

UNIQUE NOTES
COMPUTER — CLASS 10

UNIT 01

OPERATING SYSTEMS: STRUCTURE AND SERVICES

Topics Covered

1. Operating System
2. Architecture of an OS
3. Process Management in OS
4. Memory
5. Process vs Threads
6. System Calls
7. File System Structure and Management
8. Types of OS

Table of Contents

Introduction

1. Introduction to Operating System (OS)
2. Architecture of an Operating System
3. Process Management in Operating Systems (OS)
4. Memory
5. Processes vs Threads
6. System Calls
7. File System Structure and Management
8. Types of Operating Systems

Topic Wise Short Questions (Additional)

Topic Wise Multiple Choice Questions (Additional)

Solved Exercise

Short Questions

Long Questions

Solved Activities

Unit 01 — Operating Systems: Structure and Services

Introduction

In this unit, we will explore the basic functions and structure of an operating system. An operating system is the main software that manages computer hardware and makes it easier for users to work with the computer.

Q.1 What is an Operating System (OS) and how does it work as the central controller of a computer system? [10501001]

★ **Key Points:** Operating System | Central System Controller

1. Introduction to Operating System (OS)

Operating System (OS) is a type of system software that manages hardware, runs application software, and provides a user interface, like Windows, macOS, Linux, Android, and iOS, along with their basic roles.

2. Operating System (OS) as the Central System Controller

An operating system works like a traffic controller for the computer. It decides which task should be done first, how the computer's memory is used, and which devices (like a printer or speakers) should be active at a given time. In short, we can say that the OS makes sure the computer works smoothly even when multiple tasks are running at the same time.

Q.2 Explain how an operating system acts as an interface between the user and the computer hardware, translates user actions into machine language, supports multiple users, manages user accounts. [10501002]

OR

Explain user hardware interaction, multiuser responsibilities and user account management.

★ **Key Points:** OS Role in Hardware Interaction | Multiuser | Managing User Accounts

1. Role in User-Hardware Interaction

Computer hardware cannot understand human language directly. The operating system acts as a translator between the user and the hardware system.

Example: When you click an icon or type something, the OS changes those actions into instructions that the hardware can understand. This allows people to use computers easily without learning how the hardware works.

DID YOU KNOW?

Q. Which language is understandable by computers? [10501002a]

Ans. Humans speak languages like Urdu or English, whereas computers understand only binary language, which works using ON and OFF states represented by 0 and 1.

Example: When you press the letter A on a keyboard, the operating system converts it into a special binary code:

- The letter 'A' in binary is $(01000001)_2$.
- This code is then sent to the computer screen, which displays A on the screen.
- This process happens so quickly that it feels instant to the user.

2. Responsibilities in Multi-User Environments Tasks

In places like schools, offices, or online systems, many people may use the same computer. The operating system ensures:

- The creation and management of separate user accounts.
- Keeps each user's files and information private.
- Shares the computer's resources fairly so that no one slows down the systems functionalities.
- Protect shared resources from unauthorized access.

3. Creating and Managing User Accounts

The operating system allows separate accounts for different users, each with its own desktop, files, settings, and passwords.

Example: In computer lab, student login with a username and password provided by the school. This keeps each student's work separate and private. In most modern OS like Windows, a new account can be created through the Settings or Control Panel by selecting User Accounts, choosing Add New User, and entering details like username, password, and account type (Standard, Administrative, Guest).

Q.3 Describe the architecture of an operating system, explaining kernel, shell and layered design with examples. [10501003]

★ **Key Points:** OS Architecture | Layers in OS

1. Architecture of an Operating System

The architecture of an operating system is the way how its parts are organized and how they work together. Each part has a special role, and together they make the computer work smoothly just like a school has different departments that perform specific duties but work together for the smooth working of the school.

Diagram: Relationship between User, Shell, and Kernel in an OS

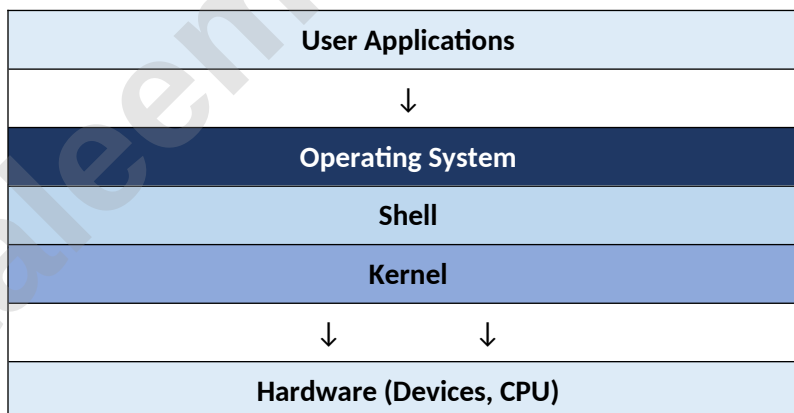


Diagram to illustrate the relationship between the user, shell, and kernel in an operating system.

2. Kernel vs Shell

a. Kernel: The kernel is the core part of the operating system. That directly controls the computer's system software and hardware such as the CPU, memory, and devices. It decides how and when different programs can use these resources.

Example: When you open a file, the kernel manages the process of reading it from the hard drive and sending it to the screen. Like the engine of a car work is kernel and accessories like steering wheel, dashboard is shell.

b. Shell: The shell is the outer part of the OS that interacts with the user. It receives commands from the user and passes them to the kernel. Shells can be:

1. Graphical shells (like Windows desktop) where you click icons and use menus.
2. Command-line shells (like the Command Prompt or Terminal) are where you type instructions.

3. OS Layers and Modular Design

In operating systems, the design is divided into layers, where each layer has a specific job:

- Lower layer: Work directly with hardware devices like the CPU, RAM and hard drive.
- Middle layer: Manages these resources and makes sure programs can use them when needed.
- Upper layer: Run applications and provide the interface that the user views on the screen.

Each layer depends on the one below it. This design makes the operating system easier to manage, repair, and improve without changing the whole system.

Example: A school system can be used as an example to understand the layered architecture of an operating system:

- The support staff (like guards and cleaners) work at the base, keeping the school ready (like the lower layers working with hardware).
- The administration manages resources, schedules, and rules (like the middle layers managing memory and storage).
- The teachers and students use these arrangements to teach and learn (like the upper layers running applications and interacting with users).

Q.4 Explain the role of system libraries and device drivers in an operating system. [10501004]

OR

Describe how system libraries provide ready-made instructions that programs can use to perform common tasks.

★ **Key Points:** System Libraries | Device Drivers

System Libraries and Device Drivers

System libraries are collections of ready-made instructions that programs can use to perform common tasks, such as opening files or showing text on the screen.

Example: When a photo editing app needs to open an image, it uses the operating system's library to read the file. This way, the app does not have to create its own method to open pictures. Device drivers are special programs that allow the operating system to communicate with hardware devices such as printers, keyboards, and graphics cards.

Q.5 Define Process Management. Explain the process life cycle in detail along with examples.

[10501005]

★ **Key Points:** Process Management | Process Life cycle | Multitasking | Concurrency

The operating system is responsible for managing all the programs that run on a computer. In this context, these running programs are called processes. Process management ensures that each process gets the resources it needs, even when multiple processes are active. Process management is one of the most important jobs of an operating system because it ensures all processes run smoothly without disturbing each other.

1. Process Life Cycle

A process goes through several stages during its life cycle:

- This happens when you start any program (like MS Word).
- The OS loads the program into memory and gives it the resources (like CPU time and memory) it needs.

b. Execution

- The process is actively running and performing tasks effectively.

c. Termination

- The process finishes its task and is closed by the user or the system.
- The OS release the resources so they can be used by other processes.

Example: Process Life cycle of opening a Web Browser on your mobile or computer like Google Chrome.

- Creation: Begins when the user clicks the browser icon. The operating system loads the program into memory and allocates the required resources.
- Execution: The browser performs tasks such as loading web pages, displaying media, and responding to user actions.
- Termination: Occurs when the browser is closed. The operating system stops the process and releases its resources for other uses.

2. Multitasking and Concurrency

Modern operating systems can manage many processes so efficiently that it appears they are all running at the same time.

a. Multitasking: The operating system allows more than one program to be open and usable by a single user at the same time. You can easily switch between them whenever you need.

Example: You can listen to music, keep a document open, and browse the internet, moving between them as needed.

b. Concurrency: More than one process is active at the same time in an OS, but the CPU processes them one by one in extremely fast cycles.

Example: Like a chef preparing three dishes, working on one for a short time, then moving to the next, and repeating, the CPU switches between processes so rapidly that the user does not notice any delay.

3. Process Scheduling Concepts

We know that many processes can be in progress during the same time period, but the CPU works on them one at a time in very fast turns. Since the CPU cannot run all processes at the same time, the operating system must decide:

- Which process should run first?
- How long each process should run?

This process is called scheduling. There are different scheduling methods used by operating systems. First Come, First Served (FCFS), which is the simplest scheduling techniques.

Q.6 Describe the First Come, First Served (FCFS) method and illustrate with examples and a numerical scenario. [10501006]

★ **Key Points:** Process Scheduling Concepts | Merits & Demerits

There are different scheduling methods used by operating systems. First Come, First Served (FCFS), which is the simplest scheduling techniques. In the FCFS method, the CPU processes tasks in the exact order they arrive. The first process to arrive is completed first, and the next process starts only after the previous one finishes.

Example: Like a queue at a shop counter, the first customer in line is served first, then the next, and so on.

Numerical Example: Let us use some simple data to see how FCFS scheduling works in practice.

Process ID	Arrival Time	Time Needed
P1	0 seconds	5 seconds
P2	1 second	3 seconds
P3	2 seconds	2 seconds

1. Step-by-Step Explanation

- P1 arrives first, so it starts running immediately.
- It requires 5 seconds to finish.
- It runs from 0s to 5s.

b. At 1 second

- P2 arrives while P1 is still running, so P2 waits.

c. At 2 seconds

- P3 arrives, but P1 is still running and P2 is already waiting.
- In FCFS, P3 will also wait until both P1 and P2 have completed.

d. At 5 seconds

- P1 finishes.
- The CPU now starts P2, which needs 3 seconds to complete.
- P2 runs from 5s to 8s.

e. At 8 seconds

- P2 finishes.
- The CPU now starts P3, which needs 2 seconds.
- P3 starts and runs from 8s to 10s.

Execution Timeline

[P1: 0–5] → [P2: 5–8] → [P3: 8–10]

P1	P2	P3
0s – 5s (finish at 5)	5s – 8s (finish at 8)	8s – 10s (finish at 10)

2. Observation

Even though P3 has the shortest duration (only 2 seconds), it must still wait until P1 and P2 have finished because they arrived earlier. This is one of the main characteristics of FCFS; shorter tasks can be delayed if they arrive after longer ones.

3. Advantages and Disadvantages of FCFS

Although FCFS is a simple and fair scheduling method, it has both advantages and disadvantages:

a. Advantages

- Easy to understand and implement
- Processes are served in the exact order they arrive.

- No process is skipped.

b. Disadvantages

- Short processes may have to wait a long time if they are queued behind longer processes.
- It can affect the overall efficiency of the system.

DID YOU KNOW?

Q. How does the CPU manage multiple processes rapidly and efficiently? [10501006a]

Ans. The CPU in your computer can switch between processes millions of times in just one second. This happens so fast that you can play music, browse the web, and download files all at once.

Q.7 What is memory? Also, explain Primary memory and Virtual Memory. [10501007]

★ **Key Points:** Memory | RAM | Virtual Memory

Memory is a fundamental component of a computer system, used to store data and instructions temporarily required for processing. It ensures that the CPU can access information quickly during program execution.

1. Primary Memory (RAM)

Primary memory, also called Random Access Memory (RAM) or Volatile Memory, is the main working area of a computer. It is a fast storage area where the computer keeps the data and instructions temporarily. Its data is erased when the computer is turned off.

Example: When you open a Word document, the computer quickly places the program and the document into RAM so the CPU can work on them right away.

2. Virtual Memory

When the RAM is full, the operating system uses part of the computer's storage drive (like Hard Disk Drives (HDD), Solid-State Drives (SSD), or Non-Volatile Memory Express (NVMe)) as virtual memory. This extra space acts like temporary RAM, allowing more programs to run at the same time. Storage drives operate at lower data transfer speeds and have higher access times as compared to RAM; that's why the use of virtual memory can result in reduced system performance. When several programs are open simultaneously, the operating system may transfer data from less active programs to virtual memory. This frees up space in RAM for the programs you are currently using, but may make those fewer active programs slower to respond.

DID YOU KNOW?

Q. How does the RAM of the first personal computer compare to modern DDR5 RAM in terms of capacity and speed? [10501007a]

Ans. The first personal computer, the IBM 5150 (1981), was equipped with only 16 KB of RAM. In contrast, modern DDR5 RAM can transfer data at speeds exceeding 50 GB per second, enabling the copying of a full high-definition movie in less than one second.

Q.8 What are the differences between Processes and Threads? [10501008]

★ **Key Points:** Processes | Threads

Processes	Threads
An independent program that is currently being executed by the computer.	The smallest unit of execution within a process.
It has its own memory space, CPU time, and other	Shares the same memory and resources within

Processes	Threads
resources (files, network).	the process.
Processes are isolated from one another to ensure stability and security.	Threads operate independently but are not isolated; they share resources.
A problem in one process does not affect another.	Allows multiple tasks within the same process to run simultaneously without waiting for each task to finish.
A web browser as a whole is one process.	Loading a webpage, playing audio/video, and downloading files are threads within the browser process.

Q.9 What is Multithreading? Also, write the benefits of multithreading. [10501009]

★ **Key Points:** Multithreading | Merits of Multithreading

Multithreading is an operating system technique that allows a single process to perform multiple tasks at the same time by dividing its work into smaller units called threads. Each thread runs independently but shares the same memory and resources of the process, enabling faster execution, better responsiveness, and efficient use of system resources.

Benefits of Multithreading

Multithreading offers several advantages, and some of the most important are discussed below:

3. Enhanced Performance: Tasks can be divided into multiple threads and executed in parallel, allowing complex operations to complete more quickly and improving overall system efficiency.
4. Improved Responsiveness: Applications remain responsive even when one thread is busy with a specific task.

Example: A word processor enables continuous typing while another thread check spelling in the background.

5. Support for Concurrent (parallel) Operations: Multiple tasks can progress during the same period, which is particularly important in applications such as games, video editing, and real-time communication tools.
6. Efficient Use of Resources: Threads share the same memory space and resources of their parent process, requiring fewer system resources compared to creating separate processes.

Q.10 Describe system calls, their purpose and its types. [10501010]

★ **Key Points:** System Call | open | read | write | fork

A system call is a request made by a program to the operating system to perform a specific task that the program cannot do directly.

1. Purpose

They act as a bridge between user programs and the kernel, allowing applications to access hardware and core OS functions safely. Without system calls, programs directly control hardware can be unsafe and complex.

Example: When you save a file in a text editor, the program uses a system call to tell the operating system to write the data to the storage drive (like a Hard drive).

2. Types of System Calls

Type	Purpose / Example
open	Opens a file for reading or writing. Example: Opening a music file to play.
read	Retrieves data from a file or input device. Example: Reading text from a document.
write	Sends data to a file or output device. Example: Saving an image to the computer.
fork	Creates a new process by duplicating an existing one. Example: Opening a new browser tab, where the OS may use fork to create another process.

DID YOU KNOW?

Q. How many system calls do Linux, macOS, and Windows have, and what do these system calls handle? [10501010a]

Ans. Linux and macOS have a few hundred system calls, while Windows uses nearly 2,000. These calls handle everything from opening files to running apps and showing graphics in the background while you work.

Q.11 Explain the file system structure and management. [10501011]

★ **Key Points:** File Structure | Files | Folders | Metadata | File Systems

An operating system not only runs programs but also stores and organizes the user's data. This is achieved through the file system, which provides a structured way for users and applications to store, locate, and manage information on storage devices. A well-designed file system ensures that data remains organized, secure, and easily retrievable, even when there are thousands of files containing complex data.

1. Files

A file is a collection of related data stored on a computer. It may contain text, images, audio, video, or program instructions.

2. Folders

A folder (also called a directory) is a container used to organize files and other folders in a logical structure, making it easier to locate and manage information.

3. Metadata

Metadata refers to details about a file or folder, such as its name, type, size, date of creation, and date of last modification. This information helps both the operating system and the user identify and manage files/folders without opening them.

4. File Systems

A file system is the way an operating system stores and organizes files on a storage device, such as a hard drive, SSD, or USB. It decides where each file will be kept and saves its location so it can be accessed later. The choice of file system affects performance, storage capacity, and security features. Some widely used file systems include:

File System	Used By
FAT32 (File Allocation Table 32)	Used in USB flash drives
NTFS (New Technology File System)	Used by Windows OS
APFS (Apple File System) / HFS+ (Hierarchical File System Plus)	Used by macOS
EXT4 (Fourth Extended File System)	Commonly used by Linux OS

DID YOU KNOW?

Q. How many system calls do Linux, macOS, and Windows have, and what do these system calls handle? [10501010a]

Ans. Linux and macOS have a few hundred system calls, while Windows uses nearly 2,000 system calls that handle everything from opening files to running apps and showing graphics in the background while you work.

Q.12 Write a comprehensive note on types of Operating Systems. [10501012]

★ **Key Points:** Operating System | Mobile Operating System

Operating systems are designed according to the needs of the device and the work it performs. Each type serves a different purpose and is optimized for specific tasks.

1. Real-Time Operating System (RTOS)

A Real-Time Operating System is designed to process data and respond within a strict time limit, known as a deadline. It is used where even a tiny delay can cause system failure or serious consequences.

- Key Feature: Processes tasks immediately as they arrive, without long waiting times.
- Used in: Critical systems such as air traffic control, heart-monitoring devices, or industrial robots.

2. Embedded Operating System (EOS)

An Embedded OS is a small and highly efficient operating system built into a specific device to control only the functions it needs. It is not meant for general purpose computing but is optimized for one task or a small set of tasks.

- Key Feature: Uses very little memory and power, and is often stored permanently inside the device.
- Used in: Home appliances (microwaves, washing machines), printers, smart TVs, and ATMs (Automated Teller Machine).

3. Network Operating System (NOS)

A Network OS manages and supports multiple computers connected through a network. It enables the sharing of resources like files, printers, and internet connections among users.

- Key Feature: Focuses on communication and coordination between computers.
- Used in: Offices, schools, and data centers.

Example: In a school computer lab, students use different computers, but all can save files to the same server and print from the same printer.

4. Mobile Operating System

A Mobile OS is designed for smartphones, tablets, and other handheld devices. It is optimized for touch-screen use, battery saving, and mobile apps.

- Key Feature: Supports wireless connectivity, cameras, sensors, and app stores.
- Used in: Smartphones, tablets, smartwatches.

DID YOU KNOW?

Q. How do early mobile operating systems like Symbian compare to modern platforms like Android and iOS? [10501012a]

Ans. The very first mobile operating systems, like Symbian (used in early Nokia phones), could only run a few small apps. Today's mobile OS platforms like Android and iOS can handle millions of apps, 3D games, and even professional video editing tools.

Topic Wise Short Questions (Additional)

Introduction to Operating System (OS)

Q.1 What is an Operating System (OS)?

Ans. An Operating System (OS) is a type of system software that manages hardware, runs application software, and provides a user interface, like MS Windows, macOS, Linux, Android, and iOS.

Q.2 How does the OS work as the central system controller?

Ans. An operating system works like a traffic controller for the computer. It decides which task should be done first, how the computer's memory is used, and which devices (like a printer or speakers) should be active at a given time. It makes sure the computer works smoothly even when multiple tasks are running at the same time.

Q.3 How does the OS enable interaction between users, applications and hardware?

Ans. The operating system acts as a translator between the user, applications and the hardware system. When you click an icon or type something, the OS changes those actions into instructions that the hardware can understand.

Q.4 Give an example of how the OS translates user actions into hardware instructions?

Ans. When you press the letter A on a keyboard, the operating system converts it into a special binary code: the letter 'A' in binary is $(01000001)_2$. This code is sent to the computer screen, which displays A.

Q.5 Do humans and computers understand the same languages?

Ans. No, humans and computers do not understand the same language. Humans speak languages like Urdu or English, whereas computers understand only binary language, which works using ON and OFF states represented by 1 and 0 respectively.

Q.6 How to convert letter 'A' into binary code?

Ans. 'A' is a character which has ASCII (American Standard Code for Information Interchange) 65. Convert 65 into binary by successive division by 2, recording remainders, then reading remainders from bottom to top: $A = (65)_{10} = (01000001)_2$.

Q.7 How does the OS manage multiple users on the same computer?

Ans. The OS ensures creation and management of separate user accounts, keeps each user's files and information private, shares the computer's resources equally so no one slows down the system, and protects shared resources from unauthorized access.

Q.8 How does the OS create and manage user accounts? Also, give its example.

Ans. The OS allows separate accounts for different users, each with its own desktop, files, settings, and passwords. Example: In a computer lab, students log in with a username and password provided by the school. A new account can be created through Settings or Control Panel by selecting User Accounts, choosing Add New User, and entering details like username, password, and account type (Standard, Administrative, Guest).

Architecture of an Operating System

Q.9 What is the architecture of an operating**Q.14 What do lower layers of the OS do?**

system?

Ans. The architecture of an operating system is the way how its parts are organized and how they work together. Each part has a special role, and together they make the computer work smoothly, just like a school has different departments that perform specific duties but work together for the smooth working of the school.

Q.10 What is the kernel?

Ans. The kernel is the core part of the operating system, that directly controls the computer's system software and hardware such as the CPU, memory, and devices. It decides how and when different programs can use these resources.

Q.11 Give an example of how the kernel works.

Ans. When you open a file, the kernel manages the process of reading it from the hard drive and sending it to the screen. Example: Like the engine of a car work is kernel and accessories like steering wheel, dashboard is shell.

Q.12 What is the shell?

Ans. The shell is the outer part of the OS that interacts with the user. It receives commands from the user and passes them to the kernel.

Q.13 Name two types of shells.

Ans. Graphical shells (like Windows desktop) and Command-line shells (like the Command Prompt or Terminal).

Ans. The lower layer works directly with hardware devices like the CPU, RAM, secondary storage (i.e. hard disk). They manage and control hardware resources.

Q.15 What is the function of the middle layer in OS?

Ans. The middle layer of the OS manages system resources (such as CPU, memory, and devices) and ensures that programs can access and use them efficiently when needed.

Q.16 What does the upper layer do?

Ans. Upper-layer run applications and provide the interface that the user views on the screen.

Q.17 What are system libraries?

Ans. System libraries are collections of ready-made instructions that programs can use to perform common tasks, such as opening files or showing text on the screen.

Q.18 What are device drivers?

Ans. Device drivers are special programs that allow the operating system to communicate with hardware devices such as printers and graphics cards.

Process Management in Operating Systems (OS)

Q.19 What is the role of the operating system in process management?

Ans. The operating system is responsible for managing all the programs that run on a computer. These running programs are called processes. Process management ensures that each process gets the resources it needs, even when multiple processes are active, without disturbing each other.

Q.20 What are the stages of a process life cycle?

Q.25 What is concurrency in an operating system?

Ans. Concurrency in an operating system means that more than one process is active at the same time. The CPU switches between processes very quickly.

Q.26 What is process scheduling?

Ans. A process goes through several stages during its life cycle: Creation, Execution, and Termination.

Q.21 What happens during the creation stage of a process?

Ans. This happens when you start any program (like MS Word). The OS loads the program into memory and gives it the resources (like CPU time and memory) it needs.

Q.22 What happens during the execution stage of a process?

Ans. During execution phase the process is actively running and performing tasks effectively.

Q.23 What happens during the termination stage of a process?

Ans. The process finishes its task and is closed by the user or the system. The OS releases the resources so they can be used by other processes.

Q.24 What is multitasking in an operating system?

Ans. The operating system allows more than one program to be open and usable by a user at the same time. You can easily switch between them whenever you need.

Memory

Q.31 What is memory in a computer system?

Ans. Memory is a fundamental component of a computer system, used to store data and instructions required for processing. It ensures that the CPU can access information quickly during program execution.

Ans. Since the CPU cannot run all processes at the same time, the operating system must decide which process should run first and how long each process should run. This process is called scheduling.

Q.27 Which is the First Come, First Served (FCFS) scheduling method?

Ans. In the FCFS method, the CPU processes tasks in the exact order they arrive. The first process to arrive is completed first, and the next process starts only after the previous one finishes.

Q.28 Give an example of FCFS scheduling with three processes.

Ans. P1 arrives first and runs from 0s to 5s, P2 runs from 5s to 8s, and P3 runs from 8s to 10s. Even though P3 has the shortest duration, it must wait until P1 and P2 have finished.

Q.29 What are the advantages of FCFS?

Ans. Easy to understand and implement, processes are served in the exact order they arrive, and no process is skipped.

Q.30 What are the disadvantages of FCFS?

Ans. Short processes may have to wait a long time if queued behind longer processes (known as the 'convoy effect'), and this can affect the overall efficiency of the system.

Q.35 What is the purpose of virtual memory?

Ans. The primary purpose of virtual memory is to extend a computer's usable memory from secondary storage like hard disk drive.

Q.32 What is primary or Volatile memory?

Ans. Primary memory, also called Random Access Memory (RAM) or Volatile Memory, is the main working area of a computer. It is a fast storage area where the computer keeps the data and instructions temporarily. Its data is erased when the computer is turned off.

Q.33 How does RAM work? Explain with an example.

Ans. When you open a Word document, the computer quickly places the program and the document into RAM so the CPU can work on them right away.

Q.34 What is virtual memory?

Ans. When the RAM is full, the operating system uses part of the computer's storage drive (like Hard Drives, Solid-State Drives (SSD), or Non-Volatile Memory Express (NVMe)) as virtual memory.

Q.36 How does virtual memory affect system performance?

Ans. Storage drives operate at lower data transfer speeds and have higher access times as compared to RAM; that's why the use of virtual memory can result in reduced system performance.

Q.37 What happens when several programs are open simultaneously?

Ans. The operating system may transfer data from less active programs to virtual memory. This frees up space in RAM for the programs you are currently using, but may make those fewer active programs slower to respond.

Processes vs Threads

Q.38 What is a process?

Ans. A process is an independent program that is currently being executed by the computer. It has its own memory space, CPU time, and other resources to run.

Q.39 What is a thread?

Ans. A thread is the smallest unit of execution within a process. Multiple threads can exist inside a single process, performing different tasks while sharing the same memory and resources.

Q.40 How do processes and threads differ in resources?

Ans. Processes have their own memory space, CPU time, and other resources such as files or network connections. Threads share the same memory and resources within the process.

Q.41 How do processes and threads differ in isolation?

Ans. Processes are isolated from one another to ensure stability and security, so a problem in one process does not affect another. Threads operate

Q.43 What is the purpose or benefit of a thread?

Ans. Threads allow multiple tasks within the same process to run simultaneously. This ensures that tasks can progress without waiting for each other to finish, improving efficiency.

Q.44 Give an example of a process.

Ans. A web browser as a whole is one process. The entire application runs independently with its own memory and resources allocated by the operating system.

Q.45 Give an example of threads within a process.

Ans. Loading a webpage, playing audio or video, and downloading files.

Q.46 What is multithreading?

Ans. Multithreading is an operating system technique that allows a single process to perform multiple tasks at the same time by dividing its

independently but are not isolated, sharing resources within the same process.

Q.42 What is the purpose or benefit of a process?

Ans. The purpose of a process is to ensure stability in the system. A problem in one process generally does not affect the execution of other processes.

work into smaller units.

Q.47 What are the benefits of multithreading?

Ans. Multithreading enhances performance by allowing tasks to run in parallel, improving system efficiency. It also improves responsiveness, supports concurrent operations, and uses system resources more efficiently since threads share the same memory and resources of the parent process.

System Calls

Q.48 What is a system call?

Ans. A system call is a request made by a program to the operating system to perform a specific task that the program cannot do directly. They act as a bridge between user programs and the kernel, allowing applications to access hardware and core OS functions safely.

Q.49 Why are system calls important?

Ans. Without system calls, programs that directly control hardware can be unsafe and complex. They allow safe interaction between programs and the operating system.

Q.50 Give an example of a system call in action.

Ans. When you save a file in a text editor, the program uses a system call to tell the operating system to write the data to the storage drive, like a Hard drive.

Q.51 How the 'open' system call work?

Ans. The open system call opens a file for reading or writing. Example: Opening a music file to play.

Q.52 How the 'read' system call work?

Ans. The read system call retrieves data from a file or input device. Example: Reading text from a document.

Q.53 How the 'write' system call work?

Ans. The write system call sends data to a file or output device. Example: Saving an image to the computer.

Q.54 How the 'fork' system call work?

Ans. The fork system call creates a new process by duplicating an existing one. Example: Opening a new browser tab, where the OS may use fork to create another process.

Q.55 How many system calls do Linux and macOS have?

Ans. Linux and macOS have a few hundred system calls.

Q.56 How many system calls does Windows have?

Ans. Windows uses nearly 2,000 system calls that handle everything from opening files to running apps and showing graphics in the background while you work.

Q.57 What is the main role of system calls in operating systems?

Ans. System calls allow programs to access hardware and core OS functions safely, acting as a bridge between user programs and the kernel.

File System Structure and Management

Q.58 What is the role of an operating system in data storage?

Ans. An operating system not only runs programs but also stores and organizes the user's data. This is achieved through the file system, which provides a structured way for users and applications to store, locate, and manage information on storage devices.

Q.59 What is a file?

Ans. A file is a collection of related data stored on a computer. It may contain text, images, audio, video, or program instructions.

Q.60 What is a folder?

Ans. A folder (also called a directory) is a container used to organize files and other folders in a logical structure, making it easier to locate and manage information.

Q.61 What is metadata?

Ans. Metadata refers to details about a file or folder, such as its name, type, size, date of creation, and date of last modification. This information helps both the operating system and the user identify and manage files/folders without opening them.

Q.62 What is a file system?

Ans. A file system is the way an operating system stores and organizes files on a storage device, such as a hard drive, SSD, or USB. It decides where each file will be kept and saves its location so it can be accessed later.

Q.63 Name some widely used file systems.

Ans. FAT32 (File Allocation Table 32): Used in USB flash drives. NTFS (New Technology File System): Used by Windows OS. APFS (Apple File System)/HFS+ (Hierarchical File System Plus): Used by macOS. EXT4 (Fourth Extended File System): Commonly used by Linux OS.

Q.64 How does the choice of file system affect a computer?

Ans. The choice of file system affects performance, storage capacity, and security features.

Q.65 Give an example of how a file system works.

Ans. If you save a photo on your computer, the file system makes sure it is stored in the right place and knows exactly where to find it when you open it again, just like a librarian placing a book in the right section and knowing where to locate it later.

Q.66 Why is a file system important for users and applications?

Ans. A well-designed file system ensures that data remains organized, secure, and easily retrievable, even when there are thousands of files containing complex data.

Q.67 What types of data can a file contain?

Ans. A file may contain text, images, audio, video, or program instructions.

Types of Operating Systems

Q.68 Why are operating systems designed differently for each device?

Ans. Operating systems are designed according to the needs of the device and the work it performs. Each type serves a different purpose and is optimized for specific tasks.

Q.75 What is a Network Operating System (NOS)?

Ans. A Network OS manages and supports multiple computers connected through a network. It enables the sharing of resources like files, printers, and internet connections among

users.

Q.69 What is a Real-Time Operating System (RTOS)?

Ans. A Real-Time Operating System is designed to process data and respond within a strict time limit, known as a deadline. It is used where even a tiny delay can cause system failure or serious consequences.

Q.70 What are the key features of RTOS?

Ans. Processes tasks immediately as they arrive, without long waiting times.

Q.71 Where is RTOS commonly used?

Ans. RTOS is used where critical systems such as air traffic control, heart-monitoring devices, or industrial robots' tasks must be completed within strict time limits.

Q.72 What is an Embedded Operating System (EOS)?

Ans. An Embedded OS is a small and highly efficient operating system built into a specific device to control only the functions it needs. It is not meant for general-purpose computing but is optimized for one task or a small set of tasks.

Q.73 What are the key feature of EOS?

Ans. The key features of EOS are; uses very little memory, power, and is often stored permanently inside the device.

Q.74 Where is EOS commonly used?

Ans. EOS are commonly used in home appliances (microwaves, washing machines), printers, smart TVs, and ATMs (Automated Teller Machine).

Q.76 What is the key feature of NOS?

Ans. The key feature of NOS is that it focuses on communication and coordination between computers in a network.

Q.77 Write the example of NOS usage.

Ans. In a school computer lab, students use different computers, but all can save files to the same server and print from the same printer.

Q.78 What is a Mobile Operating System?

Ans. A Mobile OS is designed for smartphones, tablets, and other handheld devices. It is optimized for touch-screen use, battery saving, and mobile apps.

Q.79 What are the key features of Mobile OS?

Ans. It supports wireless connectivity, cameras, sensors, and app stores, allowing mobile devices to run apps and communicate effectively.

Q.80 Where is Mobile OS commonly used?

Ans. Mobile phone OS is commonly used in: Smartphones, Tablets, Smartwatches.

Topic Wise Multiple Choice Questions (Additional)

Introduction to Operating System (OS)

1. What is an Operating System (OS)?
 - (a) Application
 - (b) System software
 - (c) Hardware
 - (d) Programming language
2. Which of the following is an example of an OS?
 - (a) Word
 - (b) Chrome
 - (c) Windows
 - (d) Photoshop
3. How does the OS control the system?
 - (a) Manages tasks and devices
 - (b) Stores files
 - (c) Runs internet
 - (d) Opens apps only
4. Which OS acts as a translator between?
 - (a) CPU and RAM
 - (b) Keyboard and mouse
 - (c) Software and files
 - (d) User and hardware
5. When you press 'A', OS converts it to?
 - (a) Unicode
 - (b) Binary
 - (c) Text
 - (d) Symbol
6. Do humans and computers understand the same language?
 - (a) No
 - (b) Yes
 - (c) Sometimes
 - (d) Always
7. OS in multi-user systems ensures.
 - (a) Single user
 - (b) Deletes files
 - (c) Private files and fair resources
 - (d) Runs games
8. What does the OS provide for each user?
 - (a) Shared desktop
 - (b) One password
 - (c) Separate accounts
 - (d) Guest access only
9. How to add a new user in Windows?
 - (a) Settings → User Accounts
 - (b) Command prompt
 - (c) Install program
 - (d) Restart PC
10. Which of the following is NOT an OS responsibility?
 - (a) Privacy
 - (b) Fair resource sharing
 - (c) Protect resources
 - (d) Create drivers automatically

Architecture of an Operating System

11. How many main types of user-facing shells are available?
 - (a) 1
 - (b) 2
16. If a printer doesn't work, which component is checked first?
 - (a) Kernel
 - (b) Driver

- (c) 3
- (d) 4

- (c) Shell
- (d) Library

12. Which OS component decides CPU usage among programs?

- (a) Shell
- (b) Kernel
- (c) Library
- (d) App

17. Why is the kernel like a car engine?

- (a) Controls core functions
- (b) Interacts with user
- (c) Shows output
- (d) Runs apps

13. Why is layered OS design useful?

- (a) Faster execution
- (b) Easier management
- (c) Removes hardware
- (d) Eliminates shell

18. Which scenario shows the shell's role?

- (a) Background processes
- (b) Managing CPU
- (c) Printer communication
- (d) Clicking desktop icons

14. CPU stands for?

- (a) Central Processing Unit
- (b) Control Processing Unit
- (c) Combine Processing Unit
- (d) None of these

19. Why are system libraries important?

- (a) Speed up hardware
- (b) Manage accounts
- (c) Provide ready-made instructions
- (d) Replace kernel

15. What does OS stand for?

- (a) Operational System
- (b) Opening System
- (c) Operating System
- (d) None of these

20. If the middle OS layer fails, what happens?

- (a) Hardware stops
- (b) Resource management affected
- (c) Desktop disappears
- (d) Drivers not needed

Process Management in Operating System (OS)

21. What are running programs called in an operating system?

- (a) Threads
- (b) Processes
- (c) Files
- (d) Tasks

26. What is multitasking?

- (a) OS allows more than one program to be open and usable by a user at the same time
- (b) CPU processes one task only
- (c) Process termination
- (d) Saving files

22. What ensures that each process gets the resources it needs?

- (a) Process management
- (b) File system
- (c) Memory management
- (d) Multithreading

27. What is concurrency in an OS?

- (a) CPU executes one process at a time
- (b) Only memory management
- (c) Only multitasking
- (d) More than one process is active, CPU switches between them rapidly

23. Which is the first stage of a process life cycle?

- (a) Execution
- (b) Termination
- (c) Creation
- (d) Scheduling

24. What happens during the execution stage?

- (a) Process is actively running and performing tasks
- (b) Process is loaded into memory
- (c) Process is closed
- (d) Resources are freed

25. What does the termination stage involve?

- (a) Loading program into memory
- (b) Process finishes its task and OS frees resources
- (c) CPU scheduling
- (d) Multitasking

28. What is process scheduling?

- (a) Deciding which process runs first and how long each process runs
- (b) Closing programs
- (c) Creating threads
- (d) Managing files

29. How does the FCFS scheduling method work?

- (a) CPU runs shortest process first
- (b) CPU processes tasks in the exact order they arrive
- (c) CPU skips long processes
- (d) CPU runs processes randomly

30. Which of the following is a disadvantage of FCFS?

- (a) Easy to implement
- (b) Processes are skipped
- (c) Short processes may wait behind longer ones (convoy effect)
- (d) Complex scheduling

Memory

31. What is memory in a computer system?

- (a) CPU only
- (b) Stores data and instructions
- (c) Storage drive only
- (d) Application software

36. RAM stands for:

- (a) Read access
- (b) Memory
- (c) Random
- (d) Recording

32. What does memory ensure for the CPU?

- (a) Quick access to information during program execution
- (b) Faster printing
- (c) Permanent data storage
- (d) Internet access

37. What is virtual memory?

- (a) Part of CPU
- (b) Part of storage drive used as temporary RAM when RAM is full
- (c) RAM that never erases
- (d) Only SSD memory

33. Which of the following is another name for primary memory?

- (a) ROM
- (b) Virtual memory

38. Why is virtual memory slower than RAM?

- (a) Lower storage capacity
- (b) It is external
- (c) Lower data transfer speeds and higher

- (c) RAM
- (d) SSD

- access times
- (d) It runs in background

34. Which of the following type of storage is RAM?

- (a) Slow storage
- (b) Fast storage area where data and instructions are kept temporarily
- (c) Permanent storage
- (d) External storage

39. What does virtual memory allow?

- (a) Fewer programs to run
- (b) More programs to run at the same time
- (c) CPU to stop working
- (d) Faster internet

35. What happens to data in RAM when the computer is turned off?

- (a) It is erased
- (b) It is saved permanently
- (c) It transfers to SSD automatically
- (d) It slows down

40. What happens when several programs are open at once?

- (a) CPU stops working
- (b) RAM is erased
- (c) OS transfers data from less active programs to virtual memory
- (d) Programs close automatically

Processes vs Threads

41. What is a process in an operating system?

- (a) Task
- (b) Independent program
- (c) CPU part
- (d) Temporary memory

46. Which of the following is an example of a process.

- (a) Web browser app
- (b) Loading page
- (c) Playing audio
- (d) Download file

42. What is a thread?

- (a) Smallest unit of execution
- (b) Independent program
- (c) CPU part
- (d) Storage

47. Which of the following are the examples of threads in a process.

- (a) Browser app
- (b) CPU
- (c) Loading page, playing media, downloading files
- (d) Hard drive

43. How do threads differ from processes in resources?

- (a) Share memory and resources
- (b) Separate memory
- (c) Cannot run together
- (d) Fully isolated

48. What is multithreading?

- (a) Single process with multiple threads
- (b) Multiple processes
- (c) Isolated execution
- (d) External memory

44. Why are processes isolated?

- (a) Faster
- (b) Stability and security
- (c) Share memory
- (d) Slow system

45. What are the main benefits of threads?

- (a) Isolate programs
- (b) One task only
- (c) Own memory
- (d) Run multiple tasks simultaneously

49. Which of the following are the key benefits of multithreading?

- (a) More memory
- (b) Prevents concurrency
- (c) Slower single task
- (d) Parallel tasks, better performance

50. How do threads use resources?

- (a) Share memory of parent process
- (b) Separate memory
- (c) No CPU use
- (d) Run after process

System Calls

51. What is a system call?

- (a) Hardware component
- (b) Request by a program to OS
- (c) File in storage
- (d) CPU instruction

56. What does the write system call do?

- (a) Retrieves data
- (b) Sends data to a file or output device
- (c) Opens the OS
- (d) Stops CPU

52. Why are system calls important?

- (a) Allow safe access to hardware and OS functions
- (b) Make programs slower
- (c) Replace RAM
- (d) Control only files

57. What does the fork system call do?

- (a) Creates a new process by duplicating an existing one
- (b) Deletes files
- (c) Opens a document
- (d) Reads memory

53. Which of the following example of a system call in action?

- (a) Playing a video
- (b) Turning off monitor
- (c) Opening CPU settings
- (d) Saving a file in a text editor

58. How many system calls do Linux and macOS have?

- (a) About 2,000
- (b) Few hundred
- (c) 10
- (d) 50

54. What does the open system call do?

- (a) Opens a file for reading or writing
- (b) Deletes a file
- (c) Closes the CPU
- (d) Runs applications

59. How many system calls does Windows have?

- (a) Few dozen
- (b) Few hundred
- (c) Nearly 2,000
- (d) 100

55. What does the read system call do?

- (a) Saves data

60. What is the main role of system calls?

- (a) To directly control hardware

- (b) Starts a process
- (c) Retrieves data from a file or input device
- (d) Deletes memory

- (b) Act as a bridge between programs and OS for safe access to hardware
- (c) Store files permanently
- (d) Replace RAM

File System Structure and Management

61. What is the role of an operating system in data storage?

- (a) Only runs programs
- (b) Stores and organizes users' data
- (c) Only manages CPU
- (d) Only manages memory

66. Which file system is used in USB flash drives?

- (a) NTFS
- (b) EXT4
- (c) FAT32
- (d) APFS

62. What is a file?

- (a) Collection of related data stored on a computer
- (b) A container for folders
- (c) Metadata of folders
- (d) A storage device

67. Which file system is commonly used by Linux OS?

- (a) EXT4
- (b) FAT32
- (c) NTFS
- (d) APFS

63. What is a folder?

- (a) A storage device
- (b) Container to organize files/folders
- (c) Type of file
- (d) CPU memory

68. How does the choice of file system affect a computer?

- (a) Only graphics
- (b) CPU speed
- (c) Performance, storage capacity, and security
- (d) Keyboard input

64. What is metadata?

- (a) Data inside a file
- (b) Storage location
- (c) Details about a file or folder
- (d) File system type

69. Give an example of how a file system works.

- (a) Runs programs in memory
- (b) Ensures a saved photo is stored in the right place and can be retrieved
- (c) Controls CPU
- (d) Only manages folders

65. What is a file system?

- (a) Program that runs applications
- (b) Way an OS stores and organizes files on a storage device
- (c) Type of folder
- (d) A file type

70. Which types of data can a file contain?

- (a) Text, images, audio, video, or program instructions
- (b) Only text
- (c) Only images
- (d) Only programs

Types of Operating Systems

71. Why are operating systems designed differently for each device?

- (a) To reduce cost
- (b) Optimized for specific tasks
- (c) Only for computers
- (d) To use more memory

72. What is a Real-Time Operating System (RTOS)?

- (a) OS for smartphones
- (b) Network OS
- (c) Processes data with strict time limits
- (d) Embedded OS

73. Which of the following are the key features of RTOS?

- (a) Processes tasks immediately
- (b) Uses little memory
- (c) Supports wireless connectivity
- (d) Manages networks

74. Where is RTOS commonly used?

- (a) Smartphones
- (b) Home appliances
- (c) Air traffic control, heart-monitoring devices
- (d) Office computers

75. What is an Embedded Operating System (EOS)?

- (a) OS for networks
- (b) Real-time OS
- (c) OS for mobile apps
- (d) Small OS built into a specific device

76. Which of the following are the key features of EOS?

- (a) Touch-screen optimized
- (b) Uses very little memory and power
- (c) Focuses on communication
- (d) Handles multiple apps

77. Where is EOS commonly used?

- (a) Smartphones
- (b) School labs
- (c) Printers, microwaves, ATMs
- (d) Servers

78. What is a Network Operating System (NOS)?

- (a) OS for handheld devices
- (b) OS for one device
- (c) Manages multiple computers in a network
- (d) Real-time OS

79. Which of the following are the key features of NOS?

- (a) Communication and coordination between computers
- (b) Battery saving
- (c) Processes tasks immediately
- (d) Touch-screen support

80. What is a Mobile Operating System used for?

- (a) Servers
- (b) Data centers
- (c) Air traffic control
- (d) Smartphones, tablets, and handheld devices

Answers

1	b	2	c	3	a	4	d	5	b	6	a	7	c	8	c
9	a	10	d	11	b	12	b	13	b	14	a	15	c	16	b

17	a	18	d	19	c	20	b	21	b	22	a	23	c	24	a
25	b	26	a	27	d	28	a	29	b	30	c	31	b	32	a
33	c	34	b	35	a	36	c	37	b	38	c	39	b	40	c
41	b	42	a	43	a	44	b	45	d	46	a	47	c	48	a
49	d	50	a	51	b	52	a	53	d	54	a	55	c	56	b
57	a	58	b	59	c	60	b	61	b	62	a	63	b	64	c
65	b	66	c	67	a	68	c	69	b	70	a	71	b	72	c
73	a	74	c	75	d	76	b	77	c	78	c	79	a	80	b

Solved Exercise

Multiple Choice Questions

1. Which is NOT an example of an operating system?
 - (a) Linux
 - (b) Windows
 - (c) Photoshop
 - (d) macOS
2. In multi-user environments, the OS ensures:
 - (a) Equal access and data privacy
 - (b) All users share the same files
 - (c) Hardware is never used
 - (d) Internet speed is increased
3. In Windows, user accounts can be created via:
 - (a) Task Manager
 - (b) Control Panel or Settings
 - (c) Disk Management
 - (d) BIOS settings
4. The core part of the OS that interacts directly with hardware is the:
 - (a) Shell
 - (b) Kernel
 - (c) Device Driver
 - (d) System Library
5. A graphical shell allows the user to:
 - (a) Type commands only
 - (b) Click icons and use menus
 - (c) Interact only via code
 - (d) Access only the BIOS
6. In the FCFS scheduling method, processes are served:
 - (a) Randomly
 - (b) Shortest job first
 - (c) In the order they arrive
 - (d) By priority only
7. Threads in the same process:
 - (a) Have separate memory spaces
 - (b) Share the same memory and resources
 - (c) Run on different OS
 - (d) Cannot run simultaneously
8. Which system call is used to create a new process?
 - (a) open
 - (b) read
 - (c) write
 - (d) fork

Answers

1	c	2	a	3	b	4	b	5	b	6	c	7	b	8	d
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Short Questions

Q.1 Why is the operating system referred to as the "central controller"?

Ans. The operating system controls all the activities of the computer system. It manages hardware like CPU, memory, and input/output devices. That is why it is called the central

Q.6 Explain one advantage and one disadvantage of FCFS scheduling.

Ans. An advantage of FCFS is that it is simple and easy to implement. A disadvantage is that long processes may cause other processes to wait. This

controller of the system.

Q.2 How does the OS act as a translator between the user and hardware?

Ans. Users give commands through applications or interfaces. The operating system converts these commands into machine-level instructions. This helps hardware understand and perform user tasks.

Q.3 Define one way to create a new user account in Windows.

Ans. A new user account can be created through the Control Panel or Settings. Go to Settings → Accounts → Family & other users → Add account. This allows multiple users to use the same computer separately.

Q.4 How is kernel different from shell.

Ans. The kernel is the core part of the operating system. It directly communicates with the hardware. The shell is the interface that allows users to interact with the OS.

Q.5 Give one example of a system library and its purpose.

Ans. An example of a system library is the C standard library. It provides useful functions like input, output, and memory handling. Libraries help programmers write programs more easily.

can increase the overall waiting time.

Q.7 Why is virtual memory slower than RAM?

Ans. Virtual memory uses storage space from the hard disk. Hard disk access is much slower than RAM access. Therefore, virtual memory performance is slower as compared to RAM.

Q.8 Define a system call.

Ans. A system call is a way for programs to request services from the OS. It allows access to hardware resources like files and memory. Examples include file creation, reading, and writing operations.

Q.9 What is the role of a file system?

Ans. A file system organizes and stores data on storage devices. It helps users create, read, update, and delete files easily. It also manages file names and storage locations.

Q.10 Define multitasking in computer system.

Ans. Multitasking means running more than one program at the same time. The operating system switches between tasks very quickly.

Long Questions

Q.1 Describe the architecture of an operating system, explaining kernel, shell, and layered design with examples. [10501191]

Ans. See Long Question No. 3 (detailed answer in Unit content, Q.3).

Q.2 Explain the process lifecycle in detail with examples. [10501192]

Ans. See Long Question No. 5 (detailed answer in Unit content, Q.5).

Q.3 Differentiate between RAM and virtual memory, and explain how multithreading improves performance. [10501193]

Ans. Difference between RAM and Virtual Memory:

RAM	Virtual Memory
RAM (Random Access Memory), also called primary memory, is the main working area of a computer where data and instructions are stored temporarily.	Virtual memory is part of the computer's storage drive used as temporary RAM when RAM is full.
It is a fast storage area that allows the CPU to access information quickly during program execution.	Storage drives operate at lower data transfer speeds and have higher access times compared to RAM, so virtual memory is slower.
Data stored in RAM is erased when the computer is turned off.	It allows more programs to run at the same time by using space from HDD, SSD, or NVMe.

Multithreading and Performance

Multithreading is an operating system technique that allows a single process to perform multiple tasks at the same time by dividing its work into smaller units called threads. Each thread runs independently but shares the same memory and resources of the process.

Multithreading improves performance because tasks can be divided into multiple threads and executed in parallel, allowing complex operations to complete more quickly, improving overall system efficiency, and keeping applications responsive.

Q.4 Describe system calls, their purpose, and give examples of at least three types. [10501194]

Ans. See Long Question No. 10 (detailed answer in Unit content, Q.10).

Q.5 A computer system uses the First-Come, First-Served (FCFS) scheduling method to manage processes. The following table shows the arrival time and the CPU burst time (time needed for execution) of three processes: [10501195]

Process	Arrival Time (seconds)	Burst Time (seconds)
P1	0	4
P2	1	3
P3	2	1

Requirements

7. Arrange the processes in the order they will be executed according to the FCFS scheduling method.
8. Show the execution sequence of all processes.

9. Calculate the start time and completion time for each process.
10. Find the waiting time for each process.
11. Calculate the average waiting time for all processes.

Ans. — Step 1: Understand FCFS Scheduling

First-Come, First-Served (FCFS) is a non-preemptive scheduling algorithm where processes are executed in the order of their arrival times. The process that arrives first gets the CPU first and runs to completion before the next process starts.

Given processes:

- P1: Arrival = 0s, Burst = 4s
- P2: Arrival = 1s, Burst = 3s
- P3: Arrival = 2s, Burst = 1s

Step 2: Arrange Processes in Execution Order (Requirement I)

Sort processes by arrival time: P1 → P2 → P3

Step 3: Execution Sequence and Gantt Chart (Requirement II)

- Start at time 0.
- P1 runs from 0s to 4s.
- P2 runs from 4s to 7s.
- P3 runs from 7s to 8s.

P1	P2	P3
0s – 4s	4s – 7s	7s – 8s

Step 4: Start Time and Completion Time (Requirement III)

Process	Start Time (s)	Completion Time (s)
P1	0	4
P2	4	7
P3	7	8

P1: Start = 0s (arrives and starts immediately); Completion = Start + Burst = 0 + 4 = 4s

P2: Start = Completion of P1 = 4s (waits until P1 finishes); Completion = 4 + 3 = 7s

P3: Start = Completion of P2 = 7s (waits until P2 finishes); Completion = 7 + 1 = 8s

Step 5: Calculate Waiting Time for Each Process (Requirement IV)

Waiting time = Start time – Arrival time (time spent waiting in the ready queue before execution begins).

Process	Waiting Time (s)
P1	0
P2	3
P3	5

- P1: Waiting time = $0 - 0 = 0s$
- P2: Waiting time = $4 - 1 = 3s$
- P3: Waiting time = $7 - 2 = 5s$

Step 6: Calculate Average Waiting Time (Requirement V)

- Total waiting time = $0 + 3 + 5 = 8s$
- Number of processes = 3
- Average waiting time = Total waiting time / Number of processes = $8 / 3 = 2.6667s$ (or $8/3 s$)

Solved Activities

Activity 1

Objective:

Identify the shell, device driver, and system library used by your computer.

Steps

12. Locate the shell on your computer (desktop or command-line).
13. Open Device Manager (Windows) or System Information (Mac/Linux).
14. Find one hardware device (e.g., printer, keyboard) and note its driver name.
15. Open a simple program (e.g., Calculator) and identify one task it performs that may use a system library (e.g., displaying numbers).
16. Record your findings in a table with these columns: OS Component (Shell / Device Driver / System Library), Your Example (e.g., Windows desktop, HP printer driver, Calculator app), Purpose/Role (what it does in the system).

Solution

- On a Windows computer, the shell is the Windows Desktop.
- If you're using the command line, you can use Command Prompt or PowerShell.

Example: Windows Desktop. Purpose: It helps you interact with your computer and sends commands to the main part of the operating system (the kernel).

2. Open Device Manager or System Information

- On Windows: Press 'Windows + X' and choose Device Manager.
- On Mac: Go to System Information.
- On Linux: Use the 'lshw' command or check System Information.

3. Find a Hardware Device and Its Driver

- In Device Manager, click on Keyboards and note the driver's name (like HID Keyboard Device).

Purpose: It helps the operating system talk to the hardware, like your keyboard. Example: Keyboard.

4. Open a Simple Program and See a Task Using System Library

- Task: Showing numbers on the screen.
- This task uses system libraries, which are like ready-made tools.
- The calculator doesn't have to build these from scratch — it can just use them.

Purpose: These libraries give easy-to-use instructions for common tasks, like showing numbers or text.

Example: Calculator App.

5. Findings Table

OS Component	Example	Purpose
Shell	Windows Desktop	Helps you interact with the computer and send commands
Device Driver	HID Keyboard Device	Lets the OS talk to the keyboard hardware
System Library	Calculator App Library	Gives instructions for tasks like showing

OS Component	Example	Purpose
		numbers

Activity 2

Objective:

Use the First Come, First Served (FCFS) method to determine the order and completion time of processes.

Steps

17. Consider the following processes:

Process	Arrival	Time
P1	0s	4s
P2	1s	5s
P3	2s	3s

18. Arrange the processes in the order they will run according to the FCFS method.

19. Calculate the start and finish time for each process.

- What is the total time taken for all processes to finish?
- Which process had to wait the longest?

Solution

Process	Arrival Time (s)	Time Needed (s)
P1	0	4
P2	1	5
P3	2	3

Step 2: Arrange processes by FCFS

FCFS (First Come, First Served) means processes are executed in the order of arrival time. Order: P1 → P2 → P3

Step 3: Calculate Start and Finish Times

- Arrival: 0s → Start: 0s
- Time Needed: 4s → Finish: $0 + 4 = 4s$

2. P2

- Arrival: 1s → Start: $\max(4, 1) = 4s$
- Time Needed: 5s → Finish: $4 + 5 = 9s$

3. P3

- Arrival: 2s → Start: $\max(9, 2) = 9s$
- Time Needed: 3s → Finish: $9 + 3 = 12s$

Process	Arrival	Start	Finish
P1	0	0	4

Process	Arrival	Start	Finish
P2	1	4	9
P3	2	9	12

Step 4: Answer the Questions

- Total time taken for all processes to finish: 12s
- Process that had to wait the longest: P3
- Waiting Time = Start Time - Arrival Time = 9 - 2 = 7s

Activity 3**Objective:**

Demonstrate how system calls work when managing files.

Instructions

- Give each student a small card labeled with a file name (e.g., photo.jpg, homework.docx, song.mp3).
- Write "open", "read", "write", and "close" on the board.

2. Steps

- Call one student the "Operating System".
- Another student plays the "Application" (e.g., Photo Viewer).
- The Application requests: "OS, please open photo.jpg".
- The OS responds by selecting the card from the "file list" and handing it over.
- Repeat with read, write, and close for different files.

3. Wrap-up Discussion

- Explain that each action represents a system call

Solution

- Give each student a card with a file name: photo.jpg, homework.docx, song.mp3, etc.
- Write these system calls on the board: open, read, write, close

Step 2: Role Assignment

- One student = Operating System (OS)
- One student = Application (e.g., Photo Viewer, Music Player)
- Remaining students = Files (holding file cards)

Step 3: Demonstration (System Calls in Action)

System Call	Application Says	OS Action	Meaning
OPEN	"OS, please open photo.jpg"	Searches among the file students, finds the correct file, hands it to the Application	The OS locates the file and gives access to it.
READ	"OS, please read photo.jpg"	Pretends to extract data from the file, tells the Application what's inside (e.g., "This is an image")	The OS reads data from the file and sends it to the application.

System Call	Application Says	OS Action	Meaning
WRITE	"OS, please write to homework.docx"	Goes to that file student, pretends to add or update information	The OS modifies or saves data into the file.
CLOSE	"OS, please close photo.jpg"	Takes the file back, returns it to the file group	The OS removes access and frees resources.

Step 4: Wrap-Up Discussion — Explain to students:

- Every request (open, read, write, close) is a system call
- Applications cannot access files directly
- They must always go through the Operating System
- The OS acts like a manager or gatekeeper