Introduction to Operating System

Introduction

We have studied the fundamental components of the computer hardware including CPU, memory, storage and Input/output devices. These components together form the backbone of the modern computer systems. The main objective of the computer system is to make it functional so that users can perform their tasks. Hardware alone can not fulfill this purpose as working directly with the hardware is almost impossible for the common person due to its complexity. That is where the operating system comes into the way to make the system user friendly through bridging the gap between hardware and user. An operating system is a software that manages all the hardware and provides the environment to run user applications efficiently.

Operating system works as resource manager and manages the CPU, memory, storage and input/output devices very efficiently and runs the applications without any errors. Operating system is divided mainly in two components: Shell and Kernel. The shell provides the user interface through which user can interact with operating system by issuing the commands, if it is text-based. Modern operating systems also provide the GUI (Graphical User Interface) based interface where user can interact by clicking the mouse. The kernel is the core of the operating system which provides the necessary software modules for managing all the hardware resources and the applications.

Modern operating systems have evolved over the years where initially it was serial processing with lots of human intervention. The batch processing is the improved version which allows grouping the tasks in a batch and a whole batch is executed with little or no intervention. Multiprogramming allows multiple programs to reside in memory at a time and improves the CPU utilization. Finally, the evolution of the time-sharing operating systems allowing multiple users to interact simultaneously laid the foundation of the modern and contemporary operating systems. The types of operating systems depend on the hardware environment on which it works. Considering this we can categorize them into desktop operating systems, server operating systems and mobile operating systems. Finally, we will learn about the major functions of the operating systems.

This chapter discusses the fundamentals of the operating systems, its evolution and types, and functions in details with examples wherever necessary.

What is an Operating System?

An operating system is a software which acts as an interface between hardware and user i.e application programs. The primary objective of the operating systems is

- Provide an interface to the user for running their application programs in convenient fashion.
- Act as a resource manager who allocates the various resources to the application programs including CPU, memory, storage and I/O devices in efficient manner.

Let us understand the role of operating system with a simple example. Assume that a manufacturing unit uses a machine to produce some items. First thing is we need to provide the raw materials as input. Once the raw materials are loaded, we have to operate the machine following step-by-step instructions provided in their user manuals to convert raw materials to the desired items as output.

Who will carry out this step-by-step process? A person called machine operator is carrying out this process. These types of environment need human intervention at every point in process. Let us assume that machine is fully automatic where almost no human intervention is needed. In this case, the automation hardware and logic built inside performs all the steps. This not only makes it convenient to operate the machine but also works in very efficient manner. This automation inside the machine is just like an operating system in the computer systems.

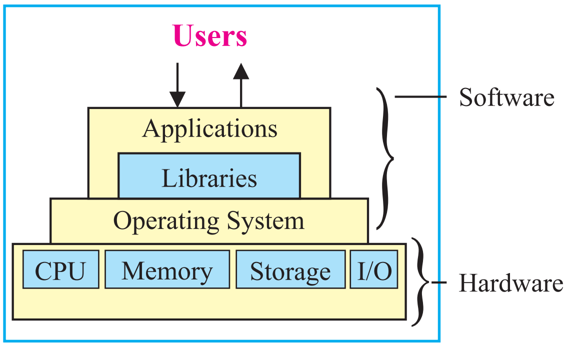


Figure 3.1 : Layered View of Computer System

Figure 3.1 shows the simplistic and layered view of the computer system divided into hardware and software layer. The hardware layer is the layer consists of CPU, memory, storage and I/O devices. The software layer further divided into operating system, libraries and application programs. The operating system hides the details of the hardware and allows the user to execute their application programs with the help of libraries. User runs their applications programs through the operating system and thus uses the hardware to carry out their tasks. User sends a request by clicking the mouse or providing input using keyboard to the application programs. Application programs in-turn internally use some library which sends request to the operating system and finally operating system activates necessary hardware to complete the request. The response will be sent to the user in reverse order and displayed on the screen. Hence, operating system plays a vital role in making use of the computer system very easy and convenient for the end users who even have no knowledge of the hardware.

Need of the Operating System

We know that any task on computer is ultimately performed by the hardware actions controlled and coordinated by the operating system. In absence of the operating system, one must have detailed knowledge of the hardware in order to perform any task which will limit the use of computer system by only expert users possessing the in-depth knowledge of hardware. That is where the need of the operating system arises. The operating system hides all the functional and circuit details of the hardware and allows user to perform the task by either writing programs or executing application programs. We can say that the operating systems have made the reach of the computers to common man by providing simple and human friendly interface and environment.

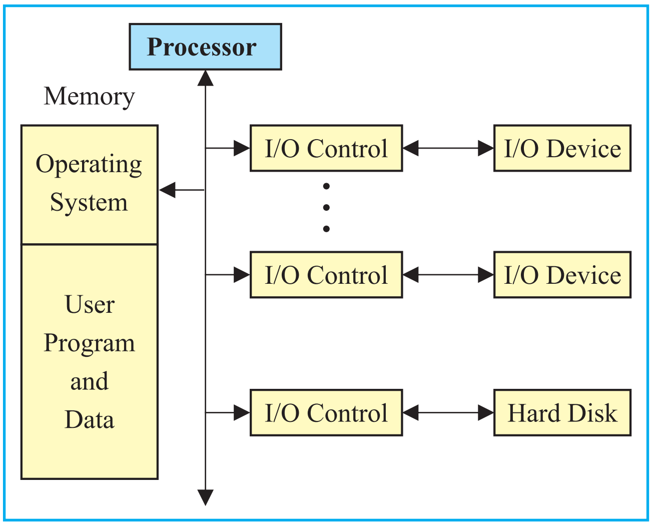


Figure 3.2 : Operating System as Resource Manager

The main job of the operating system, apart from providing interface, is to act as resource manager to ensure the efficient use of the hardware resource for executing the application programs. Figure 3.2 gives the detailed view of the operating system as resource manager.

Remember that operating system is also a software which executes on processor just like user programs or applications but the purpose is different. The main purpose is to provide the necessary environment for executing user programs by allocating necessary resource including processor, memory, storage and I/O devices. As shown in figure, some portion of the memory is occupied by the operating system and rest portion is used by the user programs and data. The portion of the operating system which remains in memory is its core or nucleus and known as kernel which provides all the services to manage the hardware for efficiently executing the user programs and applications. When computer system is started, kernel and other necessary functionalities of the operating system are loaded into memory, starts executing and provides interface to user to interact with the system. Once user program is loaded on request, execution control is transferred to the user program. After completing user program, control is regained by the operating system. Hence, control is switching between operating system and user programs or application as long as system in ON. We can say that at any time either part of operating system or user program is running. The operating system allocates the processor to the user program to execute and also allocates other resources dynamically for its execution. This makes the operating system a supervisor of the system which is used only in strict control of the operating system.

Components of Operating System

The operating system is mainly divided into two major components: Shell and Kernel. Let us discuss the role of both of them in overall working of the operating system.

Shell

We know that operating system provides the interface between user and hardware. The shell is the part of operating system which acts as user interface and directly interacts with the user. In other way, we can say shell is the face or voice of the operating system. It talks to you and listens what you want to do. It means it accepts the user request, pass it to the operating system where it is performed and produces the response which you can read. The shell can be either text-based or GUI (Graphical User Interface) based.

Most popular text-based operating system MS-DOS widely used on IBM-PCs earlier was using command-based interface. It is also called CLI (Command Line Interface). Once MS-DOS starts, it will display a prompt as

```
C:\>
```

That means, it is ready to accept the user commands. For example, following command lists the files and directories in current directory.

```
C:\>dir
```

After typing command "dir" on the prompt, once user presses the Enter key, the command is passed to the kernel for execution and the result i.e. list of files and directories will be listed on the screen and then command prompt displays again indicating control is returned back to operating system to receive the next input.

Modern operating systems like Microsoft Windows provides GUI based interface where user can interact with operating system using mouse clicks. By clicking a mouse on particular icon, operating system activates and requested task is performed. The results are displayed on the screen. In next chapter, we will get introduction of the Ubuntu operating system which supports both the types of shell interface, CLI and GUI. Normally, once you start Ubuntu, GUI interface is displayed. To start the command interface, you can switch to command prompt which will display separate window with text-based interface having the command prompt as above.

Kernel

The kernel is the core part of the operating system and it can be considered as brain of the operating system which handles all the important jobs inside the computer system. For example, when you start a new job, press a mouse button or enter data from keyboard, all of these are managed by kernel internally in smooth manner. The kernel directly communicates with all the hardware components of the computer system and ensuring smooth execution of user programs and applications. The kernel of the operating system manages the important tasks like executing the programs, managing the memory, manages the storage through file organization and handles the communication with I/O devices.

The command or request given by the user through the shell is passed to the kernel. The kernel fulfills the request by using the appropriate hardware components and completes the request and sends response back to user through the shell.

Evolution of Operating System

Modern operating systems that we use today are result of the development and innovation of many years. In initial years of computing, there were really no operating system and machines were used manually using switches and punch cards. Due to improvement in computer hardware, it was a need to use computers in faster and efficient way which resulted into gradual development of the operating systems. It evolved from serial processing to batch processing and then multiprogramming to time sharing systems. This section briefly discusses the evolution of operating systems.

Serial Processing

Early computers developed in 1940s and 1950s were having no operating systems and programmers were directly interacting with machines using toggle switches, display lights and printer. Program code in machine language was loaded using card reader. The errors during the execution were shown though lights and if completed normally, the output was sent to printer. The main problem of serial processing is scheduling i.e. in what fashion machine is accessible to programmers for executing their programs. One of the solutions is to create slots of 30 minutes. Each programmer can use the computer for 30 minutes when his/her turn comes. If programmer finishes earlier, time is wasted or if not finished in time, he/she will be in trouble. Another problem of this system is setup time. Once you get your slot, few minutes are wasted in setting up all the required things to execute the program. This method is called serial processing as users use system in serial fashion.

Batch Processing

Serial processing wastes the significant amount of processor time. Considering the high cost of the early computers, optimal utilization of the processor was important. Keeping this in mind, first batch operating system was developed by the General Motors in early 1950s for using on IBM 701 machine. Then it was refined and implemented for IBM 704 by many customers of the IBM. By the early 1960s, numbers of companies developed their own batch operating systems for their computer systems.

The basic idea of the batch operating system is that the monitor program which is the resident part of the operating system permanently resides in the memory. Rest of the memory is available for the user program and data. Programmers submit their program on card reader or tape to the operator. Operator creates the batch of these programs and put on the input device. These programs are executed one by one. Once the current program is completed, control is transferred back, monitor loads another program and transfer control to user program for its execution. Thus, control is switching between monitor and user program.

The problem of scheduling and setup time both are solved by monitor. The monitor schedules programs for execution one by one from the batch without intervention. These types of systems have two different modes of execution: user mode and kernel mode. A user program executes in user mode in which some areas of memory (area where monitor program resides) is restricted and some critical instructions are not executed. It is needed so that user program can not disturb the monitor program residing in memory. The monitor program executes in system mode known as kernel mode in which it can access restricted area of memory as well as can run privileged instructions. It is needed as operating system, monitor program here, acts as resource manager for all the hardware components and provides environment for executing user programs.

Multiprogramming

The batch programming solves the problem of automatic sequencing of the jobs or processes (program currently executing is called process). However due to the fact that I/O is much slower compared to processor, we can not use processor efficiently. The jobs are either CPU-intensive or I/O intensive or mix. The CPU-intensive jobs are most of the time performing computation with CPU while I/O intensive jobs are using I/O most of the time. When a process is busy with I/O, the processor remains idle and it results into the inefficiencies or we can say that processor utilization is not optimum. This is the characteristics of the uniprogramming i.e. only one job or process is running at a time.

In uniprogramming, the operating system occupies some portion of the memory and rest is used by the currently running process say P1. P1 executes for some time and then it requires to use I/O device, for example printer. While process P1 is using I/O, processor is idle and waiting for the I/O to finish so that it can resume the process P1. It is shown in figure 3.3. Once the process P1 finishes, monitor program removes it from memory and loads the next process and starts executing. This results into inefficient use of the processor.

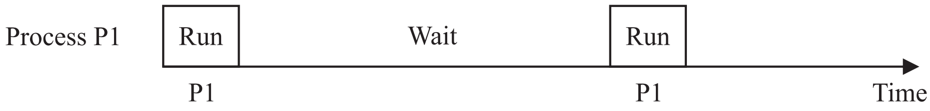
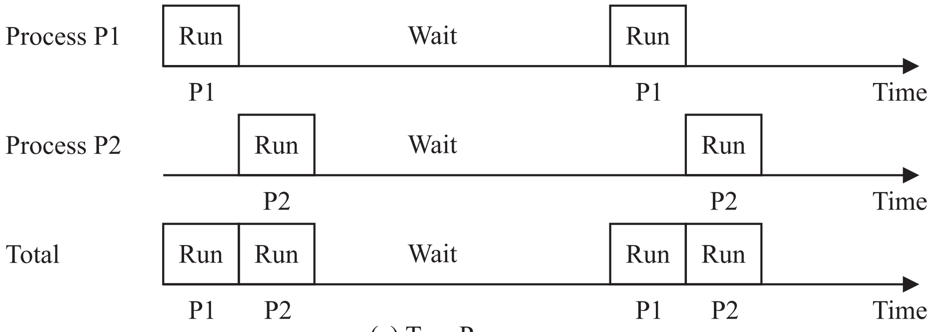
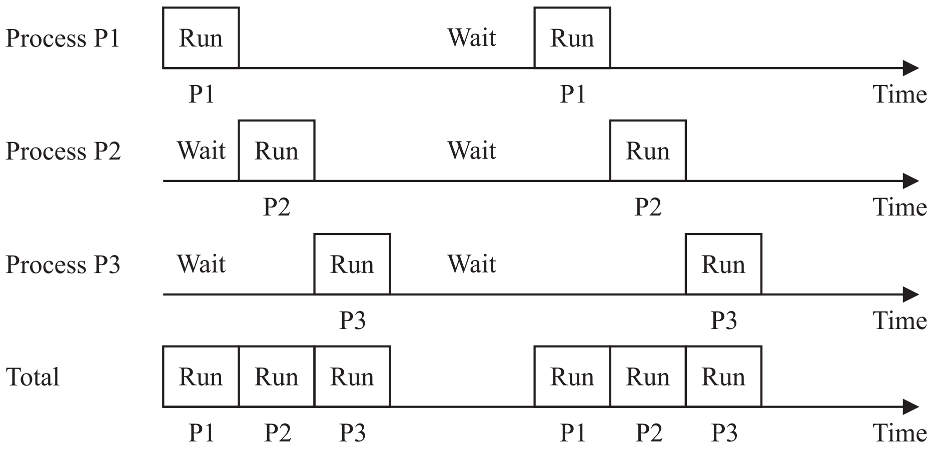


Figure 3.3 : Execution in Uniprogramming System

Assume that we have sufficiently large amount of memory and we can load two programs i.e. processes in the user part of memory without overlapping each other. Let us assume initially process P1 starts executing and after some time it wants to use some I/O device like printer. While P1 is busy with the printer, monitor program switches to the process P2 and continue to execute the same until it needs I/O or completes. If process P2 starts using I/O, then processor resumes the process P1. This can continue until one of them completes when the monitor loads third process. This is shown in figure 3.4(a). We can extend the same concept for three or more processes provided sufficient memory is available. Figure 3.4(b) shows the execution of three processes. This concept of running two or more processes at a time is called multiprogramming.



(a) Two Processes



(b) Three Processes

Figure 3.4 : Execution in Multiprogramming

The multiprogramming operating system throws new challenges. First challenge is the order in which processes are scheduling which is called scheduling policy. It may be purely first come first serve or based on other criteria like priority or the characteristics of the process itself. Second challenge is protecting one process from another for any accidental changes which needs sophisticated memory management and security mechanism. Third challenge is to avoid the conflict for using I/O devices. For example, while one process is using printer and another process demands the same as there is only one printer. Hence, the multiprogramming operating systems are on one side very sophisticated but on other side very complex. The modern operating systems follow the concept of multiprogramming in very efficient manner. Hence, we can say that modern operating systems are one of the examples of most complex software ever designed which is very sophisticated, dynamic and efficient.

Time Sharing

The batch processing with multiprogramming improves the processor utilization but users can not directly interact with their programs while programs are under execution. Modern software like transaction processing systems need to interact with program while executing. This led to development of the concept of time-sharing systems. That means we can use multiprogramming with time sharing for handling multiple interactive jobs.

The time-sharing multiprogramming system divide the processor time among the multiple processes. Each process runs for the fixed time slot called quantum in turn. Multiple users can interact with the system through terminals. For example, if there are N processes running simultaneously in interleaving manner with each process runs for 1/N time in time sharing manner. This allows N users simultaneously to interact with the system just like a dedicated personal computer as like desktop computers. MIT (Massachusetts Institute of Technology) developed first time sharing system called CTSS (Compatible Time-Sharing System) in 1961 for the IBM 709.

The primary objective of the batch processing multiprogramming systems is to improve the processor utilization whereas time sharing multiprogramming systems improve the response time in interactive systems. In batch processing the jobs are put into the bunch and controlled by the JBL (Job Controlled Language) while time sharing systems are controlled by providing commands at the terminal directly by user. The most popular multitasking (handling multiple programs at a time) provides the interactive environment using Shell for interactive users at the same time it also supports batch processing for background or noninteractive processes. Thus, combining the benefit of both processor utilization and response time with interactive environment.

Types of Operating Systems

Previous section discussed the evolution of the operating systems. Although basic concepts remain same, hardware of the computer systems decide what types of variation or changes are needed to customize the operating system for efficiently working on that hardware. The hardware of the computer systems varies depending on the purpose and use of the system. Apart from the hardware, the specific environment including the user needs also determines the types of the operating system. Considering all these facts we can broadly categorize the operating systems into following three categories.

- Desktop operating systems
- Server operating systems
- Mobile operating systems

Desktop Operating Systems

A desktop operating system is designed for personal computer environments mainly desktops and laptops. CP/M (Control Program for Microcomputers) was the first desktop operating system widely used in late 1970s and early 1980s. It was developed by Gary Kildall at Digital Research Inc. in 1974. It was targeted for systems based on 8-bit processor Intel 8080 or Ziglog Z80. The main storage system was floppy disk and interface was text-based.

MS-DOS (Microsoft Disk Operating System) was the another most popular desktop operating system used on IBM-PC machines which was developed in 1981. The MS-DOS was text-based single tasking operating system which can be used by only one user at a time. It was providing support for file system for storing the various types of files used by the different applications. IBM-PC with DOS operating system was widely used for various applications for personal use as well as in office environments for word processing, spreadsheets and small databases. People were also using it for developing small and customized applications using programming languages like BASIC, C, Pascal, COBOL, FORTRAN, etc. In 1984, Apple introduced the Macintosh System Software which popularized the GUI (Graphical User Interface). In 1985, Microsoft released the Windows 1.0 with graphical user interface which worked as foundation for the modern Windows operating systems. Linux emerged as open-source alternative in 1990s providing powerful and customizable desktop environments.

Windows 11, macOS Ventura and Ubuntu Desktop are the latest examples of the desktop operating systems.

Following are the technical features of the desktop operating systems.

- Provides GUI interface with windows, icons, menus and mouse support
- Supports advance file systems with user permissions and encryption
- Multitasking support allowing multiple programs to run at a time and multiple users having their own login accounts
- Device driver management with automatic detection and hardware management with plug-and-play and updates
- Supports network protocols for connecting to Internet
- Support various peripheral devices like printers, scanners, webcams, audio/video devices etc.

- Additionally, provides powerful security feature, software installation and updates, system configuration and power management

Server Operating Systems

A server operating system is mainly designed to provide the services to the multiple clients over the networks with resource sharing among the clients. Examples of services are database services, file sharing, web hosting, authentication, virtualization and many more. The server operating systems provide highly reliable, scalable and secured multiuser support.

The server operating system started with use of the mainframe systems like UNIVAC and IBM's os/360. The multiuser multitasking operating system UNIX developed by the Bell lab in 1970s laid the foundation of the modern server operating systems. In 1980s UNIX variant like BSD became very popular in networked computing. Microsoft came with Windows NT server operating system in market in early 1990s with GUI based administration. Linux became the strong open-source alternative for server in 1991 with Red Hat and Debian. The late 1990s and early 2000s was time for the services like web hosting, emails and database which created demand for more robust server operating systems which resulted into following developments.

- Windows Server 2000/2003 with support of Active Directory and better integration with enterprise systems.
- Enterprise UNIX systems like Solaris, HP-UX and AIX for mission-critical workloads.
- Virtualization and cloud computing like VMware, AWS (Amazon Web Services), Microsoft Azure, Google clouds etc. further contributed in reshaping the server operating systems.
- Modern server operating systems like RHEL (Red Hat Enterprise Linux), Ubuntu Server and Windows Server 2022 are very secured, scalable and cloud-ready.

Following are the technical features of the server operating systems.

- Supports concurrent access by multiple users or clients over network
- Provides enterprise level file and storage management using advanced file systems
- Provides sophisticated backup and recovery in case of failures
- Advanced security features including user authentication, role-based access, encryption and firewall
- Resource monitoring and management for tracking the usage of CPU, memory, disk and network
- Scalability for more hardware and more servers
- Provides command-line and remote administration
- Background service for web, database access, DNS and mail service
- High performance and stability with 24x7 running with high workloads and less crashes
- Maintains system logs for monitoring and auditing
- Support for network management, cloud management, virtualization, package and update management, cluster and load balancing

Mobile Operating Systems

A mobile operating system is designed for smartphones and tablets. It is mainly designed for supporting low power requirement, touchscreen interface, mobile connectivity and sensor integration. Applications are running in sandbox environment to protect the resources and user privacy.

Palm OS developed in 1996 and Symbian in late 1990s were used in PDAs (Personal Digital Assistant) and early smartphones. Major change occurred in mobile operating systems with launch of iOS for iPhone by Apple in 2007. In 2008, Google launched the Android operating system which is most widely used today. It is open-source and supporting wide range of hardware. Both of these operating systems are continuously updated to match with the hardware updates and changing demand. iOS 18.5 and Android 16 are the latest versions of the mobile operating systems.

Following are the technical features of the mobile operating systems.

- Touchscreen interface with gesture making it very user friendly
- Power efficient with sleep mode and settings
- Built-in connectivity for Wi-Fi, Bluetooth, Mobile networks, GPS etc.
- Multimedia support with audio/video, streaming and smart accessories
- Multitasking support with multiple applications running at a time and switching between them
- Built-in voice assistance with Siri on iOS and Google Assistant on Android
- Accessibility features including speech-to-text, voice control, screen magnifiers etc.
- Applications are running in sandbox model preventing application to interfere with other apps or system
- Security and privacy with app permissions, biometric, encryption and sandboxing
- Notification system, Support for apps through store, sensor integration, device compatibility, over-the-air (OTA) updates

Functions of Operating system

An operating system performs many functions to keep computer running smoothly and providing the proper environment for running the user applications. This section discusses the most important and common functions of the operating systems.

User Interface

User interface provides the way to users to interact with computer systems. It is either CLI (Command Line Interface) or GUI (Graphical User Interface). Users can issue the commands or instruct the operating system for executing the task or performing operations like deleting a file, copying a file etc. Windows desktop screen or Linux terminal/DOS prompt are the examples of interface. For example, issuing '*ls -1*' command on Linux terminal or clicking an icon on Windows desktop are example of interaction. We have already learn that Shell component of the operating system is responsible for providing the user interface.

Process Management

Modern operating systems are multitasking means they allow multiple programs to run simultaneously. Program under execution is called a process. Multiple programs running means operating system needs to manage multiple processes and ensure that all of them gets processor time to execute in fair and efficient manner. Operating system should quickly switch from one process to another as user switch from one application to another. Operating system needs to manage all the details for the running processes using very efficient data structures. In addition, process management manages the life cycles of each process with current status of each process from start to end. The various status includes

- **New:** Process is prepared to load in memory
- **Ready:** waiting in queue but ready to execute if processor time is given
- **Running:** currently executing on processor
- **Blocked:** not able to proceed as waiting for I/O device which is not available
- **Exit:** Completed the execution.

Process management uses very sophisticated and efficient algorithms to manage the process life cycle for interactive users.

Process Scheduling

Multitasking system also shares the processor time with all the ready processes. The scheduling is the policy for assigning processor for execution to one of the ready process. The scheduler selects one of the process from the ready queue and assign it to the processor whenever processor is free. Modern operating systems use time sharing policy known as Round Robin. The processor time is divided into small equal size quantum **q** and a process when scheduled runs for maximum **q** time. If it does not finish in that time, it leaves the processor and next process is assigned to processor by scheduler. The outgoing process joins the ready queue and waits for its next turn during which it resumes from the point where it left. This ensures that all the processes are given fair chance and, in each turn, they proceed further and eventually completes. Practically, the scheduling is much more complex phenomena than described as it involve many parameters including process priority, characteristics of the process, user or system process and many more.

Memory Management

Operating system needs to keep track of total memory available for the user programs and data as well as system needs. Whenever a new program or application started, it provides the memory to the new process from the available memory without disturbing the other processes. It maintains the details of which process is allocated how much memory at which locations and updates the details whenever new process is loaded or removed or additional memory is demanded by any process. Modern operating systems are using very sophisticated virtual memory management techniques to allow many and large processes to run with limited memory.

File Management

Dealing with files and folders is very common when we use the computers. We perform various operations like create, open, close, delete, move, copy on files and folders. File management is responsible for all these operations. We are using Windows Explorer to manage our files and folders on Windows operating systems. You will find very similar hierarchical directory structure in Ubuntu Linux also. These are examples of the file management. File management uses very sophisticated file systems to manage files and folders on the storage media like disk and pen drives. Windows file systems FAT32, NTFS, Linux file system ext4 and macOS file system APFS (Apple File System) are examples of latest file systems.

Device Management

I/O devices are integral part of the computer systems which include keyboard, mouse, hard disks, DVDs, monitors, printers, scanners etc. These devices are shared among the processes running on the system. Device management plays key role in controlling these devices and communicating with them. It is likely that systems have only one printer and it is demanded by two processes which are active in system. How to resolve this issue so that both the processes can print their content one by one without mixing? It is managed by device management together with process management in efficient manner.

Security and Access Control

Multiuser multitasking operating systems provide various level of protection and security. The user can access the systems using their login accounts only. File and directory access is governed by file and directory permissions. The resident portion of the operating system and one process from another process are protected by protection rules and privilege levels. Device access is also strictly governed by protection rules. Hence, at every level protection and security mechanism ensures the integrity of the system and protects from any unauthorized access and accidental changes.

Summary

The chapter begins by revisiting the fundamental components of computer hardware - CPU, memory, storage, and I/O devices - which together enable a functional computer system. However, interacting directly with hardware is complex, making operating systems essential. The operating system acts as a bridge between hardware and users, offering a user-friendly environment for running applications efficiently. The operating system functions as a resource manager managing the CPU, memory, storage, and I/O devices to ensure smooth application execution. It has two core components: the shell, which provides user interaction through command-line or graphical user interface, and the kernel, which manages hardware and system resources.

The chapter then explores the evolution of operating systems from manual serial processing to automated batch systems, multiprogramming for better CPU use, and finally to time-sharing systems that allow simultaneous multi-user interaction. Modern operating systems are categorized based on their hardware environments: desktop, server and mobile operating systems. The chapter also covers the major functions of an operating system such as process management, scheduling, memory management, file management, device management and security and access control with relevant examples.

EXERCISE

1. What is an operating system? List the purpose of the operating system.
2. Describe the layered architecture of the computer system and give the role of the operating system.
3. Why do we say operating system is a resource manager?
4. What is Shell? Give its purpose and use.
5. What is kernel? What are its important functions?
6. What is multiprogramming? What are its benefits?
7. How does time-sharing operating system work?
8. Compare the desktop and server operating systems.
9. List the technical features of the server operating system.
10. List major functions of the operating system.

11. State whether true or false.

(1) Shell is the interface of the operating system.

(2) Touch screen is the interface of mobile operating systems.

(3) Kernel of the operating system directly communicates to the hardware.

(4) Time-sharing systems improve the response time.

(5) MS-DOS is mobile operating system.

12. Fill-in the blanks.

(1) __________ operating system is used for sharing resource over network.

(2) Connectivity is the main feature of __________ operating systems.

(3) DOS stands for _______ _______ _______

(4) __________ assigns the processor to the next process.

(5) __________ and __________ popular mobile operating systems.

13. Multi choice questions. Choose the most correct answer.

(1) What is the major purpose of the operating system?

(a) Providing user interface (b) Acting as resource manager

(c) Both A and B (d) Keeping computer busy all the time

(2) Which of the following OS provides only text-based interface?

(a) MS-DOS (b) Ubuntu Desktop

(c) Windows 11 (d) Linux

(3) Which of the following operating system is designed for providing services to multiple clients?

(a) Mobile OS (b) Server OS (c) Desktop OS (d) Android OS

(4) The main feature of the mobile operating system is

(a) Services to multiple clients (b) Personal use for executing programs

(c) Communication (d) Scalability

(5) Which of the following is not the function of kernel?

(a) Allocating memory to process (b) Communication with I/O devices

(c) Process scheduling (d) User Interface

(6) Which of the file system is used on Apple machines?

(a) APFS (b) ext4 (c) NTFS (d) FAT32

(7) File access is part of .... function.

(a) Process management (b) Memory management

(c) File management (d) Security and access control

(8) …......….. operating system provides clustering and load balancing.

(a) Mobile (b) Desktop (c) Server (d) MS-DOS

(9) Purpose of batch processing is to...

(a) Improve the processor utilization (b) Improve the response time

(c) Both A and B (d) Faster exaction

(10) Which of the following concept divide the processor time among the processes?

(a) Interactive workload (b) Multiprogramming

(c) Time sharing (d) Process management