CHAPTER 1 Operating System: Structure and Services

Student Learning Outcomes

By the end of this chapter, you will be able to:

- Define an operating system and explain its role as an interface between the user and translate user actions into hardware-level instructions.
- Identify the responsibilities of the OS in managing multiple user accounts.
- Differentiate between the kernel and shell, including their specific functions.
- Explain OS layers and how each layer interacts with the others.
- Identify the role of system libraries and device drivers in OS functionality.
- Explain the stages in the process lifecycle (creation, execution, termination) with examples and apply the First-Come, First-Served (FCFS) process scheduling.
- Describe multitasking and concurrency, giving one real-world analogy.
- Distinguish between primary memory (RAM) and virtual memory.
- Define a process and a thread, and explain how threads share resources within a process.
- Give examples of how multithreading improves performance in applications.
- Define a system call and explain its purpose and types.
- Define file system, files, folders, and metadata, and explain their roles in organizing data.
- Explain how these components work together to process a user request from device to server and back.

Subjective Type Questions

Long Questions prepared in the light of new Examination Techniques
(Knowledge, Understanding, Application, Analytical & Conceptual)

Introduction

Q. What do you know about core concepts of operating system?

Ans. This chapter will discuss the core functions and structure of an operating system. The operating system is known as fundamental software that manages all the activities of computer hardware, their processes and working and facilitates seamless user interaction. The discussion will cover:

- **Operating system architecture:** Its role as the central controller of the computer, process management concepts such as the
- **Core Concepts:** Process lifecycle, Multitasking, Concurrency, and Scheduling
- **Hardware and Resources:** Memory management, Multithreading, and the use of system calls to securely access hardware resources.

- **Organization and OS Types:** File system organization and the main types of operating systems, including real time, embedded, network, and mobile OS.

1.1 Introduction to Operating System (OS)

Q.1: What is Operating system. What are the main purposes of operating system? Also give some examples of popular operating systems?

Ans.

Operating System (OS) is a type of system software that facilities and manages all the activities of computer hardware connect via wire or wireless. It also manages and execution of application software, and provides a user interface.

Main Purposes of Operating System: The operating system works as the central controller of the computer system. It ensures smooth and secure interaction between the user and the hardware components, especially in multi-user environments.

Popular Operating Systems:

- Windows
- macOS
- Linux
- Android
- iOS

Q.2. Why operating system is called central system controller. Also discuss the role of OS in user hardware interaction.

Ans:

Operating System is also called central system controller because it decides which task should be done first when more process arrived, how the computer's memory is used, and which devices (like a printer or speakers) should be active at a given time. Operating system makes sure that the computer works smoothly and efficiently if multiple tasks are running at the same time.

Role in user-hardware interaction: Computer hardware itself cannot understand human language. The operating system acts as a translator between the user application and the hardware components. This allows people to use computers easily without learning how the hardware works.

Example: If you attached a new printer with your system unit, windows OS install its driver and make device ready to use.

DO YOU KNOW ?

Humans speak languages like Urdu or English, whereas computers understand only binary language, or ON and Off switch i.e. 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).
- 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.

Q.3. Explain the responsibilities of operating system in multi-user environment. How user accounts are create and managed?

Ans:

In today's era especially online systems, more people can access the same operating system which demands operating system that manege multiuser environment. In multiuser environment, OS ensure:

- **User Accounts:** The creation and management of separate user accounts.
- **Privacy:** Keeps each user's files and information private.
- **Resource Sharing:** Shares the computer's resources fairly so that no one slows down the systems functionalities.
- **Unauthorized Access:** Protect shared resources from unauthorized access.

Creating and Managing User Accounts: The operating system allows you to create and manage different window accounts for different users. Each user has its own desktop, files, settings, and passwords.

Example: In computer lab, student's login with a username and password provided by the school. This keeps each student's work separate and private.

Procedure to Create a new Account: A new account can be created through the Settings or Control Panel

- Open Control Panel from windows setting.
- Select User Accounts
- Click on Add New User
- Entering details like username, password, and account type(Standard, Administrative, Guest).
- Click create

The new user account with mentioned credentials is created. Now you can check by restart your system.

1.2 Architecture of an Operating System

Q.4. Explain architecture of operating system also describe kernel and shell the core concept of operating system.

Ans:

The architecture of an operating system describes the way that how different parts of OS are organized and how they work together. Each part has a specific task for which he designed for, and together they make the computer work smoothly. The main purpose of OS is to make sure that all processes and execution done smoothly and efficiently.

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

Shell: The shell is the desktop screen of the operating system which interacts with the user via pointing devices or keyboard. It receives commands from the user and passes them to the kernel. There are two types of shell:

Types:

- **Graphical shells:** It is a graphical interface like MS Windows where you click icons and use menus via pointing devices e.g. Mouse or Touch Screen
- **Command-line shells** It is a command line interface like MS DOS or command prompt where you type instructions via keyboard.

Graphical shells	Command-line shells
1. It is based on graphical user interface	1. It is based on command line interface
2. It contain icons, menus and button control via pointing devices	2. It contain commands which are remembered and typed via keyboard
3. Easy to use	3. Difficult to use

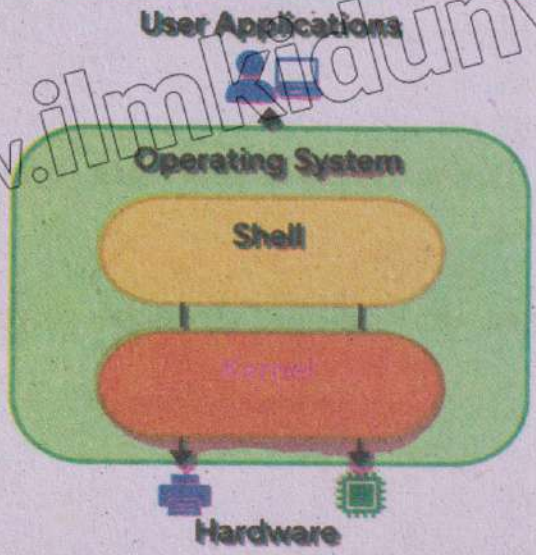

Figure 1.1: Diagram to illustrate the relationship between the user, shell, and kernel in an operating system.

Q.5. Explain operating system layers and modules.

Ans:

In most operating systems especially in MS Windows, the overall design is divided into different layers, where each layer has a specific job. Each layer depends on the layer below it. This design makes the operating system easier to manage, repair, and improve without changing the whole system.

Types of Layers:

Lower layer: This layer work directly with hardware devices like the CPU, RAM, and

storage, Hard disk.

Middle layer: This layer manages all the resources for multiple tasks running at a time and make sure programs can use them when needed.

Upper layer: This layer run application softwares and provide the interface that the user views on the screen.

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

- **Wires and Modems:** Work at the base, connectivity and signal passing identification through different devices.
- **Network Software:** Manages resources and other network nodes (like the middle layers managing memory and storage).
- **Operating system:** Use these arrangements to check and provide all the features to every computer or node (like the upper layers running applications and interacting with users).

Q.6. What are System Libraries? Also give an example. Define Device Drivers.

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.

Example: When a photo editing application 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: Device drivers are special programs that allow the operating system to communicate with hardware devices such as printers, keyboards, and graphics cards.

Activity

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

Steps:

1. Locate the shell on your computer (desktop or command-line).
2. Open Device Manager (Windows) or System Information (Mac/Linux).
3. Find one hardware device (e.g., printer, keyboard) and note its driver name.
4. Open a simple program (e.g., Calculator) and identify one task it performs that may use a system library (e.g., displaying numbers).
5. 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

Table of Findings:

OS Component	Your Example	Purpose/Role
Shell	Windows Desktop (GUI)	Provides the main graphical user interface to interact with the operating system, launch applications, and manage files.

Device Driver	"HID Keyboard Device" driver	Allows the operating system to communicate with and receive input from the connected keyboard.
System Library	Calculator app (Displaying numbers)	Uses a system graphics library (e.g., GDI++ or DirectX) to render numbers and buttons on the screen, instead of coding its own display method.

1.3 Process Management in Operating System (OS)

Q.7. What is process management in operating system? Also define process and resources.

Ans:

In process management, the operating system is responsible for managing all the programs (system or application) that run on a computer. Process management is one of the most important jobs of an operating system because it ensures all processes run smoothly without disturbing each other.

Example: Suppose if you are browsing on internet while listening music, operating system manages both process with all background running applications.

Process: A process is a program (or application) that is currently loaded into memory and being executed by the computer's CPU. It represents its own allocated memory space, state, and system resources. The operating system manages and schedules processes to run efficiently, even when multiple programs are active at the same time.

Resources: Resources are the physical or virtual components of a computer system that are required by a process to execute. These include:

- CPU time - for computation and task execution.
- Memory (RAM) - for storing the program code and data while it runs.
- I/O devices - such as keyboard, mouse, printer, or network access.
- Storage space - for reading and writing files.

Q.8. What is process life cycle? Give an example to support your answer.

Ans:

A process goes through several stages during its life cycle. The stages of process life cycle are as follows

1. **Creation:**
   - These steps refer to start an application like MS Word.
   - The OS loads the program into memory and gives it the resources (like CPU time and memory) it needs.
2. **Execution:**
   - The process is actively running by CPY and performing tasks effectively.
3. **Termination:**
   - The process finishes its task and is closed by the user or the system.

- The OS frees the resources so they can be used by other processes.

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

1. Creation:

Begins when the user clicks the browser icon. The operating system loads the program into memory and allocates the required resources.

2. Execution:

The browser performs tasks such as loading web pages, displaying media, and responding to user actions.

3. Termination:

Occurs when the browser is closed. The operating system stops the process and releases its resources for other uses.

Q.9. What is meant by multitasking and concurrency? Also write the difference between them.

Ans:

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

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.

Concurrency: In concurrency, more than one process is activated at the same time in an operating system, but the CPU processes them one by one in extremely fast cycles.

Example: You are using a computer with a single CPU core. You have three programs "running" at the same time and all three are active concurrently.

Multitasking	Concurrency
Allows more than one program to be open and usable by a single user at the same time	more than one process is activated at the same time in an operating system, but the CPU processes them one by one
User can see and use different programs together	Multiple programs are running in back ground but user can see only one who is currently active in screen
Multitasking support both single or multi core CPU	Concurrency only support single core CPU

Q.10. What is process scheduling? Also explain First Come First Served Concept.

Ans:

When many processes can be in progress during the same time period and the CPU works on them one at a time in very fast turns. Then CPU decide which process should run first and how long each process should run. This process is called scheduling. The

CPU cannot run all processes at the same time, the operating system must decide which scheduling methods used by operating systems. There are many scheduling methods but one of them is First Come First served (FCFS) scheduling method.

First Come First Served Scheduling (FCFS):

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.

Q.11. Explain FCFS method with the help of an example.

Ans:

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

Solution:

Step-by-Step Explanation:

1. **At 0 seconds:**
   - P1 arrives first, so it starts running immediately.
   - It requires 5 seconds to finish.
   - It runs from 0s to 5s.
2. **At 1 second:**
   - P2 arrives while P1 is still running, so P2 waits.
3. **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.
4. **At 5 seconds:**
   - P1 finishes.
   - The CPU now starts P2, which needs 3 seconds to complete.
   - P2 runs from 5s to 8s.
5. **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]

Time (s)	0 - 5s	5s - 8s	8s - 10s
Running Process	P1	P2	P3

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.

Q.12. Describe advantages and disadvantages of FCFS.

Ans: Advantages:

- Easy to understand and implement
- As processes are served in the exact order they arrive
- No process is skipped.

Disadvantages:

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

DO YOU KNOW?

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.

Activity

Objective:

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

Steps:

1. Consider the following processes:

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

2. Arrange the processes in the order they will run according to the FCFS method.
3. Calculate the start and finish time for each process.

Answer the questions:

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

Solution:

Step-by-Step Explanation:

1. **At 0 seconds:**
   - **P1** arrives first, so it starts running immediately.
   - It requires **4 seconds** to finish.
   - It runs from **0s to 4s**.
2. **At 1 second:**
   - **P2** arrives while P1 is still running, so **P2** waits.

3. **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.
4. **At 4 seconds:**
   - P1 finishes.
   - The CPU now starts P2, which needs 5 seconds to complete.
   - P2 runs from 4s to 9s.
5. **At 9 seconds:**
   - P2 finishes.
   - The CPU now starts P3, which needs 3 seconds.
   - P3 starts and runs from 9s to 12s.

Execution Timeline:

[P1: 0-4] → [P2: 4-9] → [P3: 9-12]

Time (s)	0 - 4s	4s - 9s	9s - 12s
Running Process	P1	P2	P3

Answers:

1. The total time to finish the all processes is 12 seconds.
2. Even though P3 has the shortest duration (only 3 seconds), it must still wait until P1 and P2 have finished because they arrived earlier.

7.4 Memory

Q.13. What is memory? Also explain primary and virtual memory.

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. There are two types of memory used in computer:

1. Primary Memory
2. Virtual memory

Primary Memory (RAM):

- Primary memory, also called Random Access Memory (RAM)
- It is the main working area of a computer.
- It is a fast storage area
- The computer keeps the data and instructions temporarily in it.
- Its data is erased when the computer is turned off that's why it is also called volatile memory

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.

Virtual Memory:

- When the RAM is full, the operating system uses part of the computer's storage drive like Hard drives (HDD) or 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.

Advantage of Virtual memory:

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.

DO YOU KNOW?

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.

1.5 Process and Threads

Q.14. Describe the concept of process and threads. Also write down the difference between them.

Ans: Process:

- 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 (such as files or network connections).
- Processes are isolated from one another to ensure stability and security.
- A problem in one process generally does not affect another.

Example: If you are running a web browser it is entire a one process.

Thread:

- Thread, on the other hand, is the smallest unit of execution within a process.
- Multiple threads can exist inside a single process, each performing a different task.
- All threads in the same process share the same memory and resources, but operate independently.

Example: Within the web browser process, different threads are responsible for:

- One thread is loading and rendering a webpage.
- The second thread is playing audio or video content.
- Another thread is downloading files in the background

Q.15. What is multithreading? Also explain the benefits of multithreading.

Ans:

Multithreading is an operating system technique that allows a single process to

perform multiple tasks at the same time. In multithreading operating system dividing its work into smaller units called threads. Each thread runs independently but shares the same memory (RAM) and resources of the process. Multithreading enabling faster execution, better responsiveness, and efficient use of system resources.

Benefits of Multithreading

- **Enhanced Performance:** All tasks can be divided into multiple threads and executed in parallel. It allowing complex operations to complete more quickly. It also improving overall system efficiency.
- **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 checks spelling in the background.

- **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.
- **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.

1.3 System Calls

Q.16. What is system call? Give an example.

Ans:

- System call is a request made by a program to the operating system to perform a specific task
- The program cannot do any task directly.
- They act as a bridge between user programs and the kernel
- It allows different applications to access hardware and core OS functions safely.
- Without system calls, programs that 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).

Q.17. Explain all the types of systems calls.

Ans: The main types of system calls include:

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.

DO YOU KNOW

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.

1.7 File System Structure and Management

Q.18. What is file system? Also explain file, folder, metadata and files systems with example

Ans:

File system 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 contain complex data.

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

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.

Metadata: Metadata refers to details about a file 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.

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 save its location so it can be accessed later. Some widely used file systems include computer-based file systems are including:

- **FAT 32** - Used in USB flash drives
- **NTFS** - Used by Windows OS.
- **APFS / HFS+** - Used by macOS.
- **EXT4** - Commonly used by Linux OS.

Example: 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.

1.8 Types of Operating Systems

Q.19. Explain types of operating systems with their key features and application in daily life?

Ans:

Real-Time Operating System (RTOS): A Real-Time Operating System is designed to process data and respond within a harsh time limit, known as a deadline. It is used where

even a tiny delay can cause system failure or serious consequences.
Key Feature: It processes tasks immediately as they arrive, without long waiting times.
Applications: It is used in critical systems such as air traffic control, heart-monitoring devices, or industrial robots.
Embedded Operating System (EOS): An Embedded operating system 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: It uses very little memory and power, and is often stored permanently inside the device.
Application: It is used in home appliances (microwaves, washing machines), printers, smart TVs, and ATMs.
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 like windows multipoint servers.
Key Feature: It focuses on communication and coordination between computers.
Application: It is used in Offices, schools, and data centers.
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: It supports wireless connectivity, cameras, sensors, and app stores.
Application: It is used in smartphones, tablets, smartwatches.

DO YOU KNOW ?
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.

Objective Type Questions

Additional MCQ's

Multiple Choice Questions prepared in the light of new Examination Techniques
(Knowledge, Understanding, Application, Analytical & Conceptual)

Introduction

1. **Fundamental software that manage hardware and user interaction:**
   (A) System software (B) Operating system
   (C) Windows (D) Linux
2. **Central controller of computer is?**
   (A) Windows (B) OS
   (C) Application software (D) Device drivers

1.1 Introduction to Operating System

3. **Main purpose of operating system is:**
   (A) Ensuring smooth work

(B) Secure interaction between hardware & user
(C) Provide multi-user environment (D) All above

4. Act as translator between user and hardware:
(A) Windows (B) Linux (C) Android (D) All

5. Binary code of A is:
(A) $(01000001)_2$ (B) $(10100001)_2$ (C) $(01111111)_2$ (D) $(01111110)_2$

6. Which of the following is not the task of OS?
(A) Manage user accounts (B) Share resources
(C) Manage cash (D) Protect access

7. Multi-user OS is one who?
(A) Install softwares (B) Manage hardware
(C) Secure each user's file and information
(D) Utility program

8. In multi-user OS, which one of the following is not advantage:
(A) Every user has its own printer (B) Every can access resources
(C) Every user has its own setting (D) All above

9. A new account can be created by:
(A) Desktop (B) Icons (C) Control Panel (D) Start Menu

10. Which allow people to use computer without knowing working of hardware?
(A) Desktop (B) Icons (C) Control Panel (D) Start Menu

1.2 Architecture of an Operating System

11. Architecture of computer was designed by?
(A) Charles Babbage (B) Bill Gates
(C) Von Neumann (D) J. J. Thomson

12. The core part of OS is:
(A) Kernel (B) Shell (C) Windows (D) Anti-virus

13. The outer part of OS is:
(A) Kernel (B) Shell (C) CPU (D) RAM

14. Commands received from user process by:
(A) Kernel (B) Shell (C) Both (D) None

15. How many types of shells are there?
(A) 1 (B) 2 (C) 3 (D) 4

16. Which of the following layer directly work with hardwere devices?
(A) Upper layer (B) Middle layer (C) Lower layer (D) OS layer

17. Provide user interface and other screen features?
(A) Upper layer (B) Middle layer (C) Lower layer (D) OS layer

18. Which program or instructions allow OS to communicate with hardware?
(A) Upper layer (B) Middle layer (C) Lower layer (D) OS layer

19. **Command prompt or terminal is also known as:**
(A) Graphics shell (B) Graphic kernel
(C) Command line shell (D) Command line kernel

20. **Resources like RAM or CPU registers are managed by:**
(A) Kernel (B) Shell (C) Windows (D) OS

1.3 Process Management in Operating System (OS)

21. **When a program runs in an OS, it is called?**
(A) Fetching (B) Decoding (C) Storing (D) Process

22. **The most important job of an operating system:**
(A) Running protocols (B) Ensure all processes run smoothly
(C) Manage memory (D) Scan virus

23. **Process life cycle contains steps:**
(A) 1 (B) 2 (C) 3 (D) 4

24. **In which process step OS frees the resources so they can be used by others?**
(A) Creation (B) Termination (C) Execution (D) Storage

25. **The processes running at the same time are called:**
(A) Concurrency (B) Multitasking (C) Process management (D) Execution

26. **When task performed by more than one processor it is called?**
(A) Multitasking (B) Multiprocessing
(C) Multithreading (D) Multi management

27. **Which of the following is a scheduling technique?**
(A) RR (B) FIFO (C) LIFO (D) FCFS

28. **Which of the following is an advantage of First Come First Serve?**
(A) Shorter task delayed (B) Less ready time
(C) Last executes first (D) None

29. **A very long waiting time can affect the:**
(A) Time limit (B) Efficiency (C) Cost (D) All above

30. **FCFS scheduling is used when user does:**
(A) Multitasking (B) Execution by itself
(C) Download file (D) Listen music

1.4 Memory

31. **Why RAM is called volatile memory?**
(A) Fast access (B) High speed
(C) Temporary nature (D) Permanent nature

32. **NVMe is used in:**
(A) RAM (B) ROM (C) External USB (D) SSD or HDD

33. **First computer IBM 5150 was designed in:**
(A) 1981 (B) 1974 (C) 1998 (D) 2010

34. **Use of virtual memory can reduce:**
(A) Cost (B) Performance (C) Execution time (D) Processing

35. **DDR5 RAM can transfer data at rate of:**
(A) 10 Gbps (B) 20 Gbps (C) 30 Gbps (D) 50 Gbps

1.5 Process and Threads

36. **Inside which shares and ensures stability?**
(A) Task (B) Process (C) Thread (D) All

37. **Smallest unit of memory is:**
(A) Bit (B) Byte (C) Process (D) Thread

38. **Which of the following is a feature of thread?**
(A) Ensure stability (B) Faster execution
(C) Manage speed & execution (D) Operate independently

39. **Which of the following has its own memory, execution & CPU time?**
(A) Task (B) Thread (C) Process (D) Schedule

40. **Which of the following is a feature of Multithreading?**
(A) Each thread has same execution (B) Each thread perform same task
(C) Each thread perform a different task
(D) All above

41. **Which of the following is a advantage of multithreading?**
(A) Improve speed (B) Improve efficiency
(C) Improve responsiveness (D) All above

42. **Which of the following will use multithreading more as compared to others?**
(A) Web browser (B) MS Word (C) Video games (D) Music App

43. **Threads and other memory resources are handled by:**
(A) Kernel (B) Shell
(C) Operating System (D) Application Software

44. **Which is called smallest unit of execution?**
(A) Process (B) Thread (C) Bit (D) Byte

45. **A thread is a:**
(A) Physical component (B) Virtual component
(C) Unit of software (D) None of these

1.6 System Calls

46. **The act as a bridge between user and the kernel:**
(A) System Process (B) System Resources
(C) System calls (D) System ports

47. **Without system hardware directly control by program?**
(A) Un-safe (B) Reliable (C) Both (D) None

1.7 File System Structure and Management

48. How many types of system calls?
(A) 1 (B) 2 (C) 3 (D) 4

49. Which of the following type of system call that open file for writing?
(A) Open (B) Read (C) Write (D) Fork

50. Which system call type send data files & output devices?
(A) Open (B) Write (C) Close (D) Fork

51. Who provide a structure way to manage program & data?
(A) File system (B) System calls (C) Kernel (D) Shell

52. File system ensure:
(A) Files are secure from being hacked
(B) Files free from virus
(C) Files easily retrievable (D) Files are free to move

53. The system can manage files in:
(A) SSD (B) RAM (C) Flash drives (D) All above

54. We can identify a file type by its:
(A) Name (B) Size (C) Extension (D) Location

55. File which access by macros:
(A) FAT32 (B) NTFS (C) APFS (D) EXT4

56. Files commonly used in Linux:
(A) FAT32 (B) NTFS (C) APFS (D) EXT4

1.8 Types of Operating System

57. Industrial temperature control OS is an example of:
(A) Real Time OS (B) Embedded OS
(C) Network OS (D) Mobile OS

58. Which types of OS cannot be changed or cannot be general purpose computing?
(A) Real Time OS (B) Embedded OS
(C) Network OS (D) Mobile OS

59. Types of OS which is used in multi user databases:
(A) Network OS (B) Mobile OS (C) Windows (D) Linux

60. As compare to Linux & MacOS, window use ___ system calls:
(A) 1000 (B) 2000 (C) 3000 (D) 400

61. Symbian OS used in:
(A) IBM Computers (B) Nokia phones
(C) BlackBerry Phones (D) ATMs

62. Which OS is beneficial for sharing resources?
(A) Real Time OS (B) Mobile OS
(C) Embedded OS (D) Network OS

63. **Smart LEDs used OS:**
(A) Embedded (B) Mobile (C) Real Time (D) Network

Answers

1. B	2. B	3. D	4. D	5. A	6. C	7. C
8. A	9. C	10. B	11. C	12. A	13. B	14. A
15. B	16. C	17. A	18. B	19. C	20. A	21. D
22. B	23. C	24. B	25. B	26. B	27. D	28. D
29. B	30. A	31. C	32. D	33. A	34. B	35. D
36. B	37. A	38. D	39. B	40. C	41. D	42. C
43. A	44. B	45. B	46. C	47. A	48. D	49. A
50. B	51. A	52. C	53. D	54. C	55. C	56. D
57. A	58. B	59. A	60. B	61. B	62. D	63. A

Additional Short Question's

Short Answer Questions prepared in the light of new Examination Techniques
(Knowledge, Understanding, Application, Analytical & Conceptual)

Introduction

1. Define operating system and write names of some popular operating systems.

Ans: Operating System (OS) is a type of system software that facilities and manages all the activities of computer hardware connect via wire or wireless. It also manages and execution of application software, and provides a user interface.

Popular Operating Systems:

- Windows
- macOS
- Linux
- Android
- iOS

2. Which are the processes and operation perform or controlled by operating system?

Ans: Core Concepts: Process lifecycle, Multitasking, Concurrency, and Scheduling Hardware and Resources: Memory management, Multithreading.

Organization: File system organization.

1.1 Introduction to OS

3. How operating system works?

Ans: The operating system works as the central controller of the computer system. It ensures smooth and secure interaction between the user and the hardware components, especially in multi-user environments.

4. How operating systems handle user hardware interaction?

Ans: Computer hardware itself cannot understand human language. The operating system acts as a translator between the user application and the hardware components. This allows people to use computers easily without learning how the hardware works.

5. Describe any two responsibilities of operating system in multi-user environment.

Ans: User Accounts: The creation and management of separate user accounts.

Privacy: Keeps each user's files and information private.

6. How can we create a new user account in MS windows?

Ans: A new account can be created through the Settings or Control Panel

- Open Control Panel from windows setting.
- Select User Accounts
- Click on Add New User
- Entering details like username, password, and account type (Standard, Administrative, Guest).
- Click create

1.2 Architecture Of An Operating System

7. Differentiate between kernel and shell.

Ans:

Kernel	Shell
The kernel is the core part of the operating system	The shell is the desktop screen of the operating system
It directly controls the computer's system	It interacts with the user via pointing devices or keyboard
It decides how and when different programs can use the system resources.	It receives commands from the user and passes them to the kernel

8. Differentiate between graphic shells and command line shells?

Ans:

Graphical shells	Command-line shells
1. It is based on graphical user interface	1. It is based on command line interface
2. It contain icons, menus and button control via pointing devices	2. It contain commands which are remembered and typed via keyboard
3. Easy to use	3. Difficult to use

9. Describe the purpose of any one layers of operating system.

Ans: Middle layer: This layer manages all the resources for multiple tasks running at a time and make sure programs can use them when needed.

10. Why device drivers are important in Ms windo.

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

11. Differentiate between process and resource.

Ans:

Process	Resource
A process is a program that is currently loaded into memory	Resources are the physical or virtual components of a computer system
It represents its own allocated memory space, state, and system resources.	they are required by a process to execute.

12. What is process life cycle? Named its stages.

Ans: A process life cycle is a collection of phases in which a process complete its execution and give us output in the form we required. A process goes through several stages during its life cycle. The stages of process life cycle are as follows: Creation, Execution and Termination.

13. Differentiate between multitasking and concurrency.

Ans.

Multitasking	Concurrency
Allows more than one program to be open and usable by a single user at the same time	more than one process is activated at the same time in an operating system, but the CPU processes them one by one
User can see and use different programs together	Multiple programs are running in back ground but user can see only one who is currently active in screen
Multitasking support both single or multi core CPU	Concurrency only support single core CPU

14. What is FCFS? Give an example.

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.

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

15. Suppose we have three processes P1, P2 and P3 having arrival time 0s, 1s, 2s and burst time 3s, 4s and 2s respectively. How much total time required to complete all three process?

Ans:

Time (s)	0 - 3s	4s - 7s	8s - 10s
Running Process	P1	P2	P3

Total time to complete all three processes is 10s.

16. Describe advantages of FCFS.

Ans: Advantages:

- Easy to understand and implement
- As processes are served in the exact order they arrive
- No process is skipped.

17. Describe dis-advantages of FCFS.

Ans: Disadvantages:

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

1.4 Memory

18. How memory of computer works? Write the names of its types.

Ans: Memory 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. There are two types of memory used in computer:

1. Primary Memory 2. Virtual memory

19. Differentiate between primary memory and secondary memory.

Ans:

Primary Memory	Virtual Memory
Primary memory, also called Random Access Memory (RAM)	When the RAM is full, the operating system uses part of the computer's storage drive
It is a fast storage area	It is a slow storage area
It can increase system performance	It can reduce the system performance

20. Why virtual memory is compulsory for today's technology?

Ans: 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.

1.5 Process and Threads

21. Differentiate between process and thread.

Ans:

Process	Threads
A process is an independent program that is currently being executed	Thread is the smallest unit of execution within a process
It has its own memory space and CPU time	All threads in the same process share the same memory and resources
Processes are isolated from one another to ensure stability	Multiple threads can exist inside a single process

22. Describe any two benefits of multithreading.

Ans: i. All tasks can be divided into multiple threads and executed in parallel

ii. Applications remain responsive even when one thread is busy with a specific task.

1.6 System Calls

23. What is system call give an example?

Ans: System call is a request made by a program to the operating system to perform a specific task. The program cannot do any task directly.

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.

24. Why system calls are important for any operating system?

Ans: It allows different applications to access hardware and core OS functions safely. Without system calls, programs that directly control hardware can be unsafe and complex

25. Which system call is used to retrieve data from input devices? Give an example.

Ans: Read is a system call which is used Retrieves data from a file or input device.

Example: Reading text from a document.

26. What is fork call and why it is used? Give an example.

Ans: A type of system call which 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.

1.7 File System Structure and Management

27. Why file system is important for data management?

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

28. Differentiate between file and folder.

Ans:

File	Folder
A file is a collection of related data stored on a computer	A folder also called a directory is a container used to organize files
It may contain text, images, audio, video, or program instructions	Folders in a logical structure, making it easier to locate and manage information

29. Named computer-based file system used by different operating systems.

Ans: Some widely used file systems include computer-based file systems are including:

- FAT 32 - Used in USB flash drives
- NTFS - Used by Windows OS.

- APFS / HFS+ - Used by macOS.
- EXT4 - Commonly used by Linux OS.

1.8 Types of Operating System

30. Which operating system is not for general public and optimal for one task? Also define it.

Ans: An Embedded operating system 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.

Example: It is used in home appliances (microwaves, washing machines), printers, smart TVs, and ATMs.

31. Describe the key features and applications of real time operating system.

Ans: Key Feature: It processes tasks immediately as they arrive, without long waiting times.

Applications: It is used in critical systems such as air traffic control, heart-monitoring devices, or industrial robots.

32. What is network operating system? Also write its applications.

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 like windows multipoint servers.

Application: It is used in Offices, schools, and data centers.

Exercise Questions (Solved)

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

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

Ans: The operating system works as the central controller of the computer system because it ensures smooth and secure interaction between the user and the hardware components, especially when multi-user environments is activated and each user access same available resources again and again.

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

Ans: Computer hardware itself cannot understand human language. The operating system acts as a translator between the user application and the hardware components. This allows people to use computers easily without learning how the hardware works.

Example: If you attached a new printer with your system unit, windows OS install its driver and make device ready to use.

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

Ans: Procedure to Create a new Account:

A new account can be created through the Settings or Control Panel

- Open Control Panel from windows setting.
- Select User Accounts
- Click on Add New User
- Entering details like username, password, and account type(Standard, Administrative, Guest).
- Click create

The new user account with mentioned credentials is created. Now you can check by restart your system

4. How is kernel different from shell.

Ans:

Kernel	Shell
The kernel is the core part of the operating system	The shell is the desktop screen of the operating system
It directly controls the computer's system	It interacts with the user via pointing devices or keyboard
It decides how and when different programs can use the system resources.	It receives commands from the user and passes them to the kernel

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

Ans:

Example: The C Standard Library e.g., libc on Unix/Linux or the C Runtime Library on Windows.

Purpose: It provides a collection of pre-written, essential functions for common programming tasks. such as: Input/Output operation, memory management etc.

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

Ans: Advantage:

Sequential Access: As processes are served in the exact order they arrive

Dis-advantage:

Waiting Queue: Short processes may have to wait a long time if they are queued

7. Why is virtual memory slower than RAM?

Ans: 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.

8. Define a system call.

Ans:

- System call is a request made by a program to the operating system to perform a specific task
- The program cannot do any task directly.
- They act as a bridge between user programs and the kernel

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

9. What is the role of a file system?

Ans: File system 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 contain complex data

10. Define multitasking in computer system.

Ans: 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

Long Questions

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

Ans: Architecture:

The architecture of an operating system describes the way that how different parts of OS are organized and how they work together. Each part has a specific task for which he designed for, and together they make the computer work smoothly. The main purpose of OS is to make sure that all processes and execution done smoothly and efficiently.

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

Shell: The shell is the desktop screen of the operating system which interacts with the user via pointing devices or keyboard. It receives commands from the user and passes them to the kernel. There are two types of shell:

Types:

- Graphical shells: It is a graphical interface like MS Windows where you click icons and use menus via pointing devices e.g. Mouse or Touch Screen.
- Command-line shells It is a command line interface like MS DOS or command prompt where you type instructions via keyboard.

Types of Layers:

Lower layer: This layer work directly with hardware devices like the CPU, RAM, and storage, Hard disk.

Middle layer: This layer manages all the resources for multiple tasks running at a time and make sure. programs can use them when needed.

Upper layer: This layer run application softwares and provide the interface that the user views on the screen.

2. Explain the process lifecycle in detail with examples.

Ans: For Answer See Q No. 8.

3. Differentiate between RAM and virtual memory, and explain how multithreading improves performance.

Ans: Primary Memory (RAM):

- Primary memory, also called Random Access Memory (RAM)
- It is the main working area of a computer.
- It is a fast storage area
- The computer keeps the data and instructions temporarily in it.
- Its data is erased when the computer is turned off that's why it is also called volatile memory

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.

Virtual Memory:

- When the RAM is full, the operating system uses part of the computer's storage drive like Hard drives (HDD) or 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.

Multithreading:

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

Benefits of Multithreading:

- **Enhanced Performance:** All tasks can be divided into multiple threads and executed in parallel. It allowing complex operations to complete more quickly. It also improving overall system efficiency.
- **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 checks spelling in the background.

- **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.
- **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.

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

Ans: System call is a request made by a program to the operating system to perform a specific task

Purpose:

- The program cannot do any task directly.
- They act as a bridge between user programs and the kernel
- It allows different applications to access hardware and core OS functions safely.
- Without system calls, programs that 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).

Types of System Calls:

The main types of system calls include:

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.

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:

Process	6. Arrival Time (seconds)	7. Burst Time (seconds)
P1	8. 0	9. 4
P2	10. 1	11. 3
P3	12. 2	13. 1

Ans: Step-by-Step Explanation:

1. **At 0 seconds:**
   - **P1** arrives first, so it starts running immediately.
   - It requires **4 seconds** to finish.
   - It runs from **0s to 4s**.
2. **At 1 second:**
   - **P2** arrives while P1 is still running, so **P2** waits.
3. **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.
4. **At 4 seconds:**
   - P1 finishes.
   - The CPU now starts **P2**, which needs **3 seconds** to complete.
   - P2 runs from **4s to 7s**.
5. **At 7 seconds:**
   - P2 finishes.
   - The CPU now starts **P3**, which needs **1 seconds**.
   - P3 starts and runs from **7s to 8s**.

Execution Timeline:

[P1: 0-4] → [P2: 4-7] → [P3: 7-8]

Time (s)	0 - 4s	4s - 7s	7s - 8s
Running Process	P1	P2	P3