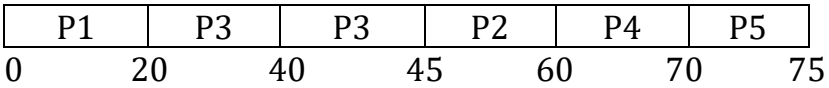
Ans:-



ii. Round Robin Scheduling with Time Quantum = 20 ms.

Ans:-

Process	AT	BT	Priority
P1	0	20	3
P2	1	15	2
P3	2	25	1 – H
P4	3	10	4
P5	4	5	5 - L



iii. Based on the information in Table 1 and Gantt Chart in Question 2b(i) and 2b(ii), write the Waiting Time for Priority Scheduling and Round Robin Scheduling.

Ans:-

Priority Scheduling

Process	AT	BT	CT	TAT	WT
P1	0	20	20	20	0
P2	1	15	60	59	44
P3	2	25	45	43	18
P4	3	10	70	67	57
P5	4	5	75	71	66

Round Robin Scheduling

Process	AT	BT	CT	TAT	WT
P1	0	20	20	20	0
P2	1	15	60	59	44
P3	2	25	45	43	18
P4	3	10	70	67	57
P5	4	5	75	71	66

c. Explain any THREE (3) conditions of deadlocks in operating system.

Ans:- Three Conditions of Deadlocks in Operating Systems:

1. Mutual Exclusion:- Deadlocks occur when processes require exclusive access to a resource, and the operating system cannot allocate it to multiple processes simultaneously. If one process holds a resource and another process requests it, but the resource is already locked, a deadlock may happen.

2. Hold and Wait:- Deadlocks can occur when processes hold some resources and request additional resources while still retaining the initial ones. If processes are not flexible in releasing resources when they cannot acquire additional ones, they can enter a state where they are waiting for resources that may never be released.

3. No Preemption:- Deadlocks may occur when resources cannot be forcibly taken away from a process. In other words, if a process is holding a resource and another process needs it, the system cannot preempt the resource from the first process. This condition can lead to a situation where processes are waiting indefinitely for resources held by others.

Q.NO. 3.

a. Describe the concept of swapping in memory.

Ans:- Swapping is a memory management technique used by operating systems to efficiently utilize physical memory (RAM) and allow processes to run. In short, swapping involves moving a portion of a process's data from RAM to a disk or secondary storage device when it's not actively needed in RAM and then bringing it back into RAM when it is needed again. This helps free up RAM for other processes, enabling efficient multitasking.

b. Explain TWO (2) types of fragmentations in memory with example of figure.

Ans:- Fragmentation refers to the inefficient utilization of memory space. There are two main types of fragmentation:

1. Internal Fragmentation:- Internal fragmentation occurs when a portion of allocated memory remains unused within a block allocated to a process. This happens because memory is allocated in fixed-size blocks or pages, and if a process's data doesn't perfectly fit the allocated block, there will be wasted space.

Example:

Imagine a memory allocation scheme where blocks are 4KB in size, and a process requires 10KB of memory. If the process is allocated two 4KB blocks, there will be 2KB of unused space in the second block, which is internal fragmentation.

Process A (10KB)	Unused (2KB)

2. External Fragmentation:- External fragmentation occurs when there is plenty of free memory space available in the system, but it is scattered in small, non-contiguous chunks, making it impossible to allocate a contiguous block of memory to a process, even though the total free memory is sufficient.

Example:

Suppose there are three free memory blocks: 5KB, 3KB, and 4KB, but a process needs 7KB of contiguous memory. Despite having a total of 12KB free memory, external fragmentation prevents the allocation of the required contiguous block.

Process A (7KB)	Unused (5KB)
Unused (3KB)	Unused (4KB)

c. Discuss TWO (2) problems in memory paging and TWO (2) solutions for each problem.

Ans:- Memory paging is a memory management technique that divides physical memory and processes into fixed-size pages. Here are two common problems associated with paging and their respective solutions:

Problem 1: Page Thrashing

Page thrashing occurs when the system spends more time swapping pages between RAM and secondary storage than executing useful instructions. This leads to severe performance degradation.

Solution

Increase RAM size: One solution is to add more physical RAM to the system, which provides more space for active pages, reducing the frequency of page swapping.

Tune page replacement algorithms: Modify the page replacement algorithms, like the Least Recently Used (LRU) or Not Recently Used (NRU) algorithm, to make better decisions about which pages to keep in RAM.

Problem 2: Page Table Size

As the number of pages in a process grows, so does the size of its page table. Large page tables consume a significant amount of memory and can become a management challenge.

Solution

Two-Level Page Tables: Implement a two-level or multi-level page table structure, which divides the page table into smaller, more manageable pieces, reducing memory overhead.

Use Hierarchical Page Tables: Employ hierarchical or inverted page tables to map virtual addresses to physical addresses more efficiently, reducing the size and complexity of individual page tables.

Q.NO. 4.

a. Explain the following types of file access in operating system:

i. Sequential File

Ans:- Sequential file access involves reading or writing data in a linear or sequential manner, where data is processed one record after another in a predetermined order. This type of file access is characterized by the fact that you can only traverse the data in a forward direction. To access a particular record, you typically have to start from the beginning of the file and read through each record until you reach the desired one. It is like reading a book from cover to cover without skipping pages.

ii. Indexed Sequential File

Ans:- Indexed sequential file access combines elements of both sequential and direct access methods. In an indexed sequential file, there is a primary index that contains pointers to the locations of data records in the file. This index allows for more efficient access to specific records compared to pure sequential access. You can locate and access records using the index, but once you've reached a specific record, you typically read it sequentially from that point.

b. Describe with an example, the character-oriented device and block-oriented device.

Ans:-

Character-Oriented Device:

A character-oriented device is a type of input or output device that processes data character by character. It treats each character, such as a letter or symbol, as a discrete unit of data. These devices are commonly used for tasks that involve processing or displaying text. An example of a character-oriented device is a computer keyboard. When you type on a keyboard, each keypress generates a character, and these characters are processed individually by the computer's software.

Example: Typing the letter "A" on a computer keyboard is an action that generates a single character ("A") and sends it to the computer for processing. The keyboard operates on a character-by-character basis, and each keypress is treated as a separate input.

Block-Oriented Device:

A block-oriented device is a type of input or output device that processes data in fixed-size blocks or chunks. It treats a group of characters or data as a unit, rather than processing individual characters separately. Block-oriented devices are often used for tasks that involve data transfer in larger chunks, such as storage devices like hard drives or data communication protocols.

Example: A hard drive is a block-oriented device. When you save a document on your computer, the data is not stored character by character; instead, it is written to the hard drive in fixed-size blocks or sectors (e.g., 4 KB or 8 KB blocks). These blocks of data are read and written together as a unit. This block-oriented approach is more efficient for handling large amounts of data, as it reduces the overhead associated with processing individual characters.

c. Discuss SIX (6) typical duties of a clock driver in in device management.

Ans:- A clock driver in device management is responsible for managing various system timers and scheduling tasks within an operating system.

Here are six typical duties of a clock driver:

1. Timekeeping:- The clock driver maintains an accurate system time by tracking the elapsed time since system startup. It ensures that the system clock remains synchronized with an external reference, such as a network time server.

2. Timer Management:- It manages hardware timers and software timers within the operating system. This includes setting timers, monitoring their expiration, and triggering associated actions or events.

3. Task Scheduling:- The clock driver plays a crucial role in scheduling processes and threads for execution. It maintains a list of processes in various states (e.g., ready, running, blocked) and uses timers to schedule their execution based on priority and time-sharing algorithms.

4. Interrupt Handling:- The clock driver handles clock interrupts generated by the system's hardware clock. These interrupts are used to periodically update the system time and to preemptively switch between running processes.

5. Timeouts and Delays:- It manages timeouts and delays in the system. For example, it can be responsible for delaying the execution of a process for a specified time or triggering actions after a certain duration has elapsed.

6. Event Synchronization:- Clock drivers are often involved in coordinating events and synchronization mechanisms within the operating system, ensuring that events occur at the correct times and that processes synchronize their actions as needed.

Q.NO. 5.

a. List any FOUR (4) system files in Ubuntu Linux operating system.

Ans:- Ubuntu Linux, like other Linux distributions, has several important system files. Here are four commonly encountered system files in Ubuntu:

1. /etc/passwd:- This file stores user account information, including usernames, user IDs (UIDs), home directories, and shell commands. It is used to authenticate users during login.

2. /etc/fstab:- The /etc/fstab file contains information about disk drives and partitions, including how they should be mounted and what options should be used. This file is essential for managing disk drives and automating the mounting process during system startup.

3. /etc/apt/sources.list:- This file defines the software repositories that Ubuntu uses to install and update packages. It specifies where the package manager (APT) should look for software packages and updates. Modifying this file can change the software sources for your system.

4. /etc/hostname:- This file stores the hostname of the system, which is the unique name used to identify it on a network. It's important for network configuration and communication with other devices on the network.

b. Assume that the current working directory is /home/ricky. Write an appropriate Linux commands for the following statements:

i. Ricky wants to create two empty directories called EC3123_Exercise and EC3123_Assignmenr.

Ans:- Create two empty directories called EC3123_Exercise and EC3123_Assignment:

mkdir EC3123_Exercise EC3123_Assignment

ii. Ricky changes to directory EC3123_Exercise and wants to create two files called Exercise1 and Exercise 2.

Ans:- Change to directory EC3123_Exercise and create two files called Exercise1 and Exercise2:

cd EC3123_Exercise

touch Exercise1 Exercise2

iii. Ricky wants to remove file Exercise 1 from the system.

Ans:- Remove file Exercise1 from the system:

rm Exercise1

iv. Ricky wants to remove directory EC3123_Exercise and any files and subdirectories it contains.

Ans:- Remove directory EC3123_Exercise and its contents:

cd /home/ricky

rm -r EC3123_Exercise

v. Ricky wants to view the list of files and file permissions in directory EC3123_Assignment.

Ans:- View the list of files and file permissions in directory EC3123_Assignment:

ls -l EC3123_Assignment

vi. Ricky wants to add a new user with user's home directory called Yusma in the system.

Ans:- Add a new user with the home directory called Yusma:

sudo useradd -m -d /home/Yusma Yusma

vii. Ricky wants to set a password for user Yusma.

Ans:- Set a password for user Yusma:

sudo passwd Yusma

viii. Ricky wants to add a new group called SeniorGroup with primary group ID 5100.

Ans:- Add a new group called SeniorGroup with primary group ID 5100:

sudo groupadd -g 5100 SeniorGroup

ix. Ricky wants to create a text file called Reports.txt and insert some texts in the file.

Ans:- Create a text file called Reports.txt and insert some text into the file:

touch Reports.txt

echo "This is some text" > Reports.txt

x. Ricky wants to view the list of files in directory /etc from home directory.

Ans:- View the list of files in the directory `/etc` from the home directory:

ls /etc

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

i. rwxrwxrwx:

ii. r-xr-xr-x:

iii. r---r---r--:

iv. rwx----:

v. rw-r----:

vi. rwxrw-rw-:

Ans:- To set these permissions for the file named "assignment," you can use the `chmod` command like this:

 # i. rwxrwxrwx: 777

chmod 777 assignment

 # ii. r-xr-xr-x: 555

chmod 555 assignment

 # iii. r---r---r--: 444

chmod 444 assignment

 # iv. rwx----: 700

chmod 700 assignment

 # v. rw-r----: 640

chmod 640 assignment

 # vi. rwxrw-rw-: 766

chmod 766 assignment