COMPUTER SCIENCE & INFORMATION TECHNOLOGY

Operating System

Comprehensive Theory
with Solved Examples *and* Practice Questions

MADE EASY Publications

www.madeeasypublications.org

MADE EASY Publications Pvt. Ltd.

Corporate Office: 44-A/4, Kalu Sarai (Near Hauz Khas Metro Station), New Delhi-110016 | **Ph. :** 9021300500

Email : infomep@madeeasy.in | **Web :** www.madeeasypublications.org

Operating System

EDITIONS

First Edition : 2015
Second Edition : 2016
Third Edition : 2017
Fourth Edition : 2018
Fifth Edition : 2019
Sixth Edition : 2020
Seventh Edition : 2021
Eighth Edition : 2022

Ninth Edition : 2023

CONTENTS

Operating System

CHAPTER 1

Basic Concepts of Operating System 3-16

CHAPTER 2

Processes and Threads 17-36

CHAPTER 3

CPU Scheduling 37-71

CHAPTER 4

Process Synchronization 72-121

CHAPTER 5

Concurrency and Deadlock 122-159

CHAPTER 6

Memory Management 160-205

CHAPTER 7

Virtual Memory 206-233

CHAPTER 8

File System 234-256

CHAPTER 9

Input/Output System 257-273

Operating System

Goal of the Subject

The goal of this book is to provide all important concepts of operating systems such as Processes and Threads, Mutual Exclusion, CPU Scheduling, Deadlock, Memory Management, Virtual Memory and File Systems.

- Understand the purpose of the operating system
- Distinguish between a resource, a program, and a process
- Solutions of semaphores
- Describe various memory page replacement algorithms
- Describe how files are stored in secondary storage

Operating System

INTRODUCTION

An operating system is a program that acts as an intermediary between a user of a computer and the computer hardware. Two primary aims of an operating systems are to manage resources (e.g. CPU time, memory) and to control users and software. The book consists of topic wise *Examples* and *Student Assignment* questions which test the important concepts from the lesson and provide practice problems. This subject includes the following chapters.

1. **Basic Concepts of Operating System:** In this chapter we discuss Basic Concepts of Operating System, Types of Operating System, Dual Mode Operations and System Call.
2. **Processes and Threads:** In this chapter we discuss Process, Operations on a Process, Scheduling, Thread and Co-operating Processes (Inter Process Communication).
3. **CPU Scheduling:** In this chapter we discuss Goals of CPU Scheduling and Scheduling Algo.
4. **Process Synchronization:** In this chapter we discuss Synchronization, Critical-Section Problem, Synchronization Techniques, Semaphores and Classical Problems of Synchronization with Semaphore Solution.
5. **Deadlock and Concurrency:** In this chapter we discuss Concurrency, Deadlock, Conditions for a deadlock, Methods of Handling Deadlocks, Prevention, Avoidance and Detection & Recovery.
6. **Memory Management:** In this chapter we discuss Logical (Virtual) Vs Physical Address Space, Memory Allocation Techniques, Internal Fragmentation and External Fragmentation, Paging, Segmentation, Segmented Paging and Buddy System.
7. **Virtual Memory:** In this chapter we discuss Page Fault, Page Replacement, Dynamic Paging Algorithms and Frame Allocation.
8. **File System:** In this chapter we discuss Directories, File Management System, File Allocation Methods and Free Space Management.
9. **Input/Output System:** In this chapter we discuss Input/Output System Structure, Magnetic Storage Devices, Disk Scheduling and Disk Scheduling Algorithms.

■■■■

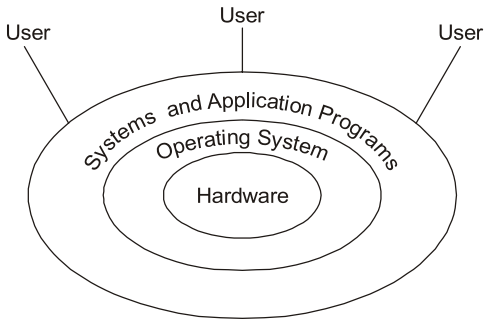
CHAPTER

1

Basic Concepts of Operating System

1.1 Operating System (OS)

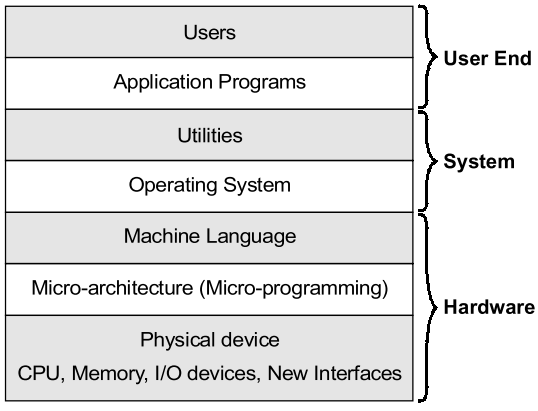
It is a program that acts as an interface between a user (applications) and the computer hardware.



1.2 Structure of Computer System

A computer system consists of:

- **Users:** People, other computers, machines, etc.
- **Application programs:** Compilers, database systems, video games, business applications, web-browsers, etc.
- **System programs:** Shells, editors, compilers, development tools, etc.
- **Operating System:** It is a system program which controls and coordinates the use of hardware among application programs.
- **Hardware:** CPU, Disk, Memory, I/O devices, etc.



NOTE: Firmware (BIOS) is a software which is permanently stored on chip but upgradable. It loads the operating system during the boot.

1.3 Layered View of Operating System Services

1. **User View:** The OS is an interface, hides the details which must be performed and presents a virtual machine to the user that makes easier to use.
   OS provides the following services to the user.
   (*i*) Execution of a program (*ii*) Access to I/O devices
   (*iii*) Controlled access to files (*iv*) Error detection (Hardware failures, and software errors)
2. **Hardware View:** The operating system manages the resources efficiently in order to offer the services to the user programs.
   OS acts as resources managers:
   (*i*) Allocation of resources (*ii*) Controlling the execution of a program
   (*iii*) Controls the operations of I/O devices (*iv*) Protection of resources
   (*v*) Monitors the data
3. **System View:** OS is a program that functions in the same way as other programs. It is a set of instructions that are executed by the processor.
   OS acts as a program to perform the following:
   (*i*) Hardware upgrades (*ii*) New services
   (*iii*) Fixes the issues of resources (*iv*) Controls the user and hardware operations

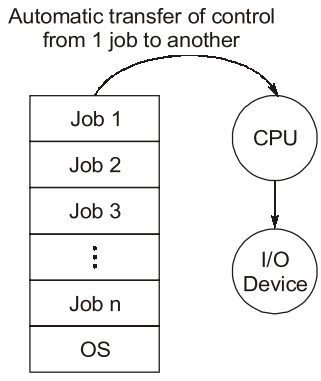
Goals of operating system: Primary goal is convenience and secondary goal is efficiency.

1.4 Types of Operating System

1. Serial OS
2. Batch OS
3. Interactive OS
4. Multiprogrammed OS
5. Time sharing OS
6. Real time OS
7. Network OS
8. Parallel OS
9. Distributed OS
10. Clustered OS
11. Handheld OS

1.4.1 Simple Batch Systems

This type of operating system does not interact with computer directly. There is an operator which takes similar jobs having same requirement and group them into batches. It is the responsibility of operator to sort the jobs with similar needs.



Batch System

Advantages of Batch OS:
(*i*) It is appropriate for executing large jobs that need little interaction.
(*ii*) Multiple users can share the batch systems.

Disadvantages of Batch OS:
(*i*) The computer operators should be well known with batch systems.
(*ii*) Batch systems are hard to debug.
(*iii*) The other jobs will have to wait for an unknown time if any jobs fails.
(*iv*) Throughput is less for batch OS.

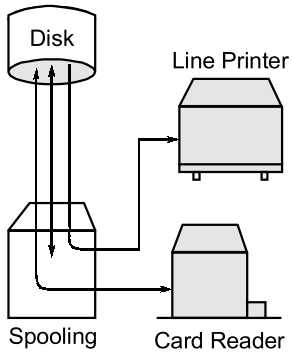
Example of batch OS: IBM OS/2

NOTE: Throughput is the number of jobs completed per unit time (per second).

Spooling

Spool is acronym for simultaneous peripheral operation On-Line. Spooling overlaps I/O of one job with the computation of other job. Spooling uses the disk as a huge buffer for reading as far ahead as possible on input devices and for storing output files until the output devices are able to accept them. A spool is a buffer that holds output for a device, such as a printer, that cannot accept interleaved data streams.

Although a printer can serve only one job at a time, several applications may wish to print their output concurrently, without having their output mixed together. The operating system solves this problem by intercepting as output to the printer. Each applications' output is spooled to a separate disk file.

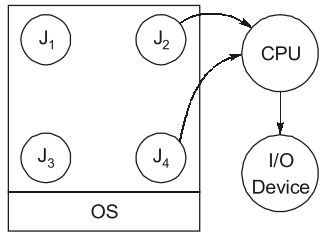


When an application finishes printing, the spooling system queue picks the next spooled-file for input to the printer.

The spooling system copies the queued spool files to the printer one at a time. Even in a simple system, the spooler may be reading the input of one job while printing the output of a different job. Spooling can keep both the CPU and the input/output devices working at much higher rates.

Difference between Spooling and Buffering: The key difference between spooling and buffering is that spooling overlaps the input/output of one job with the execution of another job while buffering overlaps the input/output of one job with the execution of same job. Buffering uses limited area in main memory whereas spooling uses the disk as a huge buffer.

1.4.2 Multi-Programmed Systems



Multi-programming System

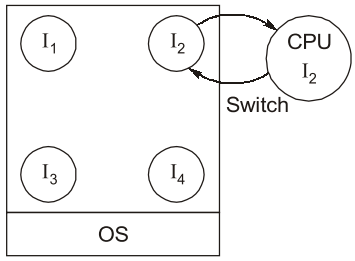
- Several jobs are kept in main memory at the same time and the CPU is multiplexed among them, to increase CPU utilization.
- A job pool on the disk consists of a number of jobs that are ready to be executed. Subsets of these jobs reside in the memory for execution.
- The operating system picks and executes one of the jobs in memory.
- When this job in execution needs an input/output operation to complete, Instead of waiting for the job to complete the input/output, it switches to the subset of jobs waiting for CPU.
- In a non multi programmed system the CPU would sit idle. In multiprogramming system the operating system, simply switches to and executes another job.
- If several jobs are ready to be brought into memory and there is not enough room for all of them, then the system must choose among then. Making this decision is job scheduling.
  Example: Windows and Unix.
- Increased throughput of the system.

1.4.3 Time Sharing System (Multitasking)

- Time sharing or multitasking is a logical extension of multiprogramming. Multiple jobs are executed by the CPU switching between them, but the switches occur so frequently that the users may interact with each program while it is running.
- An interactive computer system provides direct communication between the user and the system. The user gives instructions to the operating system or to a program directly and receives an immediate response.

- A time shared operating system uses CPU scheduling and multiprogramming to provide each user with a small portion of a time shared computer.
- A time shared operating system allows the many users to share the computer simultaneously. Since each action or command in a time shared system tends to be short, only a little CPU time is needed for each user. As the system switches rapidly from one user to the next, each user is given the impression that she has her own computer, whereas actually one computer is being shared among many users.

  Example: Windows, Linux and Unix.



Multi-tasking System

Difference between multiprogramming and multitasking:

Multiprogramming is categorised into two types:

(*i*) Non pre-emptive

(*ii*) Pre-emptive (or time-sharing or multitasking)

- All multitasking are indirectly multiprogramming.
- But only pre-emptive multiprogramming are multitasking.

1.4.4 Real Time System

Used when there are rigid time requirements on the operating of a processor or the flow of data. Systems that control scientific experiments, medical imaging systems, industrial control systems, and some display systems are real-time systems. A real time operating system has well defined, fixed time constraints. Processing must be done within the defined constraints or the system will fail. ***Example***: RTOS

Types of Real Time Systems

1. **Hard real time system:** Guarantees that critical tasks completed on time. This goal requires that all delays in the system be bounded from the retrieval of stored data to the time that it takes the operating system to finish any request made to it. Kernel delays need to be bounded and more restrictive. ***Example:*** Satellite, Missile System.
2. **Soft real time system:** A less restrictive type of real time system is a soft real time system, where a critical real time task gets priority over other tasks, and retains that priority until it completes. Soft real time is an achievable goal that can be mixed with other types of systems. However they have more limited utility than hard real time system. Given their late of deadline support they are risky to use for industrial control and robotics. ***Example:*** Banking system

1.4.5 Parallel Systems (Multiprocessors)

- Multiprocessor system have more than one processor in close communication, sharing the computer bus, the clock, and sometimes memory and peripheral devices. These systems are referred to as tightly coupled systems.
- One advantage of building this kind of system is increased throughput. By increasing the number of processors we hope to get more work done in a shorter period of time.
- Multiprocessors can also save money because the processors can share peripherals, cabinets and power supplies. If several programs are to operate on the same set of data, it is cheaper to store those data on one disk and to have all the processors share them, rather than to have many computers with local disks and many copies of the data.
- Another advantage is increased reliability.
- If functions can be distributed properly among several processors, then the failure of one processor will not halt the system, but rather will only slow it down.

- The ability to continue providing service proportional to the level of surviving hardware is called graceful degradation. Systems that are designed for graceful degradation also called fault-tolerant.
  Example: Unix

Symmetric Multiprocessing Model

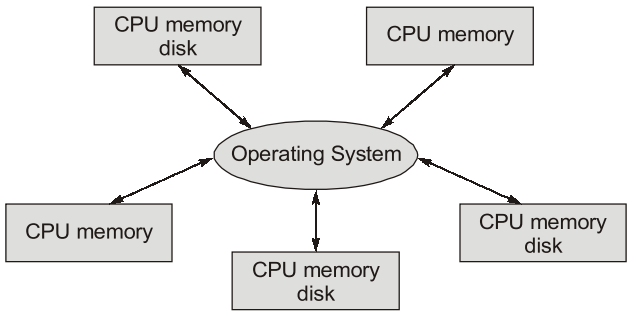
In this each processor runs an identical copy of the operating system and these copies communicate with one another as needed.

Asymmetric Multiprocessing Model

In this model each processor is assigned a specific task. A master processor controls the system; the other processors either look to master for instruction or have predefined tasks. This scheme defines a master-slave relationship. The master processor schedules and allocates work to the slave processors.

1.4.6 Distributed Systems

Various autonomous interconnected computers communicate with each other using a shared communication network. Independent system passes their own memory unit and CPU. There are referred as loosely coupled systems or distributes systems.



Example: LOCUS

Advantages of Distributed Systems

(*i*) **Resource sharing:** If a number of different sites are connected to one another then, a user at one site may be able to use the resources available at another.

(*ii*) **Computation speedup:** If a particular computation can be partitioned into a number of subcomputations that can run concurrently then a distributed system may allow us to distribute the computation among the various sites to run that computation concurrently.

(*iii*) **Reliability:** If one site fails in a distributed system, the remaining sites can potentially continue operating.

(*iv*) **Communication:** When many sites are connected to one another by a communication network, the processes at different sites have the opportunity to exchange information.

(*v*) **Scability:** Scalable as many systems can be easily added to the network.

(*vi*) Load on host computer reduces.

(*vii*) Delay in data processing reduces.

Diadvantages of Distributed Systems

(*i*) Failure of the main network will stop entire communication

(*ii*) These systems are expensive.

1.4.7 Difference between Distributed OS and Network OS

Network Operating System	Distributed Operating System
1. Each computer has its own OS. 2. It allows interaction between the machines by having a common communication architecture. 3. Independent machines accessed by the user.	1. Common OS shared by a network of computers. 2. Single OS controlling the network. 3. Dependent machines accessed by the user to share the resources.

1.5 Dual Mode Operations

A processor can support two modes of execution:

1. Kernel / Protected / Supervisor / System/Monitor / Privileged mode.
2. User mode / Non-privileged mode

Operating system runs in Kernel mode and user programs run in user mode. Mode bit is used to decide the mode of operating system. If mode bit is 0, it operates in kernel mode. Otherwise it operates in user mode when mode bit is 1.

User mode (Non-privileged mode)	Mode bit = 1
Kernel mode (Privileged mode)	Mode bit = 0

1.5.1 Kernel Mode

A process running in Kernel mode has following:

- Full access to machine instruction set.
- Direct access to hardware (memory, I/O devices, etc).

OS and device drivers must run in Kernel mode. MS DOS only support Kernel mode.

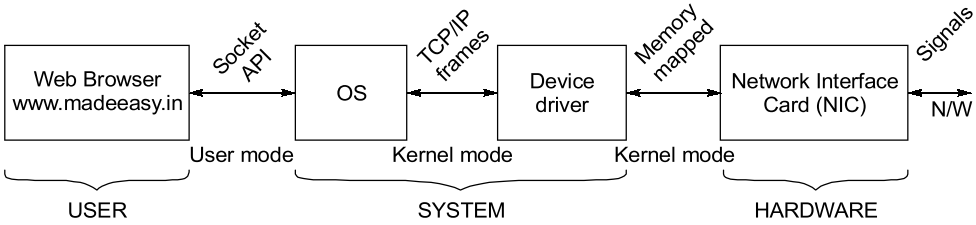
Privileged instructions: Set timer, Clear memory, Disable interrupts, Access I/O devices, etc.

1.5.2 User Mode

A process running in user mode has following:

- Access to the limited set of machine instructions.
- No direct access to hardware.
- Hardware access is coordinated by OS.

Non-privileged instructions: Read clock, Generate trap, User to Kernel mode switch, etc.



1.5.3 Privileged Instructions and Non-privileged Instructions

Privileged Instructions	Non-Privileged Instructions
I/O operations Context switching Disabling the interrupts Set system lock Remove process from memory Changing memory LOC of process	Reading system time Reading the states of CPU Sending the final printout of printer

Note: In boot time the system always start in kernel. The OS always run in kernel mode.

1.6 Functions of Operating System

- It controls all of computer resources.
- It provides valuable services to user programs.
- It coordinates the execution of user programs.
- It provides resources to user programs.
- It provides an interface (virtual machine) to the user.
- It hides the complexity of software.
- It supports the multiple execution modes.
- It monitors the execution of user programs to prevent errors.

1.7 Operating System Components

- **Process Management:** Operating system manages the process creation and deletion.
- **Main Memory Management:** Operating system keeps track of allocation and deallocation of main memory.
- **Secondary Storage Management:** Operating system uses secondary storage (Disk) to extend main memory.
- **I/O System Management:** Operating system uses I/O system to manage the devices.
- **File Management:** Operating system manages the file creation, deletion and permissions.
- **Protection System:** Operating system protects the resources by authorization.
- **Interface:** Operating system provides an interface to the user using the facilities by "command interpreter" or "Graphical user interfaces".
- **Networking:** The processors in the system are connected through a communication network. Communication takes place using a protocol. A distributed system provides user access to various system resources.

1.8 System Call

System call provides the services of operating system to the user programs via Application Program Interface (API). It provides an interface between a process and operating system to allow user level processes to request services of operating system. System calls are the only entry points into the Kernel System. All programs needing resources must use system calls.

Services Provided by System Calls

1. Process creation and management
2. Main memory management
3. File Access, Directory and File system management
4. Device handling (I/O)
5. Protection
6. Networking, etc.

Types of System Calls

1. **Process control:** end, abort, create, terminate, allocate and free memory.
2. **File management:** create, open, close, delete, read file etc.
3. Device management
4. Information maintenance
5. Communication

Q.1 In a multiprogramming and time-sharing environment, several users share the system simultaneously. This situation can result in various security problems.
(a) What are two such problems?
(b) Can we ensure the same degree of security in a time-shared machine as we have in a dedicated machine? Explain your answer.

Ans: (a) Stealing or copying one's programs or data; using system resources (CPU, memory, disk space, peripherals) without proper accounting.
(b) Probably not, since any protection scheme devised by humans can inevitably be broken by a human, and the more complex the scheme, the more difficult it is to feel confident of its correct implementation.

Q.2 What is the main advantage of multiprogramming?

Ans: Multiprogramming makes efficient use of the CPU by overlapping the demands for the CPU and its I/O devices from various users. It attempts to increase CPU utilization by always having something for the CPU to execute.

Q.3 List the steps that are necessary to run a program on a completely dedicated machine.

Ans: (a) Reserve machine (b) Manually load program into memory (c) Load starting address and begin execution (d) Monitor and control execution of program form console.

Q.4 Writing an operating system that can operate without interference from malicious or undebugged user programs requires some hardware assistance. Name three hardware aids for writing an operating system.

Ans: (a) Monitor/user mode (b) Privileged instructions (c) Timer (d) Memory protection.

Q.5 What is the purpose of the command interpreter? Why is it usually separate from the kernel?

Ans: It reads commands from the user or from a file of commands and executes them, usually by turning them into one or more system calls. It is usually not part of the Kernel since the command interpreter is subject to changes.

Q.6 What is the main advantage of the layered approach to system design?

Ans: As in all cases of modular design, designing an operating system in a modular way has several advantages. The system is easier to debug and modify because changes affect only limited sections of the system rather than touching all sections of the operating system. Information is kept only where it is needed and is accessible only within a defined and restricted area, so any bugs affecting that data must be limited to a specific module or layer.

Q.7 What are the main advantages of the micro Kernel approach to system design?

Ans: Benefits typically include the following (a) adding a new service does not require modifying the Kernel, (b) it is more secure as more operations are done in user mode than in Kernel mode, and (c) a simpler Kernel design and functionality typically results in a more reliable operating system.

Summary

- **Multitasking:** Interaction between multiple processes on the same processor.
- **Multiprogramming:** Interaction between multiple processes in the system. (either on same processor or on different processor)
- **Multiprocessing:** Interaction between multiple processors to execute the processes parallely.
- **Concurrency includes:**
  - Communication among processes/threads.
  - Sharing system resources.
  - Cooperative processing of shared data.
  - Synchronization of process/thread activities.
  - Organized CPU scheduling.
  - Solving deadlock and starvation problems.
- **Concurrency arises:**
  - Interaction between multiple processes running on one CPU.
  - Interaction between multiple threads running in one process.
  - Interaction between multiple processors running multiple processes/threads.
- **Multicomputing:** Interaction between multiple computers running distributed processes.

Student's Assignment

Q.1 The software that contains the core components of the operating system is called
(a) Controller (b) Root
(c) Kernel (d) None of the above

Q.2 When a computer is switched on, where is the operating system loaded?
(a) BIOS (b) ROM
(c) POST (d) RAM

Q.3 On which chip is the BIOS program permanently stored?
(a) RAM (b) ROM
(c) SIMBA (d) None of these

Q.4 Spooling helps because
(a) it is a more secure method of accessing data
(b) print jobs go more smoothly with less stop and go
(c) the computer is released to do other things while still printing
(d) None of the above

Q.5 When a computer is "swapping", it is
(a) moving data from hard drive to floppy drive
(b) moving data from memory to the swap file on the hard drive
(c) moving data between registers in memory
(d) None of the above

Q.6 The Operating System is responsible for
(a) controlling peripheral devices such as monitor, printers, disk drives
(b) provide an interface that allows users to choose programs to run/manipulate files
(c) manage users' files on disk
(d) All of the above

Q.7 Which of the following does an operating system do in a stand-alone computer system?
(a) manages the user's files
(b) provides the interface to allow the user to communicate with the computer
(c) controls the various peripherals
(d) All of the above

Q.8 Which of the following is true about a terminal on a time-sharing computer system?

(a) has its own CPU and some memory
(b) has no memory or CPU of its own
(c) has its own CPU but no memory
(d) has its own memory but no CPU

Q.9 An interrupt handler is a
(a) location in memory that keeps track of recently generated interrupts
(b) peripheral device
(c) utility program
(d) special numeric code that indicates the priority of a request

Q.10 Loading the OS into the memory of a PC is called
(a) thrashing (b) booting
(c) formatting (d) spooling

Q.11 An operating system is
(a) application (b) supervisory program
(c) set of users (d) set of programs

Q.12 This part of the operating system manages the essential peripherals, such as the keyboard, screen, disk drives, and parallel and serial ports.
(a) basic input/output system
(b) secondary input/output system
(c) peripheral input/output system
(d) marginal input/output system

Q.13 Executing more than one program concurrently by one user on one computer is known as
(a) multiprogramming
(b) time-sharing
(c) multitasking
(d) multiprocessing

Q.14 The simultaneous processing of two or more programs by multiple processors is
(a) multitasking (b) multiprogramming
(c) time-sharing (d) multiprocessing

Q.15 This comprises the detailed machine language necessary to control a specific device and is controlled by the operating system.
(a) driver
(b) utility program
(c) virtual memory
(d) peripheral device

Q.16 Which of the following controls the manner of interaction between the user and the operating system?
(a) language translator
(b) platform
(c) user interface
(d) none of the above is correct.

Q.17 Overlay is
(a) a part of an OS
(b) a specific memory location
(c) a single contiguous memory that was used in the olden days for running large programs by swapping
(d) overloading the system with many user fills

Q.18 Which of the following is real time system?
(a) an on line railway reservation system
(b) aircraft control system
(c) payroll processing system
(d) All of the above

Q.19 Where does the swap space residable?
(a) RAM (b) disk
(c) ROM d) on-chip cache

Q.20 What is not the major objective of an operating system?
(a) to act as a resource manager for multiple tasks running on the CPU, the memory and disk resources
(b) to provide a programming interface to the user
(c) to act as an uniform abstract machine on top of a variety of different hardware platforms
(d) to enable loading and execution of binary code with minimum intervention by the user

Q.21 System call is used to access
(a) I/O functionality
(b) Operating system functionality
(c) Application functionality
(d) None of the above

Q.22 Which of the following statement is true with respect to Interrupts?

(a) interrupts are external to the CPU
(b) when an interrupt occurs, the CPU stops executing the current process and immediately transfers the execution to an appropriate Interrupt service routine
(c) when higher priority Interrupt occurs while processing the lower priority interrupt, then CPU transfers the execution to handle the high priority interrupts
(d) All of the above

Q.23 In order to change the mode from privileged to non-privileged:
(a) a hardware interrupt is needed
(b) a software interrupt is needed
(c) a privileged instruction is needed
(d) a non-privileged instruction is needed

Q.24 Let T_m be the time taken to switch between user mode and Kernel mode (mode switching) and T_p be the time taken for context switching which relation is true for T_m and T_p?
(a) $T_m >> T_p$ (b) $T_p = T_m$
(c) $T_m << T_p$ (d) None of the above

Answer Key:

1. (c)	**2.** (d)	**3.** (b)	**4.** (c)	**5.** (b)
6. (d)	**7.** (d)	**8.** (b)	**9.** (c)	**10.** (b)
11. (b)	**12.** (a)	**13.** (c)	**14.** (d)	**15.** (a)
16. (c)	**17.** (c)	**18.** (b)	**19.** (b)	**20.** (d)
21. (b)	**22.** (d)	**23.** (d)	**24.** (c)	

Student's Assignments Explanations

1. (c)
Kernel contains the core component of OS like process management, memory management, network management etc.

2. (d)
When a computer is switched on, the ROM leads the BIOS system and OS is loaded and put into the RAM, computer's main memory.

3. (b)
BIOS program is permanently stored on ROM (Read Only Memory) because it is non-volatile and the OS needs to be on the computer every time its switched on.

4. (c)
Spooling is useful because devices access data at different rate. The spool (buffer) provides a waiting station where data can rest while slower device catches up.

5. (b)
Swapping is a mechanism in which a process can be swapped, (or moved) temporarily out of main memory to secondary storage (or disk) to make that memory available for other processes.

6. (d)
Device management, file management, I/O system management are amongst some of the responsibilities of OS.

8. (b)
Time-sharing is a technique which enables many people, located at various terminals, to use a particular computer system at the same time.

9. (c)
An interrupt handler, also known as Interrupt Service Routine (ISR), is a piece of software or more specifically a callback function in an operating system, whose execution is triggered by the reception of an interrupt.

10. (b)
Boot process loads the OS into main memory (or RAM).

11. (b)
A supervisory program is a computer program, part of an OS, that controls the execution of other routines and regulates I/O operations, task scheduling etc.

12. (a)
I/O management is important part of OS, i.e.., managing keyboard, disk drivers, printers, USB devices etc.