



CS3451 INTRODUCTION TO OPERATING SYSTEMS

&

AL3452 OPERATING SYSTEMS

CS3451 INTRODUCTION TO OPERATING SYSTEMS L T P C**COURSE OBJECTIVES:**

- To understand the basics and functions of operating systems.
- To understand processes and threads
- To analyze scheduling algorithms and process synchronization.
- To understand the concept of deadlocks.
- To analyze various memory management schemes.
- To be familiar with I/O management and file systems.
- To be familiar with the basics of virtual machines and Mobile OS like iOS and Android.

UNIT I INTRODUCTION**7**

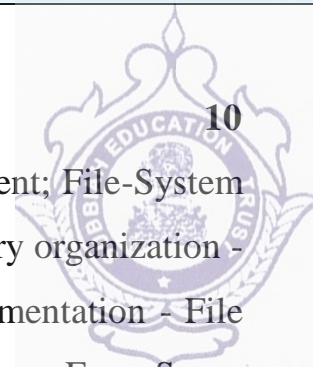
Computer System - Elements and organization; Operating System Overview - Objectives and Functions - Evolution of Operating System; Operating System Structures – Operating System Services - User Operating System Interface - System Calls – System Programs - Design and Implementation - Structuring methods.

UNIT II PROCESS MANAGEMENT**11**

Processes - Process Concept - Process Scheduling - Operations on Processes - Inter-process Communication; CPU Scheduling - Scheduling criteria - Scheduling algorithms: Threads - Multithread Models – Threading issues; Process Synchronization - The Critical-Section problem - Synchronization hardware – Semaphores – Mutex - Classical problems of synchronization - Monitors; Deadlock - Methods for handling deadlocks, Deadlock prevention, Deadlock avoidance, Deadlock detection, Recovery from deadlock.

UNIT III MEMORY MANAGEMENT**10**

Main Memory - Swapping - Contiguous Memory Allocation – Paging - Structure of the Page Table - Segmentation, Segmentation with paging; Virtual Memory - Demand Paging – Copy on Write - Page Replacement - Allocation of Frames –Thrashing.



UNIT IV STORAGE MANAGEMENT

10

Mass Storage system – Disk Structure - Disk Scheduling and Management; File-System Interface - File concept - Access methods - Directory Structure - Directory organization - File system mounting - File Sharing and Protection; File System Implementation - File System Structure - Directory implementation - Allocation Methods - Free Space Management; I/O Systems – I/O Hardware, Application I/O interface, Kernel I/O subsystem.

UNIT V VIRTUAL MACHINES AND MOBILE OS

7

Virtual Machines – History, Benefits and Features, Building Blocks, Types of Virtual Machines and their Implementations, Virtualization and Operating-System Components; Mobile OS - iOS and Android.

TOTAL: 45 PERIODS

COURSE OUTCOMES:

At the end of this course, the students will be able to:

CO1 : Analyze various scheduling algorithms and process synchronization.

CO2 : Explain deadlock prevention and avoidance algorithms.

CO3 : Compare and contrast various memory management schemes.

CO4 : Explain the functionality of file systems, I/O systems, and Virtualization

CO5 : Compare iOS and Android Operating Systems.

TEXT BOOKS:

1. Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, –Operating System Conceptslll, 10th Edition, John Wiley and Sons Inc., 2018.
2. Andrew S Tanenbaum, "Modern Operating Systems", Pearson, 5th Edition, 2022 New Delhi.

REFERENCES:

1. Ramaz Elmasri, A. Gil Carrick, David Levine, — Operating Systems – A Spiral Approachll, Tata McGraw Hill Edition, 2010.



2. William Stallings, "Operating Systems: Internals and Design Principles", 7th Edition, Prentice Hall, 2018.

3. Achyut S. Godbole, Atul Kahate, -Operating Systems, McGraw Hill Education, 2016.

CO's- PO's & PSO's MAPPING

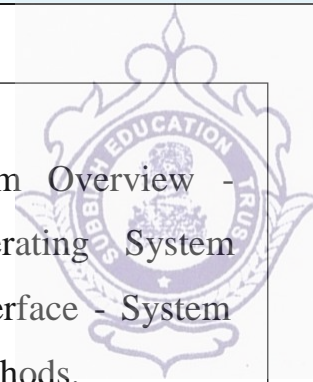
CO's	PO's												PSO's		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
1	2	1	3	2	-	-	-	-	2	1	2	3	2	1	1
2	2	1	1	1	1	-	-	-	1	3	3	3	2	3	3
3	1	3	3	3	1	-	-	-	1	2	1	2	2	1	2
4	1	2	2	3	-	-	-	-	2	3	3	1	3	1	3
5	1	2	3	2	3	-	-	-	3	1	3	3	1	3	3
Avg.	1	2	2	2	2	-	-	-	2	2	2	2	2	2	2

1 - low, 2 - medium, 3 - high, -, -" - no correlation

COURSE OUTCOME

At the end of course, students will have ability to

CO1	Understand the structure and functions of operating system.
CO2	Analyze and Contrast various Memory management schemes.
CO3	Analyze iOS and Android Operating Systems.
CO4	Design various scheduling algorithm ,deadlock Prevention and Avoidance algorithms.
CO5	Create and Implement prototype File systems.
CO6	Understand the types of virtual machines and its implementation.



UNIT I INTRODUCTION 7

Computer System - Elements and organization; Operating System Overview - Objectives and Functions - Evolution of Operating System; Operating System Structures – Operating System Services - User Operating System Interface - System Calls – System Programs - Design and Implementation - Structuring methods.

PART – A

1. List and briefly define the four main elements of a computer?

- Processor – Controls the operation of the computer & performs its data processing functions
- Main memory – Stores data & programs. It is volatile.
- I/O modules – Move data between the computer & its external environment such as disks, communication equipment & terminals.
- System Bus – Provides for communication among processors, main memory & I/O modules.

2. Define the two main categories of processor register?

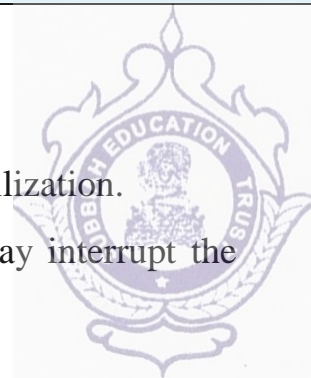
Two categories are

User- visible registers: - It enables the machine or assembly language programmer to minimize main memory references by optimizing register use.

Control & Status registers: - Registers used by the processor to control the operation of the processor.

3. In general terms, what are the four distinct actions that machine instruction can specify?

- Processor – Memory
- Processor – I/O
- Data Processing
- Control



4. What is an Interrupt?

- Interrupt are provided primarily as way to improve processor utilization.
- It is a mechanism by which other modules(I/O, Memory) may interrupt the normal sequencing of the processor.

Classes of interrupts:-

- Program
- Timer
- I/O
- Hardware failure

5. How are multiple interrupt dealt with?

Two approaches can be taken to dealing with interrupts

- Disabled Interrupt – Processor ignores any new interrupt request signal.
- Define Priority for interrupt – It allows an interrupt of higher priority.

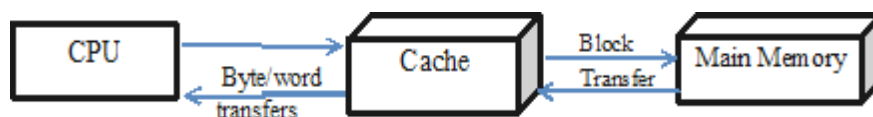
6. What characteristics distinguish the various elements of a memory hierarchy?

Characteristics are

1. Cost Per bit
2. Capacity
3. Access Time
4. Frequency of access to the memory by the processor

7. What is Cache Memory?

1. Cache memory is invisible to the OS
2. It interacts with other memory management hardware
3. Cache contains a copy of a portion of main memory .





8. List and briefly define 3 Techniques of I/O operation?

- Programmed I/O
- Interrupt Driven I/O
- Direct memory access

9. What is the distinction b/w spatial locality & temporal locality?

Temporal locality refers to the reuse of specific data and/or resources within relatively small time durations.

Spatial locality refers to the use of data elements within relatively close storage locations.

Sequential locality, a special case of spatial locality, occurs when data elements are arranged and accessed linearly, e.g., traversing the elements in a one-dimensional array.

10. Define Locality of Reference

Locality of reference, also known as the principle of locality, is the phenomenon of the same value or related storage locations being frequently accessed.

There are two basic types of reference locality.

- Temporal locality refers to the reuse of specific data and/or resources within relatively small time durations.
- Spatial locality refers to the use of data elements within relatively close storage locations.
- Sequential locality, a special case of spatial locality, occurs when data elements are arranged and accessed linearly, e.g., traversing the elements in a one-dimensional array.

11. What is an operating system? (NOV/DEC 2013)

An operating system is a program that manages the computer hardware. It acts as an intermediate between a user's of a computer and the computer hardware. It controls and coordinates the use of the hardware among the various application



programs for the various users.

12. What are the 3 objective of an OS Design?

- Convenience – An OS makes a computer more convenient to use
- Efficiency -- An OS allows the system resources to be used in efficient manner
- Ability to Evolve – An OS Constructed in such a way as to permit the effective development, testing & introducing new function.

13. List the Services of operating system function. (NOV/DEC 2013)

1. Program development
2. Program execution
3. User Interface
4. I/O Operations
5. File system Manipulation
6. Communication
7. Error Detection
8. Resource allocation
9. Accounting
10. Security

14. Define Kernel

The kernel is a software code that resides in the central core of a operating system. It has complete control over the system.

15. Define system call. (MAY/JUNE 2009, APRIL/MAY 2019)

System Call provides the interface between running program and the OS

User can request any services from OS through System Call.

Categories of system call:-

- File management
- Process Management



- Inter process Communication
- I/O Device Management
- Information Processing & Maintenance

16. What is System Programs? (APRIL/MAY 2011)

System programs provides an convenient environment to the user for developing and executing the programs.

Categories:-

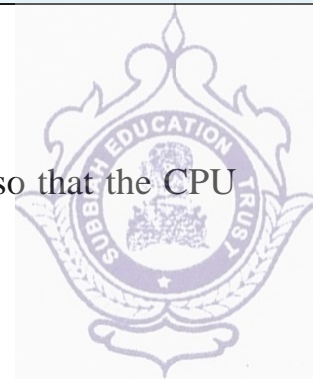
1. File management
2. Status Information
3. File Modification
4. Programming language support
5. Program loading & execution
6. Communication

17. What is Boot strapping?

The boot program is stored on disk with predetermined address called boot sector. The boot program then loads the operating system into memory to startup the computer this arrangement is known as bootstrapping.

18. Difference b/w Monolithic & Microlithic.

Monolithic	Micro lithic
Kernel size is large	Kernel size is small
OS is Complex to design	OS is easy to Design Implement & Install
Request may be serviced faster	Request may be serviced slower
All OS services are included in the Kernel	Kernel Provides only IPC and low level Device management services



19. What is Multiprogramming? (MAY/JUNE 2013)

Multi Programming increases CPU Utilization by organizing jobs so that the CPU always has one to execute.

Advantage:-

- It increase CPU utilization
- It makes efficient use of the CPU overlapping the demands for the CPU & I/O devices
- Increased throughput.
- Lower response time.

20. Define Real Time System

Real time system is one that must react to input & responds to them quickly. A real time system has well defined, fixed time constants.

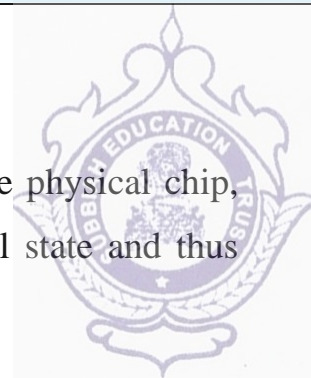
21. What does the CPU do when there are no user programs to run? (NOV/DEC 2011)

The CPU will always do processing. Even though there are no application programs running, the operating system is still running and the CPU will still have to process.

22. Describe the actions taken by a kernel to context-switch between processes.

(NOV/DEC 2013)

In general, the operating system must save the state of the currently running process and restore the state of the process scheduled to be run next. Saving the state of a process typically includes the values of all the CPU registers in addition to memory allocation. Context switches must also perform many architecture-specific operations, including flushing data and instruction caches.



23.What is multicore processor?

Hardware has been to place multiple processor cores on the same physical chip, resulting in a multicore processor. Each core maintains its architectural state and thus appears to the operating system to be a separate physical processor.

24.What is memory stall?

Researchers have discovered that when a processor accesses memory, it spends a significant amount of time waiting for the data to become available. This situation, known as a memory stall, may occur for various reasons, such as a cache miss.

25.What is Boot strapping?

- The boot program is stored on disk with predetermined address called boot sector.
- The boot program then loads the operating system into memory to startup the computer. This arrangement is known as bootstrapping.

26.Can multiple user level threads achieve better performance on a multiprocessor system than a single processor system? Justify your answer. (MAY/JUNE 2014)

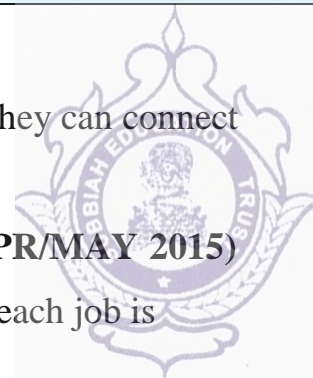
We assume that user-level threads are not known to the kernel. In that case, the answer is because the scheduling is done at the process level. On the other hand, some OS allows user-level threads to be assigned to different kernel-level processes for the purposes of scheduling. In this case the multithreaded solution could be faster

27.Mention the circumstances that would a user be better off using a time-sharing system rather than a PC or a single user workstation? (MAY/JUNE 2014)

A user is better off fewer than three situations: when it is cheaper, faster, or easier.

For example:

1. When the user is paying for management costs and the costs are cheaper for a time-sharing system than for a single-user computer.
2. When running a simulation or calculation that takes too long to run on a single PC or workstation.



3. When a user is travelling and doesn't have laptop to carry around, they can connect remotely to a time-shared system and do their work.

28. Do timesharing differ from Multiprogramming? If so, How?(APR/MAY 2015)

Time Sharing: here, OS assigns some time slots to each job. Here, each job is executed according to the allotted time slots.

Job1: 0 to 5

Job2: 5 to 10

Job3: 10 to 15

Multi-Tasking: in this operating system, jobs are executed in parallel by the operating system. But, we can achieve this multi-tasking through multiple processors (or) multicore CPU only.

CPU1: Job1

CPU2: Job2

CPU3: Job3

29. Why API s need to be used rather than system calls?(APR/MAY 2015)

System calls are much slower than APIs (library calls) since for each system call, a context switch has to occur to load the OS (which then serves the system call). Most details of OS interface hidden from programmer by API Managed by run-time support library (Set of functions built into libraries included with compiler.)

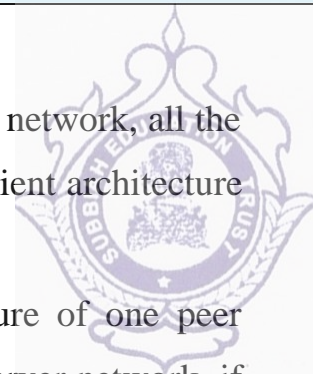
30. Describe the actions taken by a thread library to context switch between user-level threads.

The user-level threads are known only within a given process. To context switch, we only need to save the thread-specific context: the program counter, CPU registers, and the thread-specific stack pointer

31. What is the purpose of system programs? (May/Jun 2016)

System programs can be thought of as bundles of useful system calls. They provide basic functionality to users so that users do not need to write their own programs to solve common problems.

32. What are the advantages of peer-to-peer systems over client-server systems? (May/Jun 2016)



- It is easy to install and so is the configuration of computers on this network, all the resources and contents are shared by all the peers, unlike server-client architecture where Server shares all the contents and resources.
- P2P is more reliable as central dependency is eliminated. Failure of one peer doesn't affect the functioning of other peers. In case of Client –Server network, if server goes down whole network gets affected.
- There is no need for full-time System Administrator. Every user is the administrator of his machine. User can control their shared resources.
- The over-all cost of building and maintaining this type of network is comparatively very less.

33. Compare and contrast DMA and cache memory. (Nov/Dec 2015)

- DMA is a hardware device that can move to/from memory without using CPU instructions.
- For instance, a hardware device (lets say, your PCI sound device) wants audio to play back. You can either:
 - Write a word at a time via a CPU mov instructions.
 - Configure the DMA device. You give it a start address, a destination, and the number of bytes to copy. The transfer now occurs while the CPU does something else instead of spoon feeding the audio device.
- DMA can be very complex (scatter gather, etc), and varies by bus type and system.

34. Write the difference between Batch systems and Time sharing systems. (Nov/Dec 2015)

- A batch is a sequence of jobs. This batch is submitted to batch processing operating systems, and output would appear some later time in the form of a program or as program error. To speed up processing similar jobs are batched together.



- The major task of batch operating systems is to transfer control automatically from one job to next. Here the operating is always in the memory.
- Time sharing or multi-tasking is a logical execution of multiprogramming. Multiple jobs are executed by the CPU switching between them. Here the computer system provides on line communication between the user and the system.
- Here the CPU is never idle. Time shared operating system allows many users to share the computer simultaneously.
- Time sharing systems requires some sort of memory management and protection.

35. How does an interrupt differ from a trap or what is the difference between trap and interrupt? (NOV/DEC 2016) (APR/MAY 2018)

An interrupt handler is called to deal with the cause of the interrupt; control is then returned to the interrupted context and instruction. A trap is a software-generated interrupt. An interrupt can be used to signal the completion of an I/O to obviate the need for device polling.

36. What are the disadvantages of multiprocessor systems? (NOV/DEC 2016)

- If one processor fails then it will affect in the speed
- Multiprocessor systems are expensive.
- Complex OS is required.
- It's more complex.
- It requires context switching which slightly impacts

37. Consider memory systems with a cache access time of 10 ns and a memory access time of 110 ns-assume the memory access time includes the time to check the cache. if the effective access time is 10 % greater than the cache access time, What is the hit ration H? (APR/MAY 2017) [E]



$$\text{Effective Access Time} = H * T_{\text{cache}} + (1-H) * T_{\text{memory}}$$

$$1.1 * T_{\text{cache}} = H * T_{\text{cache}} + (1-H) * T_{\text{memory}}$$

$$1.1 * 10 = H * 10 + (1-H) * 110$$

$$11 = H * 10 + 110 - 110 * H$$

$$-99 = -100 * H$$

$$H = 99/100$$

We deducted 1pt for minor errors, 3 points for the correct formula but incorrect calculations, and 4 pts for the right idea but wrong formula and calculation.

38. What are the objectives of operating systems? (APR/MAY 2017) (NOV/DEC 2017)

An OS is a program that controls the execution of application programs and acts as an interface between applications and the computer hardware. Objectives of OS:
 Convenience: An OS makes a computer more convenient to use. Efficiency: An OS allows the computer system resources to be used in an efficient manner.

39. What is SYS GEN and system boot ? (NOV/DEC 2017) (NOV/DEC 2021)

SYS GEN : It is the Short for SYSTEM GENERation, sysgen is a utility that enables an operating system to configure hardware and software setups properly.

System Boot:

Booting the system is done by loading the kernel into main memory, and starting its execution.

The CPU is given a reset event, and the instruction register is loaded with a predefined memory location, where execution starts.

40. Mention the purpose of system calls. (APR/MAY 2018)

- System calls allow user-level processes to request services of the operating system.
- If a file system requires the creation or deletion of files.
- Reading and writing from files also require a system call.
- Creation and management of new processes.
- Network connections also require system calls. This includes sending and receiving packets.

- Access to a hardware devices such as a printer, scanner etc. requires a system call.

41. What is dual mode operation and what is the need of it? (APRIL/MAY 2019)

In monitor mode, the CPU can use all instructions and access all areas of memory.

In user mode, the CPU is restricted to unprivileged instructions and a specified area of memory. User code should always be executed in user mode and the OS design ensures that it is.

42. Distinguish between Multicore and Multiprocessor. (NOV/DEC 2021)

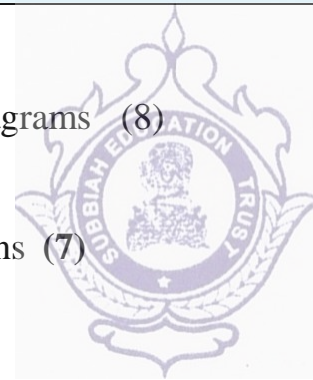
Multicore systems have a single processor with multiple processing units. These processing units are termed cores. On the other hand, **multiprocessor systems** have two or more processors. A multiprocessor system is much more reliable than a multicore system but a multiprocessor system has a complex configuration compared to a multicore system. Both multicore processors and multiprocessors are used to speeding up the computing process of the system.

PART – B & C

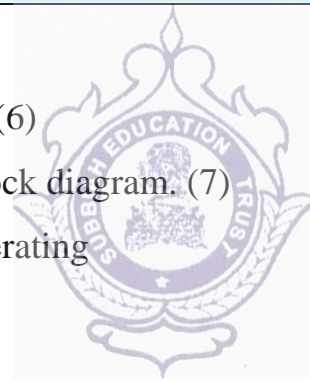
1. Explain Operating System Structure and components. (APRIL/MAY 2010, NOV/DEC 2013)
2. Discuss multiprocessor systems in detail. (MAY/JUNE 2013)
3. Explain in detail the types of system calls provided by a typical operating system. (NOV/DEC 2012)
4. Explain the purpose of system calls and discuss the calls related to device management and communications in brief. (MAY/JUNE 2014)
5. Explain the concepts of virtual machines, their implementation and benefits in details. (MAY/JUNE 2014)
6. What is a virtual machine? List out the advantages of virtualization. Explain the creation of a Virtual machine with a architecture diagram (NOV/DEC 2013)
7. Write short notes on operating system services and components. (MAY/JUNE 2012)
8. Write in detail about the real time system and multiprocessor system.



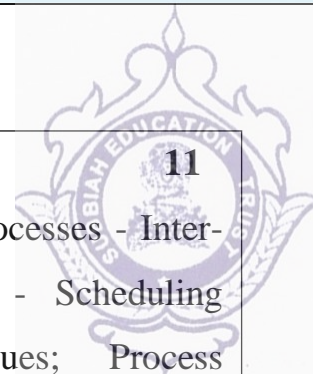
- 9.** Explain the various types of System calls with an example for each?(APR/MAY 2015)
- 10.** Discuss about the functionality of system boot with respect to operating system. (APR/MAY 2015)
- 11.** Discuss about the evolution of virtual machines. Also explain how virtualization could be implemented in operating systems. (APR/MAY 2015)
- 12.** Sketch the structure of Direct Memory Access in detail. (APR/MAY 2015)
- 13.** (i) With neat sketch discuss computer system overview
 (ii) Enumerate the different operating system structure and explain with neat sketch. (Nov/Dec 2015)
- 14.** (i) State the basic functions of OS and DMA.
 (ii) Explain system calls, system programs and OS generation. (Nov/Dec 2015)
- 15.** (i) Distinguish between the client-server and peer-to-peer models of distributed systems.
 (ii) Describe three general methods for passing parameters to the operating system with example. (MAY/JUNE 2016)
- 16.** (i) How could a system be designed to allow a choice of operating systems from which to boot? What would the bootstrap program need to do?
 (ii) Describe the differences between symmetric and asymmetric multiprocessing. What are three advantages and one disadvantage of multiprocessor systems? (MAY/JUNE 2016)
- 17.** (i) Describe a mechanism for enforcing memory protection in order to prevent a program from modifying the memory associated with other programs (8) (NOV/DEC 2016)
 (ii) What are the advantages and disadvantages of using the same system call interface for manipulating both files and devices? (8) (NOV/DEC 2016)
- 18.** (i) State and explain the major activities of an operating system with regard to file management? (8) (NOV/DEC 2016)



- (ii) Discuss the different multiprocessor organizations with block diagrams (8)
(NOV/DEC 2016)
- 19.**(i) Explain the concept of multiprocessor and Multicore organizations (7)
(APR/MAY 2017)
(ii) Discuss about direct memory access (6) **(APR/MAY 2017)**
- 20.**(i) Explain the various structures of an operating system (8) **(APR/MAY 2017)**
(ii) Describe system calls and system programs in detail with neat sketch (5)
(APR/MAY 2017)
- 21.** Explain Cache memory and its mapping. (13) **(NOV/DEC 2017)**
- 22.** Describe evolution of operating system. (13) **(NOV/DEC 2017)**
- 23.** State the operating system structure. Describe the operating-system operations in detail. Justify the reason why the lack of a hardware-supported dual mode an cause serious shortcoming in an operating system? (13) **(APR/MAY 2018)**
- 24.** i) Give reason why caches are useful. What problems do they solve? What problems do they cause? If a cache can be made as large as the device for which it is caching why not make it that large and eliminate the device? (8)
ii) Describe the major activities of operating system with regards to file management. (5) **(APR/MAY 2018)**
- 25.** Explain in detail the difference architecture of OS starting from simple structure, Layered structure, micro kernel, modules and hybrid system with suitable example OS structures, including Google's android. (13) **(APR/MAY 2019)**
- 26.** (i) Discuss the pros and cons of single processor system, multi core system and clustered system. (8)
(ii) Explain the steps involved to transfer the stored historical information in a magnetic tapes to the CPU for further processing through various storage device.(5)
(APR/MAY 2019)
- 27.** (i) Explain important and need of cache with its working principle. (7) **(NOV/DEC 2021)**



- (ii) What is function of DMA operation? Explain with neat diagram. (6)
- 28.** (i) Discuss in detail about the varies memory hierarchies with neat block diagram. (7)
- (ii) Discuss about the functionality of system boot with respect to operating system. (6) **(NOV/DEC 2021)**



UNIT II PROCESS MANAGEMENT

11

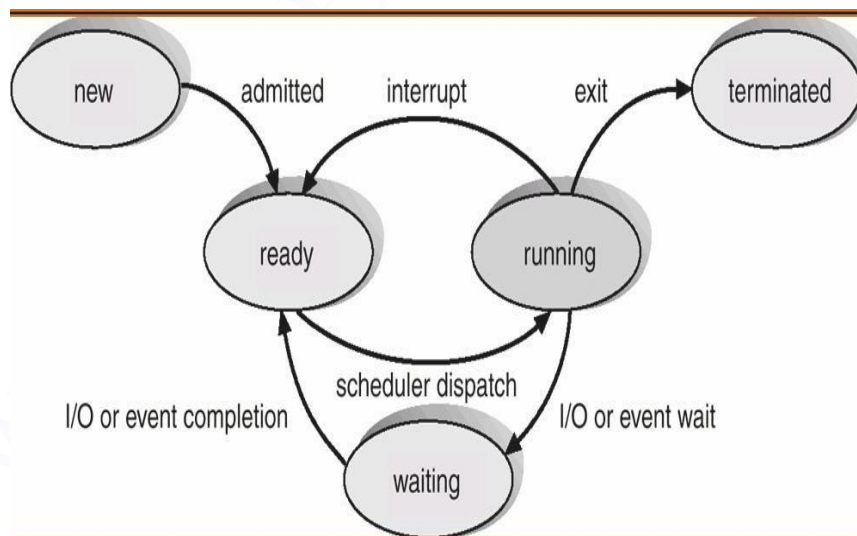
Processes - Process Concept - Process Scheduling - Operations on Processes - Inter-process Communication; CPU Scheduling - Scheduling criteria - Scheduling algorithms: Threads - Multithread Models - Threading issues; Process Synchronization - The Critical-Section problem - Synchronization hardware - Semaphores - Mutex - Classical problems of synchronization - Monitors; Deadlock - Methods for handling deadlocks, Deadlock prevention, Deadlock avoidance, Deadlock detection, Recovery from deadlock.

PART - A

1. Define Process?

A Process can be thought of as a program in execution. A process will need certain resources such as CPU time, memory, files & I/O devices to accomplish its task.

Draw & briefly explain the process states?or Name and draw five different process states with proper definition. (NOV/DEC 2017)

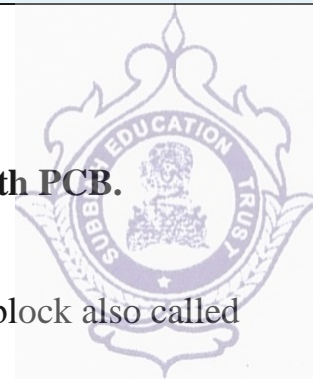


New- The process is being created.

Running – Instructions are being executed

Waiting – The process is waiting for some event to occur

Ready – The process is waiting to be assigned a processor



Terminated - the process has finished execution

2. What is process control block? List out the data field associated with PCB.

(APR/MAY 2015)

Each process is represented in the operating system by a process control block also called a task control block. (PCB) also called a task control block.

Process state
Process number
Program counter
CPU registers
Memory limits
List of open files
CPU scheduling information
Memory management information
Accounting information
I/O status information

3. What is meant by context switching?

Switching the CPU to another process requires saving the state of the old process and loading the saved state for the new process. This task is known as context switch.

4. Define co- operating process and independent process.

Independent process:

- A process is independent if it cannot affect or be affected by the other processes executing in the system.
- A process that does not share data with any other process is independent.

Cooperating process:

- A process is co-operating if it can affect or be affected by other processes executing in the system.
- Any process that shares data with any other process is cooperating.



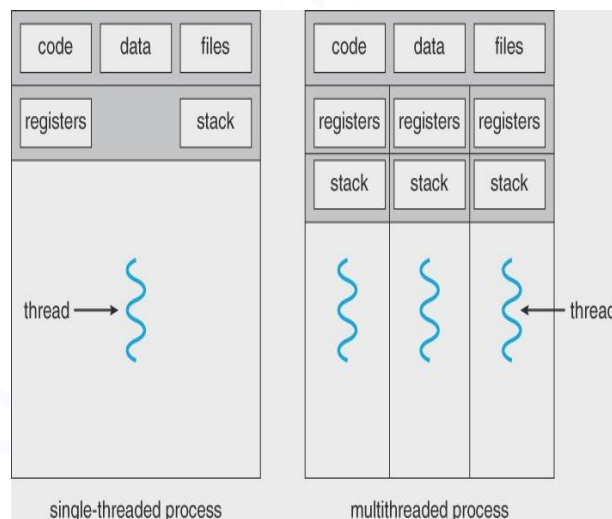
5. What are the benefits of multithreaded programming?

The benefits of multithreaded programming can be broken down into four major categories

- Responsiveness
- Resource sharing
- Economy scalability
- Utilization of multiprocessor architectures.

6. What is a thread?

A thread otherwise called a lightweight process (LWP) is a basic unit of CPU utilization, it comprises of a thread id, a program counter, a register set and a stack. It shares with other threads belonging to the same process its code section, data section, and operating system resources such as open files and signals.



7. Under What circumstances CPU scheduling decision takes place.

- (1) When a process switches from running state to waiting state
- (2) When a process switches from running state to ready state.
- (3) When a process switches from running state to waiting state to ready state
- (4) When a process terminates.



8. What are the various scheduling criteria for CPU scheduling?

The various scheduling criteria are

- CPU utilization
- Throughput
- Turnaround time
- Waiting time
- Response time

9. Write down the definition of TestAndSet() Instruction.

```
boolean TestAndSet (boolean &target)
{
    boolean rv = *target;
    *target = true;
return rv;
}
```

10. Define busy waiting and spinlock.

Busy waiting:-

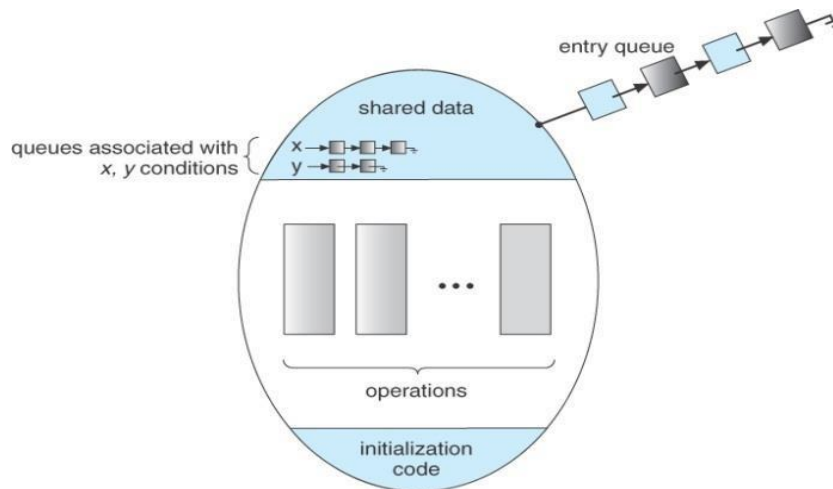
When a process is in its critical section, any other process that tries to enter its critical section must loop continuously in the entry code. This is called as busy waiting.

Spinlock:-

Busy waiting wastes CPU cycles that some other process might be able to use productively. This type of semaphore is also called a spinlock because the process spins while waiting for the lock.

11. What is meant by monitors?

A high level synchronization construct. A monitor type is an ADT which presents a set of programmer-defined operations that are provided mutual exclusion within the monitor.



12. What are the characterizations of deadlock?

1. Mutual exclusion: only one process at a time can use a resource.
2. Hold and wait: a process holding at least one resource is waiting to acquire additional resources held by other processes.
3. No preemption: a resource can be released only voluntarily by the process holding it, after that process has completed its task.
4. Circular wait: there exists a set $\{P_0, P_1, \dots, P_0\}$ of waiting processes such that P_0 is waiting for a resource that is held by P_1 , P_1 is waiting for a resource that is held by P_2 , ..., P_{n-1} is waiting for a resource that is held by P_n , and P_0 is waiting for a resource that is held by P_0 . Deadlock can arise if four conditions hold simultaneously.

13. Differentiate a Thread form a Process. (NOV/DEC 2012)

Threads

- Will by default share memory
- Will share file descriptors
- Will share file system context
- Will share signal handling

Processes

- Will by default not share memory



- Most file descriptors not shared
- Don't share file system context
- Don't share signal handling

14. What are the difference b/w user level threads and kernel level threads? (MAY /JUNE 2012) (MAY/ JUNE 2016) (NOV/DEC 2015)

User threads

User threads are supported above the kernel and are implemented by a thread library at the user level. Thread creation & scheduling are done in the user space, without kernel intervention. Therefore they are fast to create and manage blocking system call will cause the entire process to block

Kernel threads

Kernel threads are supported directly by the operating system .Thread creation, scheduling and management are done by the operating system. Therefore they are slower to create & manage compared to user threads. If the thread performs a blocking system call, the kernel can schedule another thread in the application for execution

15. What is the use of fork and exec system calls?

Fork is a system call by which a new process is created. Exec is also a system call, which is used after a fork by one of the two processes to place the process memory space with a new program.

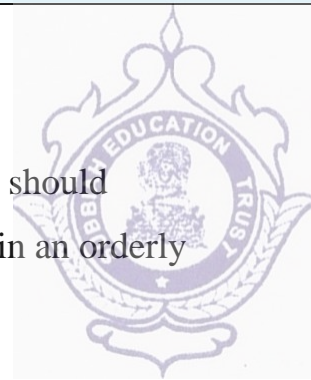
16. Define thread cancellation & target thread.

The thread cancellation is the task of terminating a thread before it has completed. A thread that is to be cancelled is often referred to as the target thread. For example, if multiple threads are concurrently searching through a database and one thread returns the result, the remaining threads might be cancelled.

17. What are the different ways in which a thread can be cancelled?

Cancellation of a target thread may occur in two different scenarios:

- **Asynchronous cancellation:** One thread immediately terminates the target thread



is called asynchronous cancellation.

- **Deferred cancellation:** The target thread can periodically check if it should terminate, allowing the target thread an opportunity to terminate itself in an orderly fashion.

18. Define PThreads

PThreads refers to the POSIX standard defining an API for thread creation and synchronization. This is a specification for thread behavior, not an implementation.

19. What is critical section problem?

Consider a system consists of 'n' processes. Each process has segment of code called a critical section, in which the process may be changing common variables, updating a table, writing a file. When one process is executing in its critical section, no other process can be allowed to execute in its critical section.

21. What are the requirements that a solution to the critical section problem must satisfy?

The three requirements are

- Mutual exclusion
- Progress & Bounded waiting

22. Define mutual exclusion. (MAY/JUNE 2013)

Mutual exclusion refers to the requirement of ensuring that no two process or threads are in their critical section at the same time.

i.e. If process P_i is executing in its critical section, then no other processes can be executing in their critical sections.

23. Define entry section and exit section.

The critical section problem is to design a protocol that the processes can use to cooperate. Each process must request permission to enter its critical section.

Entry Section: The section of the code implementing this request is the entry section.

Exit Section: The section of the code following the critical section is an exit section.



The general structure:

```
do {
    entry section
    critical section
    exit section
    remainder section
} while(1);
```

24. Give two hardware instructions and their definitions which can be used for implementing mutual exclusion.

TestAndSet

```
boolean TestAndSet (boolean &target)
{
    boolean rv = target;
    target = true;
    return rv;
}
```

Swap

```
void Swap (boolean &a, boolean &b)
{
    boolean temp = a;
    a = b;
    b = temp;
}
```

25. What is semaphore? Mention its importance in operating system. (APRIL/MAY 2010, NOV/DEC 2012)

A semaphore 'S' is a synchronization tool which is an integer value that, apart from initialization, is accessed only through two standard atomic operations; wait and



signal. Semaphores can be used to deal with the n-process critical section problem. It can be also used to solve various Synchronization problems.

26. How the mutual exclusion may be violated if the signal and wait operations are not executed automatically (MAY/JUNE 2014)

A wait operation atomically decrements the value associated with a semaphore. If two wait operations are executed on a semaphore when its value is 1, if the two operations are not performed atomically, then it is possible that both operations might proceed to decrement the semaphore value, thereby violating mutual exclusion

27. Define CPU scheduling.

CPU scheduling is the process of switching the CPU among various processes. CPU scheduling is the basis of multi programmed operating systems. By switching the CPU among processes, the operating system can make the computer more productive.

28. What is preemptive and non-preemptive scheduling? (NOV/DEC 2008 ,APRIL/MAY 2010, MAY /JUNE 2012)

Under non preemptive scheduling once the CPU has been allocated to a process, the process keeps the CPU until it releases the CPU either by terminating or switching to the waiting state.

Preemptive scheduling can preempt a process which is utilizing the CPU in between its execution and give the CPU to another process.

29. What is a Dispatcher?

The dispatcher is the module that gives control of the CPU to the process selected by the short-term scheduler. This function involves:

- Switching context.
- Switching to user mode.
- Jumping to the proper location in the user program to restart that program.

30. Define the term „dispatch latency“ (APR/MAY 2015)

The time taken by the dispatcher to stop one process and start another running



is known as dispatch latency.

31. Define throughput?

Throughput in CPU scheduling is the number of processes that are completed per unit time. For long processes, this rate may be one process per hour; for short transactions, throughput might be 10 processes per second.

32. What is turnaround time? (NOV/DEC 2013)

Turnaround time is the interval from the time of submission to the time of completion of a process. It is the sum of the periods spent waiting to get into memory, waiting in the ready queue, executing on the CPU, and doing I/O.

33. Define race condition.

When several process access and manipulate same data concurrently, then the outcome of the execution depends on particular order in which the access takes place is called race condition. To avoid race condition, only one process at a time can manipulate the shared variable.

34. Write the four situations under which CPU scheduling decisions take place (MAY/JUNE 2014)

CPU scheduling decisions take place under one of four conditions:

- When a process switches from the running state to the waiting state, such as for an I/O request or invocation of the wait () system call.
- When a process switches from the running state to the ready state, for example in response to an interrupt.
- When a process switches from the waiting state to the ready state, say at completion of I/O or a return from wait ().
- When a process terminates.

35. Define deadlock. (APRIL/MAY 2010)

A process requests resources; if the resources are not available at that time, the process enters a wait state. Waiting processes may never again change state, because the



resources they have requested are held by other waiting processes. This situation is called a deadlock.

36. What is the sequence in which resources may be utilized?

Under normal mode of operation, a process may utilize a resource in the following sequence:

- **Request:** If the request cannot be granted immediately, then the requesting process must wait until it can acquire the resource.
- **Use:** The process can operate on the resource.
- **Release:** The process releases the resource.

37. What are conditions under which a deadlock situation may arise? (MAY/JUNE 2009 , MAY/JUNE 2012, MAY/JUNE 2013) (NOV/DEC 2013)

A deadlock situation can arise if the following four conditions hold simultaneously in a system:

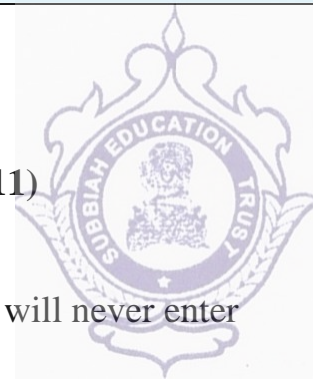
- a. Mutual exclusion
- b. Hold and wait
- c. No pre-emption
- d. Circular wait

38. What is a resource-allocation graph?

Resource allocation graph is directed graph which is used to describe deadlocks. This graph consists of a set of vertices V and a set of edges E . The set of vertices V is partitioned into two different types of nodes; P the set consisting of all active processes in the system and R the set consisting of all resource types in the system.

39. Define request edge and assignment edge.

A directed edge from process P_i to resource type R_j (denoted by $P_i \rightarrow R_j$) is called as request edge; it signifies that process P_i requested an instance of resource type R_j and is currently waiting for that resource. A directed edge from resource type R_j to process P_i (denoted by $R_j \rightarrow P_i$) is called an assignment edge; it signifies that an instance of



resource type has been allocated to a process P_i .

40. What are the methods for handling deadlocks? (APRIL/MAY 2011)

The deadlock problem can be dealt with in one of the three ways:

1. Use a protocol to prevent or avoid deadlocks, ensuring that the system will never enter a deadlock state.
2. Allow the system to enter the deadlock state, detect it and then recover.
3. Ignore the problem all together, and pretend that deadlocks never occur in the system.

41. How real-time Scheduling differs from normal scheduling? (NOV/DEC 2012)

In a normal Scheduling, we have two types of processes. User process & kernel Process. Kernel processes have time constraints. However, user processes do not have time constraints.

In a RTOS, all process are Kernel process & hence time constraints should be strictly followed. All process/task (can be used interchangeably) are based on priority and time constraints are important for the system to run correctly.

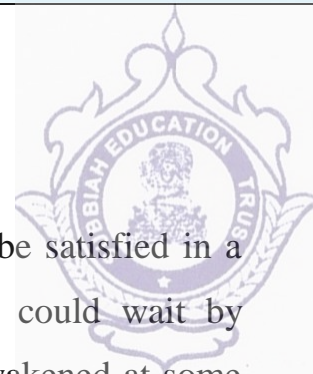
42. What do you meant by short-term scheduler (NOV/DEC 2010)

The selection process is carried out by the short-term scheduler or CPU scheduler. The scheduler selects the process from the process in memory that is ready to execute and allocates the CPU to the process.

43. What is the concept behind strong semaphore and spinlock? (NOV/DEC 2015)

A spinlock is one possible implementation of a lock, namely one that is implemented by busy waiting ("spinning"). A semaphore is a generalization of a lock (or, the other way around, a lock is a special case of a semaphore). Usually, *but not necessarily*, spinlocks are only valid within one process whereas semaphores can be used to synchronize between different processes, too.

A semaphore has a counter and will allow itself being acquired by one or several threads, depending on what value you post to it, and (in some implementations)



depending on what its maximum allowable value is.

43. What is the meaning of the term busy waiting? (May/June 2016)

Busy waiting means that a process is waiting for a condition to be satisfied in a tight loop without relinquish the processor. Alternatively, a process could wait by relinquishing the processor, and block on a condition and wait to be awakened at some appropriate time in the future.

44. Distinguish between CPU-bounded and I/O bounded processes (NOV/DEC 2016)

CPU Bound means the rate at which process progresses is limited by the speed of the CPU. A task that performs calculations on a small set of numbers, for example multiplying small matrices, is likely to be CPU bound.

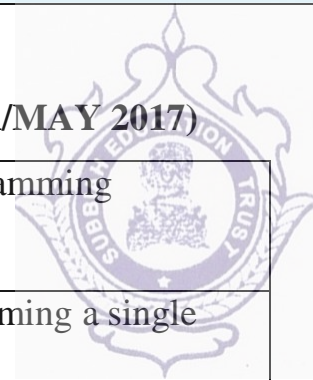
I/O Bound means the rate at which a process progresses is limited by the speed of the I/O subsystem. A task that processes data from disk, for example, counting the number of lines in a file is likely to be I/O bound.

45. What resources are required to create threads (NOV/DEC 2016)

When a thread is created, the thread does not require any new resources to execute the thread shares the resources like memory of the process to which they belong. The benefit of code sharing is that it allows an application to have several different threads of activity all within the same address space.

46. "Priority inversion is a condition that occurs in real time systems where a low priority process is starved because higher priority processes have gained hold of the CPU"-Comment on this statement. (APR/MAY 2017)

Priority inversion is a problematic scenario in scheduling in which a high priority task is indirectly preempted by a lower priority task effectively "inverting" the relative priorities of the two tasks. This violates the priority model that high priority tasks can only be prevented from running by higher priority tasks and briefly by low priority tasks which will quickly complete their use of a resource shared by the high and low priority tasks.



47. Differentiate single threaded and multi-threaded processes. (APR/MAY 2017)

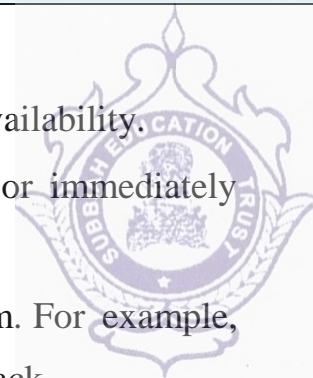
S. No.	Multithreaded Programming	Single Threaded Programming
1	In this type of programming multiple threads run at the same time	In this type of programming a single thread runs at a time.
2	Multi-threaded model doesn't use event loop with polling	Single threaded model uses a process event loop with polling
3	CPU time is never wasted.	CPU time is wasted.
4	Idle time is minimum.	Idle time is more.
5	It results in more efficient programs.	It results in less efficient programs.
6	When one thread is paused due to some reason, other threads run as normal.	When one thread is paused, the system waits until this thread is resumed.

48. Elucidate mutex locks with its procedure. (NOV/DEC 2017)

Mutex is a program object that allows multiple program threads to share the same resource, such as file access, but not simultaneously. When a program is started a mutex is created with a unique name. After this stage, any thread that needs the resource must lock the mutex from other threads while it is using the resource. The mutex is set to unlock when the data is no longer needed or the routine is finished. In mutex locks approach, in the entry section of code, a LOCK is acquired over the critical resources modified and used inside critical section, and in the exit section that LOCK is released. As the resource is locked while a process executes its critical section hence no other process can access it.

49. What are the benefits of synchronous and asynchronous communication? (APR/MAY 2018)

Benefits of synchronous communication:



- Synchronous communication enables flexibility and offer higher availability.
- There's less pressure on the system to act on the information or immediately respond in some way.
- Also, one system being down does not impact the other system. For example, emails –thousands of emails can be sent without having to revert back..

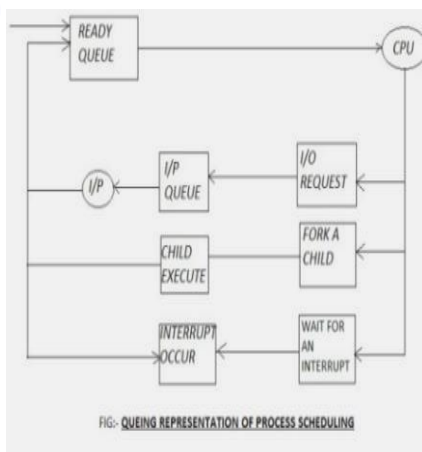
Benefits of Asynchronous communication:

- Asynchronous message passing allows more parallelism.
- Since a process does not block, it can do some computation while the message is in transit.
- In the case of receive, this means a process can express its interest in receiving messages on multiple ports simultaneously.

50. Give a programming example in which multithreading does not provide better performance than single-threaded solutions. (APR/MAY 2018)

Multi-threading does not perform well for any sequential program. For example; program to calculate an individual tax return. Another example where multithreading does not work good would be shell program like -Korn|| shell.

51. Give the queuing diagram representation of process scheduling. (APR/MAY 2019)



52. List out the benefits and challenge of thread handling. (APR/MAY 2019)

Benefits



- Responsiveness.
- Resource sharing
- Economy
- Scalability.

Challenges

- Dividing activities
- Balance
- Data splitting
- Data dependency
- Testing and debugging

PART-B&C

1) Explain the FCFS, preemptive and non-preemptive versions of Shortest-Job First and Round Robin (time slice = 2) scheduling algorithms with Gantt charts for the four Processes given. Compare their average turnaround and waiting time. [E] (NOV/DEC 2012)

Process	Arrival Time	Waiting Time
P1	0	8
P2	1	4
P3	2	9
P4	3	5

2) Discuss how scheduling algorithms are selected for a system. What are the criteria considered? Explain the different evaluation Methods. (MAY/JUNE 2014)

3) Write in detail about several CPU scheduling algorithms. (APRIL/MAY2011)

4) What is critical section? Specify the requirements for a solution to critical section problem. (NOV/DEC 2012)

5) How monitors help in process synchronization. (NOV/DEC 2009)



- 6) Write in detail about deadlock avoidance. (NOV/DEC 2009)
- 7) Write in detail about deadlock recovery. (APRIL/MAY2011)
- 8) Explain the Banker algorithm for deadlock avoidance in detail with an example. (APRIL/MAY2010, NOV/DEC 2012) (NOV/DEC 2013)
- 9) Consider the following set of processes, with the length of the CPU – burst time given in Milliseconds:

Process	Burst Time	Priority
P1	10	3
P2	1	1
P3	2	3
P4	1	4
P5	5	2

The processes are arrived in the order P1, P2, P3, P4, P5, all at time 0.

- 1. Draw 4 Gantt charts illustrating the execution of these processes using FCFS, SJF Priority and RR (Time Slice = 1) scheduling
- 2. What is the turnaround time of each process for each of the scheduling?
- 3. Calculate the waiting time for each of the process [E] (MAY/JUNE2012) (NOV/DEC2015)
- 10) Consider the following questions based on the banker’s algorithm: [E] (MAY/JUNE 2012)

Process	Allocation	Max	Available
P0	A B C D	A B C D	A B C D
P1	0 0 1 2	0 0 1 2	1 5 2 0
P2	1 0 0 0	1 7 5 0	
P3	1 3 5 4	2 3 5 6	
P4	0 6 3 2	0 6 5 2	
P5	0 0 1 4	0 6 5 6	



- (1) Define safety algorithm.
- (2) What is the content of the matrix Need?
- (3) Is the system in a safe state?
- (4) If a request from process P1 arrives for (0, 4, 2, 0), can the request be granted immediately?

11) (i) What is meant by critical section problem? Propose a solution based on bakery algorithm.

(ii) Consider the following snapshot of a system:

P0 – P4 are 5 processes present and A, B, C, D are the resources. The maximum need of a Process and the allocated resources details are given in the table.

Answer the following based on banker’s algorithm.

- (1) What is the content of NEED matrix?
- (2) Is the system in a safe state?
- (3) If a request from process P0 arrives for (0, 2, 0) can the request be granted immediately. [E]

	Allocation			Max			Available		
	A	B	C	A	B	C	A	B	C
P0	0	1	0	7	5	3	3	3	2
P1	2	0	0	3	2	2			
P2	3	0	2	9	0	2			
P3	2	1	1	2	2	2			
P4	0	0	2	4	3	3			

12) Discuss the threading issues which are considered with multithreaded programs.

MAY/JUNE 2014)(APRIL/MAY2011, MAY/JUNE 2012)



Consider the following snapshot of a system:

P0-P4 are 5 processes present and A, B, C, D are the resources .The maximum need of a process and the allocated resources details are given in the table.

	Allocation				Max				Available			
	A	B	C	D	A	B	C	D	A	B	C	D
P0	0	0	1	2	0	0	1	2	1	5	2	0
P1	1	0	0	0	1	7	5	0				
P2	1	3	5	4	2	3	5	6				
P3	0	6	3	2	0	6	5	2				
P4	1	0	1	4	0	6	5	6				

Answer the following based on banker’s algorithm

- i) What is the content of NEED matrix?
- ii) Is the system in a safe state?
- iii) Which processes may cause deadlock if the system is not safe.
- iv) If a request from process p1 arrives for (0, 4, 3, 1) can the request be granted immediately? Justify.

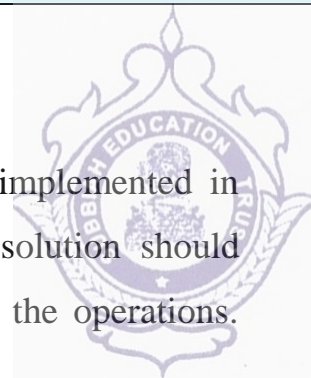
[E] (MAY/JUNE 2014)

13) Discuss in detail the critical section problem and also write the algorithm for Readers-Writers Problem with semaphores (NOV/DEC 2013)

14) Explain the FCFS, preemptive and non-preemptive versions of Shortest-Job First and Round Robin (time slice = 2) scheduling algorithms with Gantt charts for the four Processes given. Compare their average turnaround and waiting time.

(APR/MAY 2015)

Process	Arrival Time	Waiting Time
P1	0	10
P2	1	6
P3	2	12
P4	3	15



Discuss how deadlocks could be detected in detail. **(APR/MAY 2015)**

15) Show how wait () and signal () semaphore operations could be implemented in multiprocessor environments using the test and set instruction. The solution should exhibit minimal busy waiting. Develop pseudo code for implementing the operations.

(APR/MAY 2015)

16) Discuss about the issues to be considered in the multithreaded program.

(APR/MAY 2015)

17) (i) Explain thread and SMP management.

(ii) Illustrate Semaphores with neat example.

(iii) The operating system contains 3 resources, the number of instance of each resource type are 7, 7, 10. The current resource allocation state is as shown below:

Process	Current Allocation			Maximum Need		
	R ₁	R ₂	R ₃	R ₁	R ₂	R ₃
P ₁	2	2	3	3	6	8
P ₂	2	0	3	4	3	3
P ₃	1	2	4	3	4	4

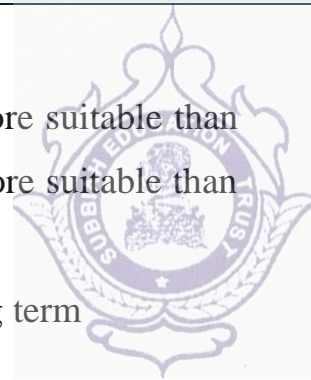
18) Is the current allocation in a safe state? [E] **(NOV/DEC 2015) (MAY/JUNE 2016)**

20) (i) Is it possible to have concurrency but not parallelism? Explain.

(ii) Consider a system consisting of four resources of the same type that are shared by three processes, each of which needs at most two resources. Show that the system is deadlock free.

(i) Describe the actions taken by a kernel to context-switch between processes.

(ii) Provide two programming examples in which multithreading does not provide better performance than a single-threaded solution. **(MAY/JUNE 2016)**



19) (i) Give an example of a situation in which ordinary pipes are more suitable than named pipes and an example of a situation in which named pipes are more suitable than ordinary pipes. **(8) (NOV/DEC 2016)**

(ii) Describe the differences among short-term, medium-term, and long term scheduling **(8) (NOV/DEC 2016)**

20) (i) Explain why interrupts are not appropriate for implementing synchronization primitives in multiprocessor systems **(8) (NOV/DEC 2016)**

(i) What are the different thread libraries used? Explain any one with example **(8) (NOV/DEC 2016)**

21) Consider the following set of processes, with the length of the CPU-burst time in given ms:

Process	Burst Time	Arrival Time
P1	8	0.00
P2	4	1.001
P3	9	2.001
P4	5	3.001
P5	3	4.001

Draw four Gantt charts illustrating the execution of these processes using FCFC, SJF, Priority and RR (Quantum=2) scheduling. Also calculate waiting time and turnaround time for each scheduling algorithms [E]. **(13) (APR/MAY 2017)**

22) What is a race condition? Explain how a critical section avoids this condition. What are the properties which a data item should possess to implement a critical section?

Describe a solution to the Dining philosopher problem so that no races arise. **(13) (APR/MAY 2017) (APR/MAY 2019).**

23) i) What is a process ? Discuss components of process and various states of a process with the help of a process state transition diagram. **(8) (NOV/DEC 2017)**

ii) Write the difference between user thread and kernel thread. **(5) (NOV/DEC 2017)**



24) i) What is the average turnaround time for the following processes using

- a) FCFS (3)
- b) SJF non-preemptive. (3)
- c) Preemptive SJF.(3) (NOV/DEC 2017)

Process	Arrival Time	Burst Time
P1	0.0	8
P2	0.4	4
P3	1.0	1

ii) With example elucidate livelock. (4) (NOV/DEC 2017)

25) Describe the difference among short-term, medium-term and long term scheduling with suitable example. (APR/MAY 2018)

29) Explain the differences in the degree to which the following scheduling algorithms discriminate in favor of short processes: (APR/MAY 2018)

- i) RR
- ii) Multilevel feedback queues.

30) What do you mean by term synchronization? 'What is Semaphore? Explain how semaphore can used as synchronization tool. Consider a coke machine that has 10 slots. The producer is the delivery person and the consumer is the student using the machine- It uses the following three semaphores (15)(APR/MAY 2017)

semaphore mutex

semaphore fullBuffer /* Number of filled slots: */

semaphore emptyBuffer /* Number of empty slots */

(i) Write pseudo code for delivery_person() and student()

(ii)What will be the initial values of the semaphores?

(iii)Write a solution that guarantees the mutual exclusion and has no deadlocks



31) What is deadlock? What are the necessary conditions for deadlock to occur? Explain the deadlock prevention method of handling deadlock. (15) (APR/MAY 2017)

Consider the following information about resources in a system.

- (i) There are two classes of allocatable resource labeled R1 and R2
- (ii) There are two instances of each resource
- (iii) There are four processes labeled p1 through p4
- (iv) There are some resource instances already allocated to processes as follows:
 - One instance of R1 held by p2, another held by p3
 - One instance of R2 held by p1, another held by p4
- (v) Some processes have requested additional resources, as follows:-
 - p1 wants one instance of R1 .
 - p3 wants one instance of R2

(1) Draw the resource allocation graph for this system

(2) What is the state (runnable, waiting) of each process? For each process that is waiting indicate what it is waiting for

(3) Is this system deadlocked? If so, state which processes are involved. If not, give an execution sequence that eventually ends, showing resource acquisition and release at each step.

32) Consider the following system snapshot using data structures in the Banker's algorithm, with resources A, B, C and D and process P0 to P4 : [E](NOV/DEC 2017)

	Max	Allocation	Need	Available
	ABCD	ABCD	ABCD	ABCD
P0	6 0 1 2	4 0 0 1		3 2 1 1
P1	1 7 5 0	1 1 0 0		
P2	2 3 5 6	1 2 5 4		
P3	1 6 5 3	0 6 3 3		
P4	1 6 5 6	0 2 1 2		

Using Banker's algorithm, answer the following questions:



a) How many resources of type A, B, C and D are there? (2)

b) What are the contents of the need matrix? (3)

c) Is the system in a safe state? Why? (3)

d) If a request from process P4 arrives for additional resources of (1,2,0,0), can the Banker's algorithm grant the request immediately? Show the new system state and other criteria. (7)

33) i) Consider the atomic fetch-and-set x, y instruction unconditionally sets the memory location x to 1 and fetches the old value of x in y without allowing any intervening access to the memory location x. Consider the following implementation of P and V functions on a binary semaphore. (15) (NOV/DEC 2017)

```
void P (binary_semaphore *s) {
```

```
    unsigned y;
```

```
    unsigned *x = & (s->value);
```

```
    do {
```

```
        fetch-and-set x, y; b
```

```
    } While (3');
```

```
}
```

```
void V (binary_semaphore *s) {
```

```
    S->value = 0;
```

```
}
```

Write whether the implementation may or may not work if context switching is disabled in P.

(ii) Consider a situation where we have a file shared between many people. If one of the people tries editing the file, no other person should be reading or writing at the same time, otherwise changes will not be visible to him/her. However if some person is reading the file, then others may read it at the same time. (NOV/DEC 2017)

a) What kind of situation is this?

b) Consider the following problem parameters to solve this situation.



Problem parameters:

- 1) One set of data is shared among a number of processes.
- 2) Once a writer is ready, it performs its write. Only one writer may write at a time.
- 3) If a process is writing, no other process can read it.
- 4) If at least one reader is reading, no other process can write.
- 5) Readers may not write and only read.

34) Consider a system consisting of 'm' resources of the same type being shared by n Processes. Resource can be requested and released by processes only one at a time. Show that the system is deadlock free if the following two conditions hold :**(15) (APR/MAY 2018)**

- i) The maximum need of each process is between 1 and m resources.
- ii) The sum of all maximum needs is less than $m + n$.

35. Consider the following set of processes, with the length of the CPU burst given in milliseconds: **[E] (APR/MAY 2018)**

The process is assumed to have arrived in the order P1, P2, P3, P4, P5 time 0.

i) Draw Gantt charts that illustrate the execution of these processes using the scheduling algorithms FCFS (smaller priority number implies higher priority) and RR (quantum = 1). **(10)**

ii) What is the waiting time of each process for each of the scheduling algorithms? **(5)**

25. Write the algorithm using test and set() instruction that satisfy all the critical section requirements. **(5) (APR/MAY 2019)**

26. Consider the following snapshot of a system:

P0-P4 are 5 processes present and A, B, C, D are the resources. The maximum need of a process and the allocated resources details are given in the table.

	Allocation				Max				Available			
	A	B	C	D	A	B	C	D	A	B	C	D
P0	2	0	0	1	4	2	1	2	3	3	2	1



P1	3	1	2	1	5	2	5	2				
P2	2	1	0	3	2	3	1	6				
P3	1	3	1	2	1	4	2	4				
P4	1	4	3	2	3	6	6	5				

Answer the following based on banker’s algorithm

1. Illustrate that the system is in safe state by demonstrating an order in which the process may complete?
2. If a request from a process p1 arrives for (1,1,0,0) can the request be granted immediately.
3. If the request from p4 arrives for (0,0,2,0) can the request be granted immediately?

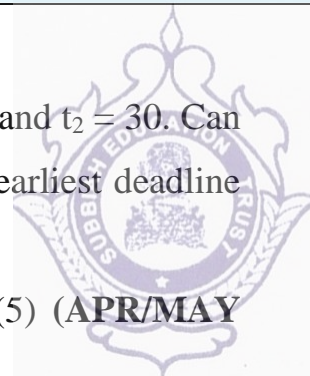
(13) [E] (APR/MAY 2019)

39. (i) Consider the following set of processes with the length of CPU- burst time given in milliseconds.

Process	Burst Time	Priority	Arrival Time
P1	10	3	0
P2	1	1	1
P3	2	3	2
P4	1	4	1
P5	5	2	2

Draw the Gantt chart for the execution of these processes using FCFS, SJF, SRTS, pre-emptive and non-pre-emptive priority, round robin with time slice of 2 ms. Find the average waiting and turnaround time using each of the methods. (10)

(ii) Explain Multi level queue and multi-level feedback queue scheduling with suitable example. (5) **(APR/MAY 2019)**



40. (i) Consider two processes, p_1 and p_2 where $p_1 = 50$, $t_1 = 25$, $p_2 = -75$ and $t_2 = 30$. Can these two processes be scheduled using rate-monotonic scheduling and earliest deadline first scheduling. Illustrate your answer using Gantt charts. (10)

(ii) Explain in detail about paging in 32 bit and 64 bit architectures. (5) (APR/MAY 2019)

41. (i) Explain banker algorithm for deadlock avoidance with suitable example. (7)

(ii) A system has four processes and five resources. The current allocation and maximum need are as follows (NOV/DEC 2021)

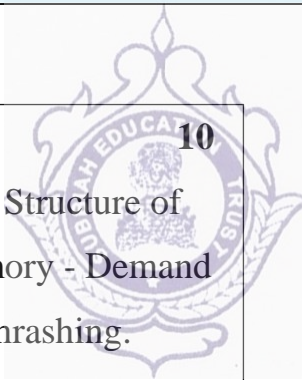
	Allocated	Maximum	Available
Process A	1 0 2 1 1	1 1 2 1 3	0 0 x 1 1
Process B	2 0 1 1 0	2 2 2 1 0	
Process C	1 1 0 1 0	2 1 3 1 0	
Process D	1 1 1 1 0	1 1 2 2 1	

Consider value of x as 1,2,3.

What is the smallest value of x in which the above system become a safe state?

42. (i) What is critical section? Discuss in detail reader's writer's problem. (7) (NOV/DEC 2021)

(ii) Define Deadlock. State the condition for deadlock. Explain the steps involved in deadlock recovery. (6)



UNIT III MEMORY MANAGEMENT

10

Main Memory - Swapping - Contiguous Memory Allocation – Paging - Structure of the Page Table - Segmentation, Segmentation with paging; Virtual Memory - Demand Paging – Copy on Write - Page Replacement - Allocation of Frames –Thrashing.

1. Why page are sizes always powers of 2?

Recall that paging is implemented by breaking up an address into a page and offset number. It is most efficient to break the address into X page bits and Y offset bits, rather than perform arithmetic on the address to calculate the page number and offset. Because each bit 25 26 position represents a power of 2, splitting an address between bits results in a page size that is a power of 2.

2. Consider a logical address space of eight pages of 1024 words each, mapped onto a physical memory of 32 frames.[E]

- a. How many bits are there in the logical address?
- b. How many bits are there in the physical address?

Each page/frame holds 1K; we will need 10 bits to uniquely address each of those 1024 addresses. Physical memory has 32 frames and we need 5 bits to address each frame, requiring in total 5+10=15 bits. A logical address space of 64 pages requires 6 bits to address each page uniquely, requiring 16bits in total.

- a. Logical address: 13 bits
- b. Physical address: 15 bits

3. In the IBM/370, memory protection is provided through the use of keys. A key is a 4-bit quantity. Each 2K block of memory has a key (the storage key) associated with it. The CPU also has a key (the protection key) associated with it. A store operation is allowed only if both keys are equal, or if either is zero. Which of the following memory-management schemes could be used successfully with this hardware? [E]

- a. Bare machine



- b. Single-user system
- c. Multiprogramming with a fixed number of processes
- d. Multiprogramming with a variable number of processes
- e. Paging
- f. Segmentation

Answer:

- a. Protection not necessary set system key to 0.
- b. Set system key to 0 when in supervisor mode.
- c. Region sizes must be fixed in increments of 2k bytes, allocate key with memory blocks.
- d. Same as above.
- e. Frame sizes must be in increments of 2k bytes, allocate key with pages.
- f. Segment sizes must be in increments of 2k bytes, allocate key with segments

4. What is address binding?

The process of associating program instructions and data to physical memory addresses is called address binding, or relocation.

5. Difference between internal and external fragmentation (NOV/DEC 2013)

Internal fragmentation is the area occupied by a process but cannot be used by the process. This space is unusable by the system until the process release the space.

External fragmentation exists when total free memory is enough for the new process but it's not contiguous and can't satisfy the request. Storage is fragmented into small holes.

6. Consider the following page reference string: 1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6. How many page faults would occur for the following replacement algorithms, assuming one, two, three, four, five, six, or seven frames? Remember all frames are initially empty, so your first unique pages will all cost one fault each. • LRU replacement • FIFO replacement • Optimal replacement Answer:



Number of frames	LRU	FIFO	Optimal
1	20	20	20
2	18	18	15
3	15	16	11
4	10	14	8
5	8	10	7
6	7	10	7
7	7	7	7

7. Define dynamic loading.

To obtain better memory-space utilization dynamic loading is used. With dynamic loading, a routine is not loaded until it is called. All routines are kept on disk in a relocatable load format. The main program is loaded into memory and executed. The calling routine checks whether the routine has been loaded. If not, there relocatable linking loader is called to load the desired program into memory.

8. Define dynamic linking.

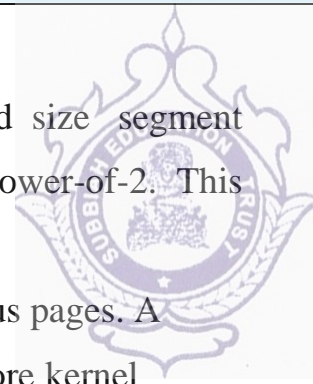
Dynamic linking is similar to dynamic loading, rather than loading being postponed until execution time, linking is postponed. This feature is usually used with system libraries, such as language subroutine libraries.

9. What are overlays? Compare swapping and overlays

To enable a process to be larger than the amount of memory allocated to it, overlays are used. The idea of overlays is to keep in memory only those instructions and data that are needed at a given time. When other instructions are needed, they are loaded into space occupied previously by instructions that are no longer needed.

10. List the strategies for managing free memory in kernel?

1. Buddy System
2. Slab Allocation



Buddy System: - The buddy system allocates memory from a fixed size segment consists of physical contiguous pages. Memory is allocated using power-of-2. This allocation satisfy request in units sized as a power of 2.

Slab Allocation:- A Slab is made up of one or more physically contiguous pages. A cache consists of one or more slabs. The slab allocation uses caches to store kernel Objects.

**11. What is virtual memory? Mention its advantages. (NOV/DEC 2012)
(MAY/JUNE 2014)**

Virtual memory is a technique that allows the execution of processes that may not be completely in memory. It is the separation of user logical memory from physical memory. This separation provides an extremely large virtual memory, when only a smaller physical memory is available.

The main visible advantage of this scheme is that programs can be larger than physical memory.

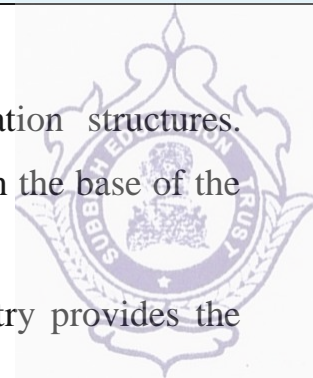
12. Define Demand paging and write advantages.

Virtual memory is commonly implemented by demand paging. In demand paging, the pager brings only those necessary pages into memory instead of swapping in a whole process. Thus it avoids reading into memory pages that will not be used anyway, decreasing the swap time and the amount of physical memory needed.

13. What is the purpose of paging the page tables?

In certain situations the page tables could become large enough that by paging the page tables, one could simplify the memory allocation problem (by ensuring that everything is allocated as fixed-size pages as opposed to variable-sized chunks) and also enable the swapping of portions of page table that are not currently used.

14. Compare paging with segmentation with respect to the amount of memory required by the address translation structures in order to convert virtual addresses to physical addresses.



- Paging requires more memory overhead to maintain the translation structures. Segmentation requires just two registers per segment: one to maintain the base of the segment and the other to maintain the extent of the segment.
- Paging on the other hand requires one entry per page, and this entry provides the physical address in which the page is located

15. What do you mean by thrashing? (APR/MAY 2015) (APR/MAY 2019)

Thrashing is the coincidence of high page traffic and low CPU utilization.

16. How do you limit the effects of thrashing?

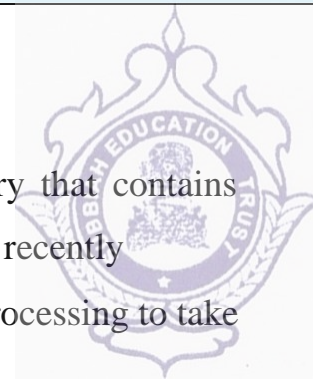
To limit the effect of thrashing we can use **local replacement algorithm**. With Local replacement algorithm, if the process starts thrashing, it cannot steal frames from another process and cause the latter to thrash as well. The problem is not entirely solved. Thus the effective access time will increase even for the process that is not thrashing.

17. What do mean by page fault?

Page fault is the situation in which the page is not available whenever a processor needs to execute it.

18. Differentiate between Global and Local page replacement algorithms.

Global Page Replacement Algorithm	Local Page Replacement Algorithm
Allows a process to select a replacement frame from the set of all frames, even if that frame is currently allocated to some other process	Each process select form only its own set of allocated frames
The number of frames allocated to a process can change since a process may happen to select only frames allocated to other processes, thus increasing the number of frames allocated to it.	The number of frames allocated to a process does not change
A process cannot control its own page-fault rate	A process can control its own page-fault rate

**19. Define TLB.**

- Translation Look-Aside Buffer, a table in the processors_memory that contains information about the pages in memory the processor has accessed recently
- The TLB enables faster computing because it allows the address processing to take place independent of the normal address-translation pipeline

20. Define Pre paging.

It is an attempt to prevent the high level of initial paging. This strategy is to bring into memory at one time all the pages the will be needed.

Example: - Solaris uses pre paging.

21. Define logical address and physical address.

An address generated by the CPU is referred as logical address. An address seen by the memory unit that is the one loaded into the memory address register of the memory is commonly referred as physical address

22. What is the main function of the memory-management unit?

The runtime mapping from virtual to physical addresses is done by a hardware device called a memory management unit (MMU).

23. What is difference between demands paging n pure demand paging?

In demand paging, a page is not loaded into main memory until it is needed. In pure demand paging, even a single page is not loaded into memory initially. Hence pure demand paging causes a page fault.

24. Define Copy-on-write.

Copy-on-write finds its main use in virtual memory operating systems; when a process creates a copy of itself, the pages in memory that might be modified by either the process or its copy are marked copy-on-write.

25. Define swapping. (NOV/DEC 2013)

A process needs to be in memory to be executed. However a process can be swapped temporarily out of memory to a backing store and then brought back into



memory for continued execution. This process is called swapping.

26. What are the common strategies to select a free hole from a set of available holes?

The most common strategies are

- A.** First fit **B.** Best fit **C.** Worst fit

27. Define lazy swapper.

Rather than swapping the entire process into main memory, a lazy swapper is used. A lazy swapper never swaps a page into memory unless that page will be needed.

28. Define effective access time.

Let p be the probability of a page fault ($0 \leq p \leq 1$). The value of p is expected to be close to 0; that is, there will be only a few page faults. The effective access time is

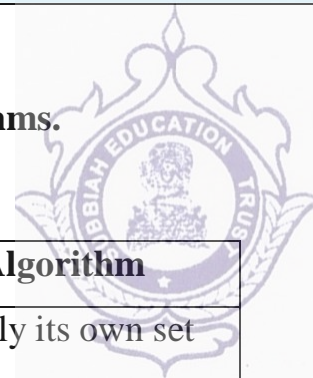
Effective access time = $(1-p) * m_a + p * \text{page fault time}$. m_a : memory-access time

29. What is the basic approach of page replacement?

If no frame is free is available, find one that is not currently being used and free it. A frame can be freed by writing its contents to swap space, and changing the page table to indicate that the page is no longer in memory. Now the freed frame can be used to hold the page for which the process faulted.

30. What is the various page replacement algorithms used for page replacement?

- FIFO page replacement
- Optimal page replacement
- LRU page replacement
- LRU approximation page replacement
- Counting based page replacement
- Page buffering algorithm.



31. Differentiate between Global and Local page replacement algorithms.

(NOV/DEC 2012)

Global Page Replacement Algorithm	Local Page Replacement Algorithm
Allows a process to select a replacement frame from the set of all frames, even if	Each process select form only its own set of allocated frames
The number of frames allocated to some other process can change since a process may happen to select only frames allocated to other processes, thus increasing the number of frames allocated to it.	The number of frames allocated to a process does not change
A process cannot control its own page-fault rate	A process can control its own page-fault rate

32. What are the major problems to implement demand paging?

The two major problems to implement demand paging is developing

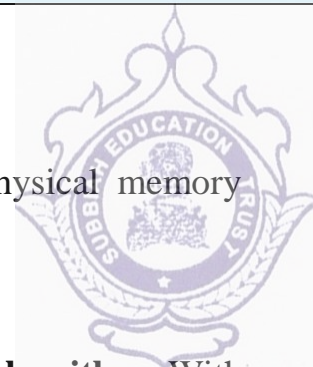
- Frame allocation algorithm
- Page replacement algorithm

33. What is a reference string?

An algorithm is evaluated by running it on a particular string of memory references and computing the number of page faults. The string of memory reference is called a reference string.

34. Differentiate a page from a segment. (APRIL/MAY 2010)

In segmentation, the address space is typically divided into a preset number of segments like data segment (read/write), code segment (read-only), stack (read/write) etc. And the programs are divided into these segments accordingly. Logical addresses are represented as tuple <segment, offset>. While with paging, the address space is divided into a sequence of fixed size units called "pages". And logical addresses take the form of a tuple <page, offset>.



35. What is address binding? (NOV/DEC 2010)

The process of associating program instructions and data to physical memory addresses is called address binding, or relocation.

36. How do you limit the effects of thrashing? (APRIL/MAY 2011)

To limit the effect of thrashing we can use **local replacement algorithm**. With Local replacement algorithm, if the process starts thrashing, it cannot steal frames from another process and cause the latter to thrash as well. The problem is not entirely solved. Thus the effective access time will increase even for the process that is not thrashing.

37. Mention the significance of LDT and SDT. (APR/MAY 2015)

The Global Descriptor Table or GDT is a data structure used by Intel x86-family processors starting with the 80286 in order to define the characteristics of the various memory areas used during program execution, including the base address, the size and access privileges like executability and writability. These memory areas are called segments.

The **Local Descriptor Table** (LDT) is a memory table used in the x86 architecture in protected mode and contains memory segment descriptors: start in linear memory, size, executability, writability, access privilege, actual presence in memory, etc.

- The LDT is supposed to contain memory segments which are private to a specific program, while the GDT is supposed to contain global segments.
- The x86 processors contain facilities for automatically switching the current LDT on specific machine events, but no facilities for automatically switching the GDT.
- The LDT is the sibling of the Global Descriptor Table (GDT) and defines up to 8192 memory segments accessible to programs –
- Unlike the GDT, the zeroth entry is a valid entry, and can be used like any other LDT entry.
- Unlike the GDT, the LDT cannot be used to store certain system entries: TSSs or LDTs.



38. Define demand paging in memory management. What are the steps required to handle a page fault in demand paging. (Nov/Dec 2015)

A demand paging system is quite similar to a paging system with swapping where processes reside in secondary memory and pages are loaded only on demand, not in advance. When a context switch occurs, the operating system does not copy any of the old program's pages out to the disk or any of the new program's pages into the main memory. Instead, it just begins executing the new program after loading the first page and fetches that program's pages as they are referenced. While executing a program, if the program references a page which is not available in the main memory because it was swapped out a little ago, the processor treats this invalid memory reference as a page fault and transfers control from the program to the operating system to demand the page back into the memory.

39. How does the system detect thrashing? (May/Jun 2016)

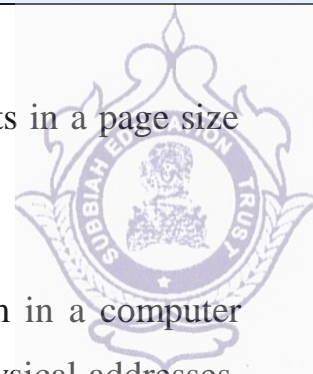
Thrashing is caused by under allocation of the minimum number of pages required by a process, forcing it to continuously page fault. The system can detect thrashing by evaluating the level of CPU utilization as compared to the level of multiprogramming. It can be eliminated by reducing the level of multiprogramming.

40. Name two differences between logical and physical addresses. (May/Jun 2016)

A logical address does not refer to an actual existing address; rather, it refers to an abstract address in an abstract address space. Contrast this with a physical address that refers to an actual physical address in memory. A logical address is generated by the CPU and is translated into a physical address by the memory management unit (MMU). Therefore, physical addresses are generated by the MMU.

41. Why page sizes are always powers of 2? (NOV/DEC 2016)

Paging is implemented by breaking up an address into a page and offset number. It is most efficient to break the address into X page bits and Y offset bits, rather than perform arithmetic on the address to calculate the page number and offset. Because each bit



position represents a power of 2, splitting an address between bits results in a page size that is a power of 2.

42. What is the purpose of paging the page tables (NOV/DEC 2016)

A page table is the data structure used by a virtual memory system in a computer operating system to store the mapping between virtual addresses and physical addresses. Virtual addresses are used by the accessing process, while physical addresses are used by the hardware, or more specifically, by the RAM subsystem.

43. What is the difference between a user-level instruction and a privilege instruction? Which of the following instruction should be privileged and only allowed to execute in kernel mode?

(a) Load a value from a memory address to a general-purpose register

(b) Set a new value in the program counter (PC) register

(c) Turn off interrupts (APR/MAY 2017)

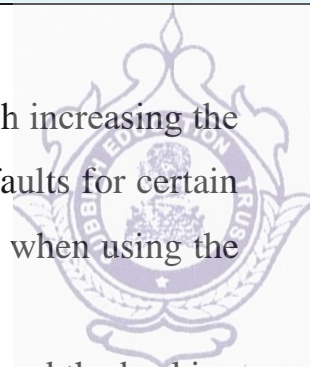
Privileged instruction

- A privileged instruction is a processor op-code (assembler instruction) which can only be executed in "supervisor" (or Ring-0) mode. These types of instructions tend to be used to access I/O devices and protected data structures from the windows kernel.

User Level instruction

- User-level is generic and can run on any operating system.
 - ✓ Load a value from a memory address to a general-purpose register (User Level Instruction)
 - ✓ Set a new value in the program counter (PC) register (User Level Instruction)
 - ✓ Turn off interrupts (privilege Instruction)

44. Will optimal page replacement algorithm suffer from Belady's anomaly? Justify your answer (APR/MAY 2017)



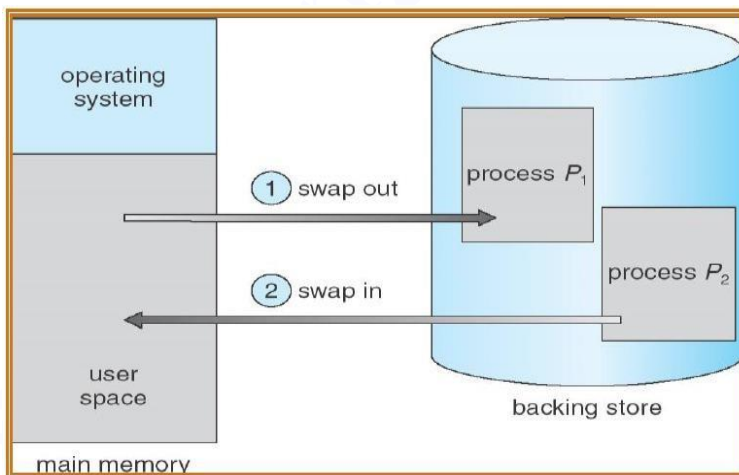
In computer storage, Belady’s anomaly is the phenomenon in which increasing the number of page frames results in an increase in the number of page faults for certain memory access patterns. This phenomenon is commonly experienced when using the first-in first-out (FIFO) page replacement algorithm.

45. Write about swapping. Let us assume the user process is of size 1MB and the backing store is a standard hard disk with a transfer rate of 5MBPS. Calculate the transfer rate. [E] (NOV/DEC 2017)

Swapping: A process can be swapped temporarily out of memory to a backing store (SWAP OUT) and then brought back into memory for continued execution (SWAP IN).

Let us assume the user process is of size 1MB & the backing store is a standard hard disk with a transfer rate of 5MBPS.

$$\begin{aligned} \text{Transfer time} &= 1000\text{KB}/5000\text{KB per second} \\ &= 1/5 \text{ sec} = 200\text{ms} \end{aligned}$$



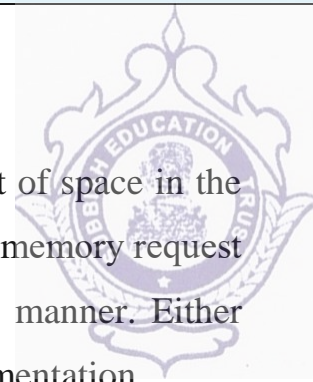
46. Consider the following page-reference string : [E] (NOV/DEC 2017)

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12.

How many page faults and page fault ratio would occur for the FIFO page replacement algorithm ? Assuming there is four frames.

The number of page faults = 12

The page fault ratio is 1:1



47. Define external fragmentation. (APR/MAY 2018)

External fragmentation occurs when there is a sufficient amount of space in the memory to satisfy the memory request of a process. But the process's memory request cannot be satisfied as the memory available is in a non-contiguous manner. Either first-fit or best-fit memory allocation strategy will cause external fragmentation.

48. What is the counting based page replacement algorithm? (APR/MAY 2018)

In this algorithm, we keep the counter of the number of reference that have been made to each page. In this 2 schemes are used :-

i) Least Frequency Used (**LFU**) Page Replacement Algorithm

It requires that the page with smallest count to be replaced.

ii) Most Frequency (**MFU**) Used Page Replacement Algorithm: It is based on the argument that the page with the smallest count was probably just brought in and has yet to be used.

49. Consider the following segmentation table.

Segment	Base	Length
0	219	600
1	2300	14
2	90	100
3	1327	580
4	1952	96

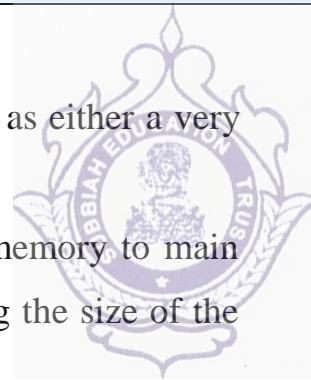
What are the physical addresses for the logical addresses 3400 and 0110? (APR/MAY 2019)

a. illegal reference; traps to operating system

b. $2300 + 0110 = 2410$

50. What is thrashing? and how to resolve this problem? (APR/MAY 2019)

In a virtual storage system (an operating system that manages its logical storage or memory in units called pages), thrashing is a condition in which excessive paging



operations are taking place. A system that is thrashing can be perceived as either a very slow system or one that has come to a halt.

One of the recommended ways to eliminate thrashing is to add more memory to main memory. Another way of resolving the issue of thrashing is by adjusting the size of the swap file.

51. When thrashing is used? (NOV/DEC 2021)

Refer pervious

52. What is demand paging? (NOV/DEC 2021)

Refer pervious

PART-B &C

1. Describe the hierarchical paging technique for structuring page tables. (8)
(MAY/JUNE 2013)
2. What is the cause for thrashing? How does the system detect thrashing? Once it detects, what can the system do to eliminate this problem? **(MAY/JUNE 2009)**
3. Write in detail about Segmentation.**(NOV/DEC 2009)**
4. Write in detail about Segmentation with Paging. **(APRIL/MAY2010)**
5. Explain the segmentation with paging implemented in OS/2 32-bit IBM system.
Describe the following algorithms: **(APRIL/MAY2010)**
 - a. First fit
 - b. Best Fit
 - c. Worst Fit
6. Explain how paging supports virtual memory. With a neat diagram explain how logical address is translated into physical address. **(NOV/DEC 2012)**
7. Explain the principles of segmented and paging implemented in memory with a diagram. **(NOV/DEC2013)**
8. Explain the segmentation with paging implemented in MULTICS system.
9. Explain the various page table structures in detail.
(APRIL/MAY2011)(MAY/JUNE 2014)



10. Write short notes on LRU, FIFO and clock replacement strategies?

(APRIL/MAY 2010, APRIL/MAY 2011)

11. Explain any four page replacement algorithms in detail? **(NOV/DEC 2009)**

(NOV/DEC 2013)

12.(i) Why page sizes are always powers of 2?

(ii) Consider the following segment table:

	Segment	Base	Length
i.	0	0219	600
ii.	1	2300	14
ii.	2	090	100
iv.	3	1327	580
v.	4	1952	96 (APR/MAY 2019)

13. What are the physical addresses for the following logical addresses?

0430

110

2500

400

4112 [E]

14. What is thrashing? Explain the working set model in detail. **(MAY/JUNE 2009)**

15. Given memory partitions of 100KB, 500KB, 200KB, 300KB and 600KB (in order),

how would each of the first-fit, best-fit and worst-fit algorithms place processes of 212KB, 417KB, 12KB and 426KB (in order)? Which algorithm makes the most efficient use of memory? **(NOV/DEC 2008)**

16.(i) Explain in briefly and compare, fixed and dynamic memory partitioning schemes.

(ii) Explain FIFO, optimal and LRU page replacement algorithms with an example reference Strings. Mention the merits and demerits of each of the above algorithms.

(NOV/DEC 2012)

17. Consider the following page reference string **(MAY/JUNE 2012) (APR/MAY 2015)**



1,2,3,4,2,1,5,6,2,1,3,7,6,3,2,1,3,6.

How many page faults would occur for the following replacement algorithms, assuming one, two, three and four frames?

- i) LRU replacement
- ii) FIFO replacement
- iii) Optimal replacement [E]

18.(i) Consider the following page reference string:

i. 2, 1, 0, 3, 4, 0, 0, 0, 2, 4, 2, 1, 0, 3, 2.

How many page faults would occur if the working set policy were used with a window size of 4? Show when each page fault would occur clearly. [E]

(ii) What is meant by thrashing? Discuss in detail. (MAY/JUNE 2013) **19.** Explain the concept of demand paging in detail with neat diagram. (MAY/JUNE 2014)

20. Why are translation look-aside buffers important? Explain the details stored in a TLB table entry? (MAY/JUNE 2014)

21. Consider the following page reference string :

1,2,3,4,2,1,5,6,1,2,3,7,6,3,2,1,2,3,6. How Many page faults would occur for the LRU, FIFO, LFU and optimal page replacement algorithms, assuming two and five frames? [E] (MAY/JUNE 2014)

22. Explain the concept of demand paging and the performance issue of short process Explain the issue of demand paging (NOV/DEC 2013)

23. With a neat sketch, explain how logical address is translated into physical address using paging mechanism? (APR/MAY 2015)

24. Write short notes on Memory Mapped Files. (APR/MAY 2015)

25.(i) Consider the following page reference string:

1,2,3,2,5,6,3,4,6,3,7,3,1,5,3,6,3,4,2,4,3,4,5,1



Indicate page faults and calculate total number of page faults and successful ratio for FIFO, optimal and LRU algorithms. Assume there are four frames and initially all the frames are empty. [E]

(ii) Explain the effect of thrashing. **(NOV/DEC 2015) 26.** Discuss the given memory management techniques with diagrams.

a. Partition Allocation Methods

b. Paging and Translation Look-aside Buffer. **(NOV/DEC 2015)**

27.(i) Describe a mechanism by which one segment could belong to the address space of two different processes.

(ii) Why are segmentation and paging sometimes combined into one scheme? Explain them in detail with example. **(MAY/JUNE 2016)**

28.(i) Under what circumstances do page faults occur? Describe the actions taken by the operating system when a page fault occurs.

(ii) Discuss situations in which the least frequently used (LFU) page replacement algorithm generates fewer page faults than the least recently used (LRU) page replacement algorithms. Also discuss under that circumstances the opposite holds good. **(MAY/JUNE 2016)**

29.(i) What is the copy-on-write feature, and under what circumstances is its use beneficial? Why hardware support is required to implement this feature? (8) **(NOV/DEC 2016)**

(ii) Consider a system that allocates pages of different sizes to its processes. What are the advantages of such a paging scheme? What modifications to the virtual memory system provide this functionality? (8) **(NOV/DEC 2016)**

30.(i) Explain the difference between internal and external fragmentation (8) **(NOV/DEC 2016)**

(ii) Discuss situations in which the most frequently used (MFU) page replacement algorithm generates fewer page faults than the least recently used (LRU) page-



replacement algorithm. Also discuss under what circumstances the opposite holds. (8)

(NOV/DEC 2016)

31. Discuss the given memory management techniques with diagrams

(i) Partition Allocation Methods

(ii) Paging and Translation Look-aside Buffer **(APR/MAY 2017)**

32.(i) Describe about free space management on I/O buffering and blocking (7)

(APR/MAY 2017)

(ii) Discuss the concept the buddy system allocation with neat sketch (6)

(APR/MAY 2017)

33. Draw the diagram of segmentation memory management scheme and explain its principle. (13) **(NOV/DEC 2017)**

34. When do page faults occur ? Consider the reference string : [E] **(NOV/DEC 2017)**

1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6.

How many page faults and page fault rate occur for the FIFO, LRU and optimal replacement algorithms, assuming one, two, three, four page frames?

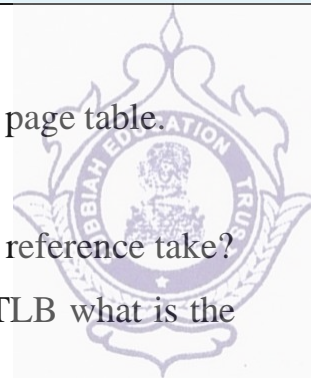
35. Explain why sharing a reentrant module is easier when segmentation is used than when pure paging is used with example. **(APR/MAY 2018)**

36. Discuss situation under which the most frequently used page replacement algorithm generates fewer page faults than the least recently used page replacement algorithm. Also discuss under which circumstances the opposite holds. **(APR/MAY 2018)**

37. (i) Consider a computer system with 16 bit logical address and 4 KB page size. The system supports up to 1 MB of physical memory. Assume that the actual process size is only 33 KB, page table base register contains 1000 and free frame list contains 13,1,9,7,5,3,1,2,4,6,8.

Construct the physical and logical memory structures page table of the corresponding process.

Find the physical address of 13,256 and another logical address with page number 2 and offset of 128.



Discuss about the possible valid invalid bit and possible protection bits in page table.

(ii) Consider a paging system with page table stored in memory.

1) If a memory reference takes 50 ns, how long does a paged memory reference take?

2) If we add TLB and 75% of all page table reference are found in TLB what is the effective memory reference time. **(13) (APR/MAY 2019)**

38. (i) Explain the global and local frame allocation algorithms and their pros and cons.

(3)

(ii) Consider the following page reference string.

1, 2 5,4 24, 5, 6 2s, 2-3, 74, 3, 21, 2-358

3) How many page faults would occur for the following replacement algorithms, assuming 1. and 3 free frames. Remember that all frames are initially empty so that first unique page request will all cost one fault each. LRU a FIFO, Optimal replacement, LFU and MFU. **(10) (APR/MAY 2019)**

27.(i) Compare paging with segmentation in terms of memory requirement by the address translation structure in order to convert virtual addresses to physical memory.

(ii) Explain in detail about page replacement algorithm with suitable example. **(NOV/DEC 2021)**

28.With a neat diagram. discuss about a mechanism of paging scheme. **(NOV/DEC 2021)**

UNIT IV STORAGE MANAGEMENT**10**

Mass Storage system – Disk Structure - Disk Scheduling and Management; File-System Interface - File concept - Access methods - Directory Structure - Directory organization - File system mounting - File Sharing and Protection; File System Implementation - File System Structure - Directory implementation - Allocation Methods - Free Space Management; I/O Systems – I/O Hardware, Application I/O interface, Kernel I/O subsystem.

PART-A**1) What is a file?**

A file is a named collection of related information that is recorded on secondary storage. A file contains either programs or data. A file has certain "structure" based on its type.

2) List the various file attributes. (APRIL/MAY 2011, NOV/DEC 2012)(MAY/JUNE 2014)(APRIL/MAY 2015)

A file has certain other attributes, which vary from one operating system to another, but typically consist of these:

- Identifier
- Name
- Type
- Location
- Size
- Protection
- Time
- Date
- User identification

3) What are the various file operations? (NOV/DEC 2012, APRIL/MAY 2015)

The six basic file operations are:

- Creating a file



- Writing a file
- Reading a file
- Repositioning within a file
- Deleting a file
- Truncating a file

4) What are all the information's associated with an open file?

Several pieces of information are associated with an open file which may be:

- File pointer
- File open count
- Disk location of the file
- Access rights

5) What are the different accessing methods of a file? (APRIL/MAY 2010)

The different types of accessing a file are:

Sequential access: Information in the file is accessed sequentially

Direct access: Information in the file can be accessed without any particular order.

Other access methods: Creating index for the file, indexed sequential access method (ISAM), etc.

6) What is Directory?

The device directory or simply known as directory records information-such as name, location, size, and type for all files on that particular partition. The directory can be viewed as a symbol table that translates file names into their directory entries.

7) What are the operations that can be performed on a directory?

The operations that can be performed on a directory are

- Search for a file
- Create a file
- Delete a file
- Rename a file



- List directory
- Traverse the file system

8) What are the most common schemes for defining the logical structure of a directory? (MAY/JUNE 2012)

The most common schemes for defining the logical structure of directory

- Single-Level Directory
- Two-level Directory
- Tree-Structured Directories
- Acyclic-Graph Directories
- General Graph Directory

9) Define UFD and MFD.

In the two-level directory structure, each user has her own user file directory (UFD). Each UFD has a similar structure, but lists only the files of a single user. When a job starts the system's master file directory (MFD) is searched. The MFD is indexed by the user name or account number, and each entry points to the UFD for that user.

10) What is a path name?

A pathname is the path from the root through all subdirectories to a specified file. In a two-level directory structure a user name and a file name define a path name.

11) What are the various layers of a file system?

The file system is composed of many different levels. Each level in the design uses the feature of the lower levels to create new features for use by higher levels.

- i. Application programs
- ii. Logical file system
- iii. File-organization module
- iv. Basic file system
- v. I/O control vi Devices



12) What are the structures used in file-system implementation? (APRIL/MAY 2011)

Several on-disk and in-memory structures are used to implement a file system

On-disk structure include

Boot control block

Partition block

Directory structure used to organize the files

File control block (FCB)

In-memory structure include

In-memory partition table

In-memory directory structure

System-wide open file table

Per-process open table

13) What are the functions of virtual file system (VFS)?

- It separates file-system-generic operations from their implementation defining a clean VFS interface. It allows transparent access to different types of file systems mounted locally.
- VFS is based on a file representation structure, called a vnode. It contains a numerical value for a network-wide unique file. The kernel maintains one vnode structure for each active file or directory.

14) Define seek time and latency time.

The time taken by the head to move to the appropriate cylinder or track is called seek time. Once the head is at right track, it must wait until the desired block rotates under the read-write head. This delay is latency time.

15) What are the allocation methods of a disk space?

Methods of allocating disk space which are widely in use are

- Contiguous allocation



- Linked allocation
- Indexed allocation

16) What are the advantages of Contiguous allocation?

The advantages are

- Supports direct access
- Supports sequential access
- Number of disk seeks is minimal.

17) What are the drawbacks of contiguous allocation of disk space?

The disadvantages are

- Suffers from external fragmentation.
- Suffers from internal fragmentation.
- Difficulty in finding space for a new file.
- File cannot be extended.
- Size of the file is to be declared in advance.

18) What are the disadvantages of linked allocation?

The disadvantages are

- Used only for sequential access of files.
- Direct access is not supported.
- Memory space required for the pointers.
- Reliability is compromised if the pointers are lost or damaged

19) What are the advantages of Indexed allocation?

The advantages are

- No external-fragmentation problems.
- Solves the size-declaration problems.
- Supports direct access.



20) How can the index blocks be implemented in the indexed allocation scheme?

The index block can be implemented as follows

- Linked scheme
- Multilevel scheme
- Combined scheme

21) What is garbage collection? (MAY /JUNE 2012)

Garbage Collection (GC) is a form of automatic memory management. The garbage collector, or just collector, attempts to reclaim garbage, or memory occupied by objects that are no longer in use by the program.

22) Mention the objectives of File Management System. (APR/MAY 2010)

The system that an operating system or program uses to organize and keep track of files. For example, a hierarchical file system is one that uses directories to organize files into a tree structure.

23) What is the content of a typical file control block? (APR/MAY 2011, APR/MAY 2010)

File permissions
File dates (create, access, write)
File owner, group, ACL
File size
File data blocks

File Control Block (FCB) is a file system structure in which the state of an open file is maintained.

24) What are the two types of system directories? (MAY/JUNE 2012) [R]

Device directory, describing physical properties of files.



File directory, giving logical properties of the files.

25) What is meant by polling?(MAY/JUNE 2014)

Polling is the process where the computer waits for an external device to check for its readiness. The computer does not do anything else than checking the status of the device .Polling is often used with low-level hardware. Example: when a printer connected via a parallel port the computer waits until the next character has been received by the printer. These processes can be as minute as only reading 1 Byte. Polling is the continuous (or frequent) checking by a controlling device or process of other devices, processes, queues, etc.

26) State any three disadvantages of placing functionality in a device controller, rather than in the kernel. (MAY/JUNE 2014)

Three advantages:-

- a. Bugs are less likely to cause an operating system crash.
- b. Performance can be improved by utilizing dedicated hardware and hard-coded algorithms.

The kernel is simplified by moving algorithms out of it.

Three disadvantages:

- a. Bugs are harder to fix - a new firmware version or new hardware is needed
- b. Improving algorithms likewise require a hardware update rather than just kernel or device driver update
- c. Embedded algorithms could conflict with application's use of the device, causing decreased performance.

27) How free-space is managed using bit vector implementation?

The free-space list is implemented as a bit map or bit vector. Each block is represented by 1 bit. If the block is free, the bit is 1; if the block is allocated, the bit is 0.

28) List the attributes of a file(MAY/JUNE 2014)

Name, Identifier, Type, Location, Size, Protection, Time, Date and User authentication.



29) What are the information contained in a boot control block and partition control block? (MAY/JUNE 2014)

Boot control block:

Contain information needed by the system to boot an operating from that partition. If the disk does not contain an operating system, this block can be empty. It is typically the first block of a partition. In UFS, this is called the boot block.

Partition Control block:

Contains partition details, such as number of blocks in the partition, size of the blocks, free block count and free block pointers, and free FCB count and FCB pointers.

30) Define buffering.

A buffer is a memory area that stores data while they are transferred between two devices or between a device and an application. Buffering is done for three reasons

- a. To cope with a speed mismatch between the producer and consumer of a data stream
- b. To adapt between devices that have different data transfer sizes
- c. To support copy semantics for application I/O

31) Define caching.

A cache is a region of fast memory that holds copies of data. Access to the cached copy is more efficient than access to the original. Caching and buffering are distinct functions, but sometimes a region of memory can be used for both purposes.

32) Define spooling.

A spool is a buffer that holds output for a device, such as printer, that cannot accept interleaved data streams. When an application finishes printing, the spooling system queues the corresponding spool file for output to the printer. The spooling system copies the queued spool files to the printer one at a time.

33) Define rotational latency and disk bandwidth. (NOV/DEC 2010, MAY/JUNE 2013)

Rotational latency is the additional time waiting for the disk to rotate the desired sector to the disk head.



Disk bandwidth is the total number of bytes transferred, divided by the time between the first request for service and the completion of the last transfer.

34) What are the various disk-scheduling algorithms?

The various disk-scheduling algorithms are

- First Come First Served Scheduling
- Shortest Seek Time First Scheduling
- SCAN Scheduling
- C-SCAN Scheduling

35) What is the need for disk scheduling? (NOV/DEC 2012)

In operating systems, seek time is very important. Since all device requests are linked in queues, the seek time is increased causing the system to slow down.

Disk Scheduling Algorithms are used **to reduce the total seek time** of any request.

36) What is low-level formatting?

Before a disk can store data, it must be divided into sectors that the disk controller can read and write. This process is called low-level formatting or physical formatting. Low-level formatting fills the disk with a special data structure for each sector. The data structure for a sector consists of a header, a data area, and a trailer.

37) What is the use of boot block?

For a computer to start running when powered up or rebooted it needs to have an initial program to run. This bootstrap program tends to be simple. It finds the operating system on the disk loads that kernel into memory and jumps to an initial address to begin the operating system execution. The full bootstrap program is stored in a partition called the boot blocks, at fixed location on the disk. A disk that has boot partition is called boot disk or system disk.

38) What is sector sparing?

Low-level formatting also sets aside spare sectors not visible to the operating system. The controller can be told to replace each bad sector logically with one of the



spare sectors. This scheme is known as sector sparing or forwarding.

39) What is seek time? (MAY /JUNE 2012)

Seek time: the time to position heads over a cylinder (~8 msec on average).

40) What are storage area networks? (April/May 2011)

A storage area network (SAN) is a dedicated network that provides access to consolidated, block level data storage. SANs are primarily used to make storage devices, such as disk arrays, tape libraries, and optical jukeboxes, accessible to servers so that the devices appear like locally attached devices to the operating system.

41) Write a brief note on RAID. (MAY/JUNE 2013)

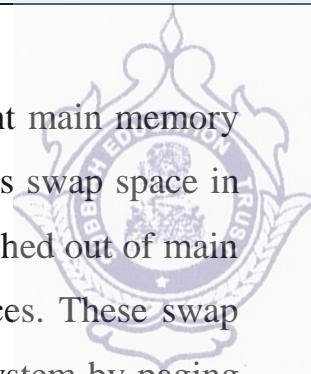
RAID (redundant array of independent disks; originally redundant array of inexpensive disks) is a way of storing the same data in different places (thus, redundantly) on multiple hard disks. By placing data on multiple disks, I/O (input/output) operations can overlap in a balanced way, improving performance. Since multiple disks increase the mean time between failures (MTBF), storing data redundantly also increases fault tolerance.

42) What Characteristics determine the disk access speed? (MAY /JUNE 2012)

- Seek time
- Rotational latency
- Command processing time
- Settle time

43) Give the importance of Swap space Management. (NOV/DEC 2012, APR/MAY2010, NOV/DEC 2010)

Swap-space management: Swap-space management is low- level task of the operating system. The main goal for the design and implementation of swap space is to provide the best throughput for the virtual memory system.



Swap-space use: The operating system needs to release sufficient main memory to bring in a process that is ready to execute. Operating system uses this swap space in various ways. Paging systems may simply store pages that have been pushed out of main memory. UNIX operating system allows the use of multiple swap spaces. These swap space are usually put on separate disks, so the load placed on the I/O system by paging and swapping can be spread over the systems I/O devices.

Swap-space location: Swap space can reside in two places:

1. Separate disk partition
2. Normal file system

44) Write three basic functions which are provided by the hardware clocks and timers. (APRIL/MAY 2011)

- OSTickInit()
- OSTimeSet()
- OSTimeGet()

45) What are the advantages of Linked allocation?

The advantages are

- No external fragmentation.
- Size of the file does not need to be declared.

46) Define FAT (NOV/DEC 2014)

FAT is a much older file-system format that is understood by many systems besides Windows, such as the software running on cameras. A disadvantage is that the FAT file system does not restrict file access to authorized users. The only solution for securing data with FAT is to run an application to encrypt the data before storing it on the file system.

47) What is Relative block number? (NOV/DEC 2014)

Relative block number is an index relative to the beginning of a file. Thus the 1st relative block of the file is 0, the next is 1, and so on.



48) What is double buffering? (NOV/DEC 2014)

OS can use various kinds of buffering:

1. Single buffering — OS assigns a system buffer to the user request
2. Double buffering — process consumes from one buffer while system fills the next
3. Circular buffers — most useful for burst I/O

49) Draw the diagram for interrupt driven I/O cycle? (NOV/DEC 2014)

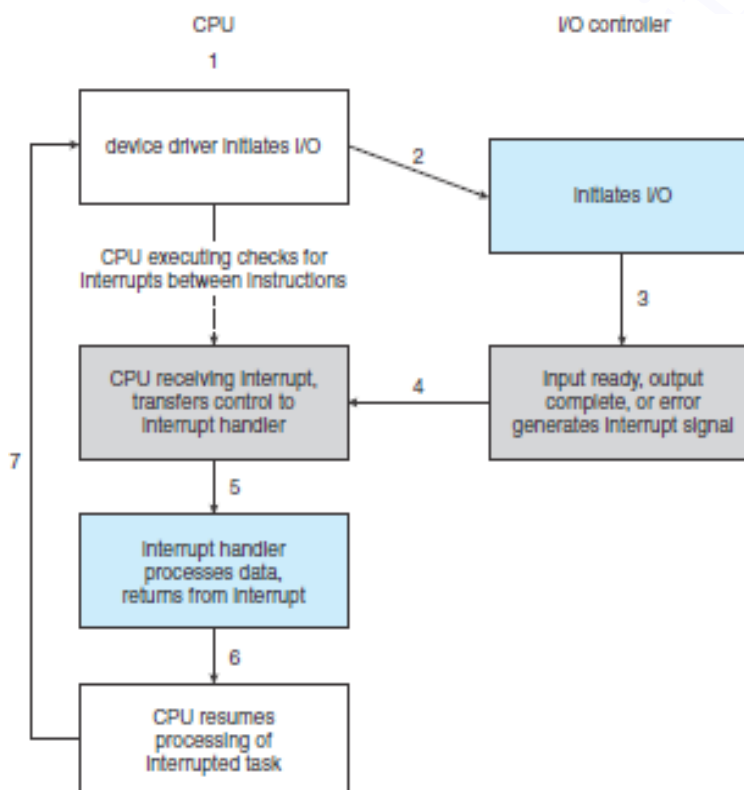
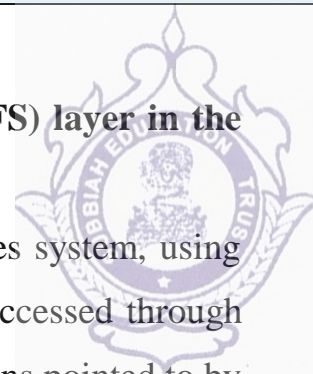


Figure 13.3 Interrupt-driven I/O cycle.

50) What is HSM? Where it is used?

Hierarchical storage management (HSM) is a data storage technique, which automatically moves data between high-cost and low-cost storage media. HSM systems exist because high-speed storage devices, such as hard disk drive arrays, are more expensive (per byte stored) than slower devices, such as optical discs and magnetic tape drives.



51) Identify the two important functions of Virtual File System (VFS) layer in the concept of file system implementation. (Nov/Dec 2015)

Linux VFS provides a set of common functionalities for each file system, using function pointers accessed through a table. The same functionality is accessed through the same table position for all file system types, though the actual functions pointed to by the pointers may be file system-specific. Common operations provided include `open()`, `read()`, `write()`, and `mmap()`.

52) How does DMA increase system concurrency? (May/June 2016)

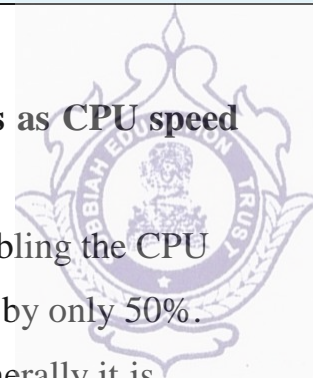
DMA increases system concurrency by allowing the CPU to perform tasks while the DMA system transfers data via the system and memory buses. Hardware design is complicated because the DMA controller must be integrated into the system and the system must allow the DMA controller to be a bus master.

53) Why rotational latency is usually not considered in disk scheduling? (May/June 2016)

Most disks do not export their rotational position information to the host. Even if they did, the time for this information to reach the scheduler would be subject to imprecision and the time consumed by the scheduler is variable, so the rotational position information would become incorrect. Further, the disk requests are usually given in terms of logical block numbers, and the mapping between logical blocks and physical locations is very complex.

54) Define C-SCAN scheduling (NOV/DEC 2016)

In the **C-Scan** all the processes are arranged by using some circular list. Circular list is that in which there is no start and end point of the list means the end of the list is the starting point of the list. In the C-Scan scheduling the CPU will search for the process from start to end and if an end has found then this again start from the starting process.



55) Why it is important to scale up system –bus and device speeds as CPU speed increases? (NOV/DEC 2016)

Consider a system which performs 50% I/O and 50% computers. Doubling the CPU Performance on this system would increase the total system performance by only 50%. Doubling both system aspects would increase performance by 100%. Generally, it is important to remove the current system bottleneck and to increase overall system performance rather than blindly increasing the performance of individual system components.

56) Suppose that the disk rotates at 7200 rpm. What is the average rotational latency of the disk drive? (APR/MAY 2017) [E]

Average disk access time = average seek time + average rotational delay + transfer time + controller overhead.

For 7200 RPM

$$= 12 \text{ ms} + 0.5 \text{ rotation} / 7200 \text{ RPM} + 0.5 \text{KB} / 5 \text{MB/sec} + 2 \text{ms} = 12 \text{ms} + 4.15 \text{ms} + 0.1 \text{ms} + 2 \text{ms} = 18.25 \text{ms}$$

57) Differentiate between file and directory (NOV/DEC 2016)

Directory is a collection of files and folders. difference between directory and File : A file is any kind of computer document and a directory is a computer document folder or filing cabinet. Directory is a collection of a the folders and files.

58) Suppose that the disk rotates at 7200 rpm [E] (NOV/DEC 2017)

a) What is the average rotational latency of the disk drive ?

b) Identify seek distance can be covered in the time ?

a) 7200 rpm gives 120 rotations per second.

Therefore Full rotation takes 8.33 ms and Average rotational latency takes 4.167ms.

$$b) t = 4.167 \text{ms}$$

$$t = 0.7561 + 0.24L$$

$$4.167 = 0.756 + 0.2439L$$

$$L = 195.58$$



Therefore We can seek over 195 tracks(about 4% of the disk) during an average rotational latency

59) Enlist different types of directory structure. (NOV/DEC 2017)

There are many types of directory structure in Operating System. They are as follows

- :-
- 1) Single Level Directory
 - 2) Two Level Directory
 - 3) Tree Structured Directory
 - 4) Acyclic Graph Directory
 - 5) General Graph Directory

60) State the typical bad-sector transactions. (APR/MAY 2018)

Typical bad sector transactions might be as follows :

- The OS tries to read logical block.
- The controller calculates the ECC and finds that the sector is bad. It reports this finding to the OS.
- The next time that the system is rebooted ,a special command is run to tell the SCSI controller to replace the bad sector with a spare.
- After that, whenever the system requests logical block, the request is translated into the replacement sector's address by the controller.

61) What is the advantage of bit vector approach in free space management ?

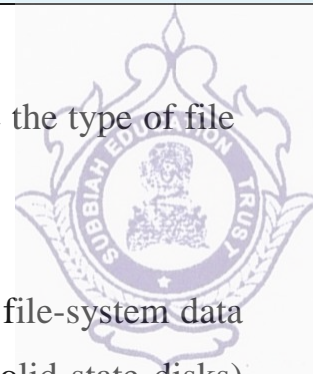
(APR/MAY 2018)

Advantages of bit vector approach:

- Relatively simple
- Efficient to find the first free blocks or n consecutive free blocks on the disk

62. Write short notes on file system mounting. (APR/MAY 2019)

The mount procedure is straightforward. The operating system is given the name of the device and the mount point—the location within the file structure where the file system is to be attached. Some operating systems require that a file system type be



provided, while others inspect the structures of the device and determine the type of file system. Typically, a mount point is an empty directory.

63. What is SSD? (APR/MAY 2019)

The highest level, the operating system may maintain a cache of file-system data in main memory. In addition, electronic RAM disks (also known as solid-state disks) may be used for high-speed storage that is accessed through the file-system interface.

64. Enlist different types of file directory structure. (NOV/DEC 2021)

65. Is FAT file system advantageous? Justify. (NOV/DEC 2021)

PART-B&C

- 1) Explain the different disk scheduling algorithms with examples. (APRIL/MAY 2010, MAY/JUNE 2012, APRIL/MAY 2011, MAY/JUNE 2013) (MAY/JUNE 2014)
- 2) Explain and compare FCFS, SSTF, C-SCAN and C-LOOK disk scheduling algorithms with examples. (NOV/DEC 2012)
- 3) Write short notes on disk management. (NOV/DEC 2009)
- 4) Write short notes on file system in Linux. (NOV/DEC 2009) (NOV/DEC 2014)
- 5) Write an elaborate note on RAID and RAID Levels. (APRIL/MAY 2010, MAY/JUNE 2012, NOV/DEC 2012, MAY/JUNE 2013)
- 6) Explain the services provided by Kernel I/O subsystem. (APRIL/MAY 2010, APRIL/MAY 2011, NOV/DEC 2012, MAY/JUNE 2013)
- 7) Consider the following I/O scenarios on a single-user PC.
 - A mouse used with a graphical user interface.
 - A tape drive on a multitasking operating system (assume no device preallocation is available)
 - A disk drive containing user files.
 - A graphics card with direct bus connection, accessible through memory-mapped I/O
 For each of these I/O scenarios, would you design the operating system to use



buffering, Spooling, caching, or a combination? Would you use polled I/O, or interruption driven I/O?

8) Write short notes on

1. File types
2. File attributes
3. File operations

9) Explain the file allocation methods. **(APRIL/MAY 2010)**

10) Explain the role of Access Matrix for protection in files. **(APRIL/MAY 2010)**

11) Write in detail the security measures taken in file system?

1. Write short notes on file system mounting
2. Write in detail about the various file organizations

12) Explain the allocation of frames in detail.

13) Explain directory subsystem **(APRIL/MAY 2011)**

- (i) Explain Linked File Allocation method (6)
- (ii) Explain the issues in designing a file system. (8)
- (ii) Explain the various file directory structures. (8) **(NOV/DEC 2012)**

14) (i) Explain the different file access methods in detail. (8) **(MAY/JUNE 2014)**

(ii) Describe the two level and acyclic graph schemes for defining the logical structure of a directory. **(MAY/JUNE 2013)**

15) Explain the Linked list and indexed file allocation methods with neat diagram.

Mention their advantages and disadvantages. (8) **(MAY/JUNE 2013)**

16) What are the most common schemes for defining the logical structure of a directory? **(MAY/JUNE 2014)**

17) Write a brief note on the steps involved in DMA transfer **(MAY/JUNE 2014)**

18) Explain the data structures supported by kernel I/O system **(MAY/JUNE 2014)**

19) Write a brief note on tertiary storage devices **(MAY/JUNE 2014)**

20) Explain different directory implementation methods **(NOV/DEC 2013)**



- 21) Why disk scheduling is necessary? Explain the different seek optimization techniques (NOV/DEC 2013)
- 22) Explain about RAID structure in disk management with various RAID levels of organization in detail (Apr/May 2015)
- 23) Briefly discuss about the various directory structures. (APR/MAY 2015)
- 24) Compare the functionalities of FCFS, SSTF, SCAN AND C-LOOK disk scheduling algorithms with an example for each. (APR/MAY 2015)
- 25) Write short notes on free space management. (APR/MAY 2015, MAY/JUNE 2013)
- 26) (i) Discuss the functions of files and file implementation.
(ii) Explain free space management with neat example. (Nov/Dec 2015)
- 27) On a disk with 200 cylinders, numbered 0 to 199, compute the number of tracks the disk arm must move to satisfy the entire request in the disk queue. Assume the last request received was at track 100. The queue in FIFO order contains requests for the following tracks. 55, 58, 39, 18, 90, 160, 150, 38, 184. Perform the computation to find the seek time for the following disk scheduling algorithms.
- (i) FCFS
 - (ii) SSTF
 - (iii) SCAN
 - (iv) C-SCAN
 - (v) LOOK [E] (Nov/Dec 2015)
- 28) (i) Why it is important to balance file-system I/O among the disks and controllers on a system in a multitasking environment?
(ii) Discuss the advantages and disadvantages of supporting links to files that cross mount points. (MAY/JUNE 2016)
- 29) (i) Explain why logging metadata updates ensures recovery of a file system after a file-system crash.



- (iii)** Could a RAID level 1 organization achieve better performance for read requests than a RAID level 0 organizations? If so, how? **(MAY/JUNE 2016)**
- 30)** (i) Describe some advantages and disadvantages of using SSDs as a caching tier and as a disk-drive replacement compared with using only magnetic disks. (8) **(NOV/DEC 2016)**
- (ii) Discuss how performance optimizations for file systems might result in difficulties in maintaining the consistency of the systems in the event of computer crashes (8) **(NOV/DEC 2016)**
- 31)** (i) Distinguish between a STREAMS driver and a STREAMS module (8) **(NOV/DEC 2016)**
- (ii) Could a RAID level 1 organization achieve better performance for read requests than a RAID level 0 organization? If so, how? Explain. (8) **(NOV/DEC 2016)**
- 32)** (i) Discuss about the various file access methods (7) **(APR/MAY 2017)**
- (ii) With neat sketch explain about the Directory structure, File Sharing (6) **(APR/MAY 2017)**
- 33)** (i) Explain about kernel I/O subsystem and transforming I/O to hardware operations (7) **(APR/MAY 2017)**
- 34)** (ii) On a disk with 1000 cylinders, numbers 0 to 999, Compute the number of tracks the disk arm must move to satisfy the entire request in the disk queue. Assume the last received was at track 345 and the head is moving towards track 0. The queue in FIFO order contains requests for the following tracks. 123,874,692,475,105 and 376. Find the seek length for the following scheduling algorithm. (1) SSTF (2) LOOK (3) CSCAN **(APR/MAY 2017)**
- 35)** (i) In a variable partition scheme the operating system has to keep track of allocated and free space. Suggest a means of achieving this. Describe the effects of new allocations and process terminations in your suggested scheme. (5) **(NOV/DEC 2017)**



(ii) What are different allocation methods in disk storage? Explain with neat sketch. (8)

36) Consider a disk queue with requests for I/O to blocks on cylinders

93, 183, 37, 122, 14, 124, 65, 67

If the disk head starts at 53, then find out the total head movement with respect to FCFS, SSTF, SCAN, C-SCAN and LOOK scheduling. (13) [E] (NOV/DEC 2017)

37) What are the various disk space allocation methods? Explain any two in detail.

(APR/MAY 2018)

38) State and explain the FCFS, SSTF and SCAN disk scheduling with examples.

(APR/MAY 2018)

39. Suppose that a disk drive has 5000 cylinders, numbered 0 through 4999. The drive is serving a request at cylinder 143. The queue of pending requests, in FIFO order is 86, 1470, 913, 1774, 948, 1509, 1022, 1750, 130. Starting from the head position, what is the total distance (in cylinders) that the disk arm moves to satisfy all the pending requests for each of the following disk-scheduling algorithms? FCFS, SSTF, SCAN, LOOK, C-SCAN, C-LOOK. Explain the pros and cons of all disk scheduling algorithms. (13)

(APR/MAY 2019)

40. (i) Explain in detail the various allocation methods with their pros and cons. (8)

(ii) Brief the various procedures that need to be followed in disk management. (5)

(APR/MAY 2019)

41.(i) What do you mean by directory structure? Also discuss Tree-Structure Directories and Acyclic-Graph Directories. (7) (NOV/DEC 2021)

(ii) Describe in detail about file system implementation and file allocation method (6).

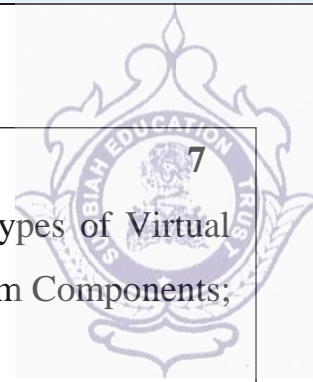
42. Suppose that a disk drive has 200 cylinders, numbered 0 to 199. The work queue is: 23, 89, 132, 42, 187. Determine the total distance for the following disk scheduling algorithms

(i) SCAN (ii) LOOK (iii) C-SCAN (iv) C-LOOK



Work Queue : 23,89,132,42,187

- There are 200 cylinder numbered from 0 – 199
- The disk head starts at number 100. (**NOV/DEC 2021**)



UNIT V VIRTUAL MACHINES AND MOBILE OS

7

Virtual Machines – History, Benefits and Features, Building Blocks, Types of Virtual Machines and their Implementations, Virtualization and Operating-System Components; Mobile OS - iOS and Android.

PART-A

1) What is Linux distribution?

A Linux distribution includes all the standard components of the Linux system, plus a set of administrative tools to simplify the initial installation and subsequent upgrading of Linux and manage installation and removal of other packages on the system.

2) What is the use of User mode?

Under Linux, no user code is built into the kernel. Any operating-system-support code that does not need to run in kernel mode is placed into the system libraries and runs in user mode.

3) What are the components of kernel mode

The module support under Linux has four components:

1. The **module-management** system allows modules to be loaded into memory and to communicate with the rest of the kernel.
2. The **module loader and unloader**, which are user-mode utilities, work with the module-management system to load a module into memory.
3. The **driver-registration system** allows modules to tell the rest of the kernel that a new driver has become available.
4. A **conflict-resolution mechanism** allows different device drivers to reserve hardware resources and to protect those resources from accidental use by another driver.

4) What is process Identity?

Each process has a unique identifier. The PID is used to specify the process to the operating system when an application makes a system call to signal, modify, or wait for



the process. Additional identifiers associate the process with a process group (typically, a tree of processes forked by a single user command and login session).

5) Define DNS

The Domain Name System (DNS) provides host-name-to-network-address translations for the entire Internet. Before DNS became widespread, files containing the same information were sent via e-mail or ftp between all networked hosts.

6) What is virtualization?

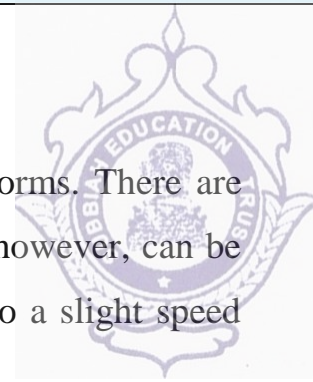
Virtualization, in computing, refers to the act of creating a virtual (rather than actual) version of something, including but not limited to a virtual computer hardware platform, operating system (OS), storage device, or computer network resources.

7) What is pluggable authentication modules

The pluggable authentication modules (PAM) system is based on a shared library that can be used by any system component that needs to authenticate users. An implementation of this system is available under Linux. PAM allows authentication modules to be loaded on demand as specified in a system-wide configuration file. If a new authentication mechanism is added at a later date, it can be added to the configuration file, and all system components will immediately be able to take advantage of it. PAM modules can specify authentication methods, account restrictions, session setup functions, and password-changing functions (so that, when users change their passwords, all the necessary authentication mechanisms can be updated at once).

8) What is the use of firewall manager

The firewall manager maintains a number of separate firewall chains and allows a skbuff to be matched against any chain. Chains are reserved for separate purposes: one is used for forwarded packets, one for packets being input to this host, and one for data generated at this host. Each chain is held as an ordered list of rules, where a rule specifies one of a number of possible firewall-decision functions plus some arbitrary data for matching purposes.

**9) Do FAT file system is advantageous? Why?**

FAT File System is best for cross-compatibility with other platforms. There are NTFS file system drivers for Linux, but not really for Windows. FAT, however, can be read more or less transparently by both operating systems. There is also a slight speed gain in FAT.

10) What is the responsibility of kernel in Linux operating system?

Kernel is the core part of Linux. It is responsible for all major activities of this operating system. It consists of various modules and it interacts directly with the underlying hardware. Kernel provides the required abstraction to hide low level hardware details to system or application programs.

11) Why Virtualization is required? (Nov/Dec 2015)

Virtualization reduces the number of physical servers, reducing the energy required to power and cool them. Save time. With fewer servers, you can spend less time on the manual tasks required for server maintenance. It's also much faster to deploy a virtual machine than it is to deploy a new physical server.

12) Enumerate the requirements for Linux system administrator. Brief any one.

(Nov/Dec 2015)

1. While specific knowledge is a boon, most hiring managers require that you possess basic knowledge about all aspects of Linux. For example, a little knowledge about Solaris, BSD or various flavors of Linux never hurt anyone!
2. Knowledge in at least one of the upper tier scripting language is a must. You have options before you, for instance, Python, Perl, Ruby or more, but you need to make yourself proficient in at least one of them.
3. Experience is welcome, but you at least need to have some hands-on experience of system management, system setup and managing Linux or Solaris based servers as well as configuring them.



4. Knowledge in shell programming and architecture is valued very much in the job market. If you know Bourne or Korn well, you can even score a high-paying salary with minimal experience.

5. Storage technologies like FC, NFS or iSCSI is great, while knowledge regarding backup technologies is a must for a system administrator.

13) State the components of a Linux System? (May/Jun 2016)

- **Kernel:** The kernel is responsible for maintaining all the important abstractions of the operating system, including such things as virtual memory and processes.
- **System libraries:** The system libraries define a standard set of functions through which applications can interact with the kernel. These functions implement much of the operating-system functionality that does not need the full privileges of kernel code.
- **System utilities:** The system utilities are programs that perform individual, specialized management tasks. Some system utilities are invoked just once to initialize and configure some aspect of the system.

14) Define the function of Caching-only servers. (May/Jun 2016)

All DNS servers cache answers to queries they receive from outside their own zone of authority. A cache-only DNS server is not authoritative for any zone. Related Topics: DNS root servers: Root servers are critical to the function of a DNS server that is directly connected to the Internet.

15) What is virtualization? (NOV/DEC 2016)

In computing, virtualization refers to the act of creating a virtual (rather than actual) version of something, including virtual computer hardware platforms, storage devices, and computer network resources.

16) What scheduling algorithm is used in linux operating system to schedule jobs? (NOV/DEC 2016)

The Completely Fair Scheduler (CFS) Algorithm is used in Linux operating system to schedule jobs.

**17) Mention any two features of Linux file systems. (APR/MAY 2017)**

- ✓ Portable
- ✓ Open Source
- ✓ Multi user
- ✓ Multi Programming

18) Enlist the advantages of using kernel modules in Linux (APR/MAY 2017)

- Loading and unloading a module is much more flexible and faster than recompiling a kernel and rebooting.
- You can try different options each time you load a module. Most drivers that handle hardware will take options for I/O addresses, IRQ or DMA numbers, plus more esoteric options like full or half duplex. When you have problems getting a card to run correctly, the ability to try different options can save hours.
- Makes it easier to maintain multiple machines on a single kernel base.

17) List the advantages of Linux OS.(NOV/DEC 2017)**Advantages of Linux OS:**

Low cost

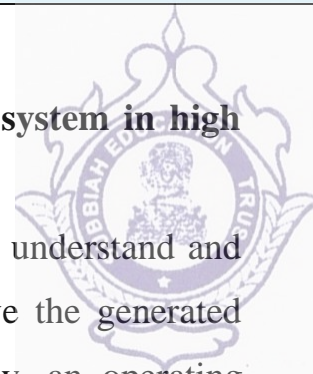
Stability

Performance

Security, Flexibility, Compatibility

18) Write the purpose of using virtualization. (NOV/DEC 2017)

Operating system virtualization provides application-transparent virtualization to users by decoupling applications from the OS. The OS virtualization technique offers granular control at the application level by facilitating the transparent migration of individual applications. OS virtualization can also be used to migrate critical applications to another running operating system instance.



19) List the advantages and disadvantage of writing an operating system in high level language such as C. (APR/MAY 2018)

The code can be written faster, is more compact, and is easier to understand and debug. In addition, improvements in compiler technology will improve the generated code for the entire operating system by simple recompilation. Finally, an operating system is far easier to port to move to some other hardware if it is written in a higher-level language

20) What is handle? How does a process obtain a handle? (APR/MAY 2018)

The handle is an integer value in the operating system assigned to the associated process when the process was started. The system uses this handle to keep track of process attributes. An application can obtain a handle to a process that can be used as a parameter to many process-information and control functions.

21. Write short notes on driver registration in Linux. (APR/MAY 2019)

The **driver registration** allows modules to tell the rest of the kernel that a new driver has become available.

22. List out the methods used to recover from the deadlock. (APR/MAY 2019)

We can use a protocol to prevent or avoid deadlocks, ensuring that the system will *never* enter a deadlocked state.

- We can allow the system to enter a deadlocked state, detect it, and recover.
- We can ignore the problem altogether and pretend that deadlocks never occur in the system.

44. What are the Components of a Linux System?

Every OS has component parts, and the Linux OS also has the following components parts:

- Boot loader
- OS Kernel
- Background services
- OS Shell



- Graphics server
- Desktop environment
- Applications.

45. Which layer of iOS contains fundamental system services for apps?

The **Core Services layer** contains the fundamental system services that all applications use, The Core Services layer provides an abstraction over the services provided in the Core OS layer. It provides fundamental access to iOS services and consists of the following components:

- Collections
- Address Book
- Networking
- File Access
- SQLite
- Core Location
- Net Services
- Threading
- Preferences
- URL Utilities

PART-B & C

1. Explain in detail about the concepts of Linux system.
2. Explain in detail about virtualization
3. Explain in detail about setting up a Linux mainframe server
4. Explain in detail about Linux host and adding guest OS
5. Explain the significance and steps involved in setting up Xen, VMware software's on Linux host for successful virtualization in detail. **May/June 2015**
6. Briefly discuss about the requirements to become a Linux system administrator.

May/June 2015

7. Discuss about the steps involved in the installation of a Linux Multifunction server.

May/June 2015



8. Write a short note on Linux Network Services. **May/June 2015**

9. Write about LINUX architecture and LINUX kernel with neat sketch. (Nov/Dec 2015)

10. Explain in detail about LINUX multifunction server, DNS VMware on LINUX host. (Nov/Dec 2015) 11.(i)

Why is live migration possible in virtual environments but much less possible for anative operating system?

(ii) What are the primary goals of the conflict-resolution mechanism used by the Linux kernel for loading kernel modules. (May/June 2016)

12. Explain in step-by-step procedure for setting up a Linux multifunction server. (May/June 2016) 13.(i)

Discuss three advantages of dynamic (shared) linkage of libraries compared with static linkage. Describe two cases in which static linkage is preferable (8) (NOV/DEC 2016)

(ii) Explain the step by step procedure for setting up a local network services. (8) (NOV/DEC 2016)

14. Explain the concepts of domain name system and multifunction server (13) (APR/MAY 2017)

15. Write short notes on Linux kernel and virtualization with neat sketch. (13) (APR/MAY 2017)

16. What do you mean by term synchronization? What is semaphore? Explain how semaphore can be used as synchronization tool. Consider a coke machine that has 10 slots. The producer is the delivery person and the consumer is the student using the machine. It uses the following three semaphores:

Semaphore mutex

Semaphore fullBuffer /*Number of filled slots */

Semaphore emptyBuffer /* Number of empty slots */



- (i) Write pseudo code for delivery person() and student ()
- (ii) What will be the initial values of the semaphores?
- (iii) Write a solution that guarantees the mutual exclusion and has no deadlocks (15)

(APR/MAY 2017)

17. What is deadlock? What are the necessary conditions for deadlock to occur? Explain the deadlock prevention method of handling deadlock.

Consider the following information about resources in a system.

- (i) There are two classes of allocatable resource labeled R1 and R2
- (ii) There are two instances of each resource
- (iii) There are four processes labeled p1 through p4
- (iv) There are some resource instances already allocated to processes as follows:
 - One instance of R1 held by p2, another held by p3
 - One instance of R2 held of P1, another held by p4
- (v) Some processes have required additional resources, as follows:
 - P1 wants one instance of R1
 - P3 wants one instance of R2

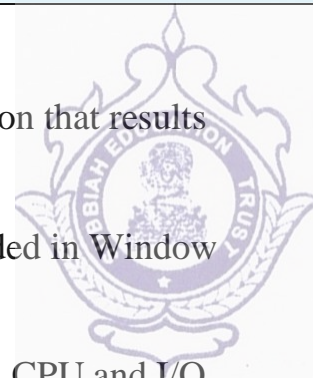
1. Draw the resource allocation graph for this system
2. What is the state (runnable, waiting) of each process ? For each process that is waiting indicate what it is waiting for.

(iv) Is this system deadlocked? If so, state which processes are involved. If not, give an execution sequence that eventually ends, showing resource acquisition and release at each step. **(APR/MAY 2017)**

18. (i) Explain the components of Linux system with neat sketch. (6) [U (ii) Write the various system administrator roles in LINUX OS. (7)] **(NOV/DEC 2017)**

19. (i) How to install and configuring network services in LINUX. (9)

(ii) Describe the benefits of virtualization in LINUX OS. (4) **(NOV/DEC 2017)**



20. i) Under what circumstance would an user process request an operation that results in the allocation of a demand-zero memory region.(8) **(APR/MAY 2018)**

ii) Describe an useful application of the no-access page facility provided in Window XP.(5) **(APR/MAY 2018)**

21. i) What optimization were used to minimize the discrepancy between CPU and I/O speeds on early computer systems. (8)

ii) What manages cache in Windows XP ? How is cache managed?(5)
(APR/MAY 2018)

22. Discuss the process and memory management in Linux. (13) **(APR/MAY 2019)**

23. Explain the architecture of iOS. Discuss the media and service layers clearly. (13)
(APR/MAY 2019)

24. (i) Explain in details about how process is managed and scheduled in linux?

(ii) Discuss about Inter Process Communication (IPC) in linux. **(NOV/DEC 2021)**

46. With frame work explain the working function of android operating system architecture. Compare the feature of iOS and android. (13) **(NOV/DEC 2021)**
