I. K. Gujral Punjab Technical University

Refresher

Subject : Operating System

B.tech CSE 4th SEM

@AdarshKumar 😄

~ Syllabus ~

1. Introduction

Concept of Operating Systems, Generations of Operating Systems, Types of Operating Systems, OS Services, System Calls, Structure of an OS-Layered, Monolithic, Microkernel Operating Systems, Concept of Virtual Machine. Case study on UNIX and Windows Operating System.

2. Processes

Definition, Process Relationship, Different states of a Process, Process State transitions, Process Control Block (PCB) Context swtiching.

Thread : Definition, Various states, Benefits of threads, Types of threads, Concept of multithreads.

Process Scheduling : Foundation and Scheduling objectives, Types of Schedulers, Scheduling criteria, CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time ; Scheduling algorithms : Pre-emptive and Non-pre-emptive, FCFS, SJF, RR ; Multiprocessor scheduling : Real Time Scheduling : RM and EDF.

3. Inter-process Communication

Critical Section, Race Conditions, Mutual Exclusion, Hardware Solution, Strict Alternation, Peterson's Solution, The Producer/Consumer Problem, Semaphores, Event Counters, Monitors, Message Passing, Classical IPC Problems : Reader's & Writer Problem, Dinning Philosopher Problem etc.

4. Deadlocks

Definition, Necessary and sufficient conditions for Deadlock, Deadlock Prevention, Deadlock Avoidance : Banker's algorithms, Deadlock detection and Recovery.

5. Memory Management

Basic Concept, Logical and Physical address map, Memory allocation : Contiguous Memory allocation - Fixed and variable partition - Internal and External fragmentation and Compaction; Paging : Principle of operation - Page allocation - Hardware support for paging, Protection and sharing, Disadvantages of paging.

Virtual Memory : Basics of Virtual Memory - Hardware and control structures - Locality of reference, Page fault, Working Set, Dirty page/Dirty bit - Demand paging - Page Replacement algorithms : Optimal, First in First Out (FIFO), Second Change (SC), Not recently used (NRU) and Least Recently used (LRU).

6. I/O Hardware

I/O devices, Device Controllers, Direct memory access, Principles of I/O software : Goals of Interrupt handlers, Device drivers, Device independent I/O software, Secondary Storage Sctructure : Disk structure, Disk scheduling algorithms.

File Management : Concept of File, Access methods, File types, File opeation, Directory structure, File system structure, Allocation methods (contiguous, linked, indexed), Free Space Management (bit vector, linked list, grouping), directory implementation (linear list, hash table), efficiency and performance.

Disk Management : Disk structure, Disk scheduling - FCFS, SSTF, SCAN, C-SCAN, Disk reliability, Disk formatting, Boot-block, Bad blocks.

Chapter 1

Introduction

Contents

Concept of Operating Systems, Generations of Operating Systems, Types of Operating Systems, OS Services, System Calls, Structure of an OS-Layered, Monolithic, Microkernel Operating Systems, Concept of Virtual Machine. Case study on UNIX and Windows Operating System.

POINTS TO REMEMBER

- The operating system attempts to schedule computational activitieis to ensure good performance of the computing system.
- To improve the overall performance of the computer system developers introduced the concept of multiprogramming so that several jobs could be kept in memory at one time.
- Multiprogramming also allows time sharing. Time sharing operating system terms allow many users to use a computer interactively at the same time.
- In multitasking system the computing is done by multitasks, also known as processes, share common processing resources such as CPU.
- The operating system must ensure correct operation of the computer system. To prevent users programs from interacting with the proper operation of the system, the hardware has two modes; user mode and monitor mode.
- A process is a program in execution. The main objective of the process management module of an operating system is to manage the process submitted to the system in a manner to minimize idle time.
- The process executing in the operating system may be either independent processes or co-operating processes.
- Each process is represented by a PCB and the PCB can be linked together to form ready queue.
- Each process may be in one of the following stages : new, ready, running, waiting or terminated.
- Operating systems are now almost always written in a system implementation language or in a higher level language. This feature improves that implementation, maintenance and portability.
- A process is a program in execution. As a process executes, it changes states. The state of a process is defined by that process current activity.
- Operating system provides a number of services. At the lowest level system calls allow a running program to make request from the operating system directly. At a higher level the command interpretes or shell provides a mechanism.
- Operating system can treat as a resource manager and as an implementor of virtual computers.

- A shell is a piece of software that provides an interface for users to an operating system which provides access to the services of a kernel.
- The user view of the computer varies according to interface being used.
- Real time computing system is a number of possible external activities needed to be controlled by a single processor system.
- The kernel responsibilities include managing the system's resource, communication between hardware and software components.
- The system call level must provide the basic functions such as process control and file and device management.
- Linux is a multiuser system provided protection between processes and running multiple processor according to a time sharing schedule.
- Linux is a modern operating system based on Unix standard.
- It is designed to run efficiently and reliably on common PC hardware.
- It provides programmer interface and use interface compatiable with standard UNIX system.
- Many modern operating systems provide kernel support for threads, among these are Windows NT, Window 2000, Solaries 2.
- Solaries 2 use priority based process scheduling.
- It has four classes of scheduling which are in order of priority, real time, system, time sharing and interactive.
- In the UNIX system, directory protection is handled similarly to file protection.
- The selected threads to runs on the CPU until one of the following occurs.
  - (i) It blocks
  - (ii) It uses its time slice
  - (iii) It is preempted by a higher priority thread.
- A real time process will run before on a process in any other class. In general few processors belong to the real time class.
- Threads in the real time class are given the highest priority to run among all classes.
- Scheduling policy for the system class does not need time slice. Rather a thread belonging to the system class runs until it either blocks or in preempted by a higher priority thread.
- Solaries 2 uses the system class to run kernel processes, such as the schedules and paging daemen.
- Pages are loaded on demand when they are first referenced and are paged back out to backing store according to a LFU algorithm if physical memory need to be reclaimed.
- [illegible]anagement system uses page sharing and copy on write to minimize the duplication of data shared by different processes.
- The core LINUX operating system kernel is entirely original but it allows much existing free unix software to run, resulting in an entire unix compatible operating system free from proprietory code.

QUESTION-ANSWERS

Q 1. What is an operating system? Why it is required? **(PTU, May 2015, 2010)**

Ans. It has been rightly said that operating system is the first software. We see that we turn-on the computer and the last software we see when the computer is turned-off.

Operating system is a program that, after being initially loaded into the computer by a boot program, manages all other programs in a computer. The most common operating systems are the windows family of operating system.

(Windows 95, 98, 2000, NT) the UNIX family of operating systems (which includes LINUX, BSD UNIX) and the macintosh operating system.

It is required in the form of processor management, memory management, input-output management, file management, establishment and enforcement of a job priority system.

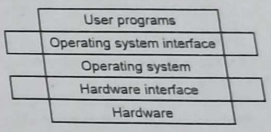
Q 2. What are different operating system services? **(PTU, Dec. 2017, 2007)**

Ans. 1. It performs basically two unrelated functions, i.e., extending the machine and manages resources.

2. Along with hardware the operating system and other software forms a complete, versatile, efficient and easy to use system for the users.
3. It is an interface between computer uses and hardware.
4. Operating system is also resource sharing and management interface resources.
5. It is the most powerful and important piece of software in computer system.

Q 3. Where does an operating system fit in?

Ans. An operating system is a layer of the computer system between the hardware and the user programs (user software). An operating system does what all softwares do. It implements some desired functionality by building on the functionality available in lower levels.



Q 4. Name any four multi-user operating system. **(PTU, Dec. 2008)**

Ans. 1. Windows
2. UNIX
3. LINUX
4. MAC.OS

Q 5. What is an operating system? Discuss various classification of operating system. **(PTU, Dec. 2019, 2010, 2009, 2005)**

OR

What are different types of operating system? Explain. **(PTU, May 2019, 2010, 2004))**

Ans. Operating system : An operating system is a program that interfaces between a user of a computer and the computer hardware.

Types of operating system :

1. Batch operating system : The uses of such system did not interact directly with the computer system. The user prepared a job, which consisted of the program and the data.

In order to use this system, a batch of these programmed cards instead of single card would be

given to the computer system, batch jobs would be stored up during working hours and then executed during off time or when computer is idle.

Advantages of Batch Processing : Batch processing is useful for operations that require the computer or a peripheral device for an extended period of time and very little user interaction.

Disadvantages of Batch Processing :

1. No interaction is possible with the user.
2. CPU sat idle during the transmission from one job to another.

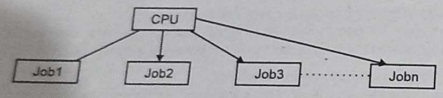
2. Multiprogramming : The operating system could support for keeping several jobs in memory at one time. The operating system could pick and start the execution of one of the jobs in the memory. Whenever the job does not need CPU, i.e., the job has to do only with the input-output devices, the CPU is idle at that time, the operating system switches to another job and CPU executes a portion of it till this job issues a request for input/output so it is called multiprogramming.

Advantages of multiprogramming :

1. High CPU utilization.
2. It appears that many programs are allotted to CPU for simultaneously work.

Disadvantages of multiprogramming : Many jobs may be ready to run on the CPU, which implies that we need CPU scheduling.

3. Time sharing : Time sharing is an operating system feature that allows several users to run several tasks concurrently on one processor or on many processors.



4. Real time system : A real time system is one in which the correctness of computations not only depends upon the logical correctness of the computation but also upon the time at which result is produced.

"Hence, it is important that timing constraints of the system are guaranteed to be met."

Areas where real time system used :

1. Air-traffic control
2. Robotics
3. Display screen of a airplane.

Q 6. What are the two main responsibilities of an operating system?

(PTU, Dec. 2019, 2011)

Ans. 1. An operating system manages hardware, runs application, provides an interface for users, and store, retrieves and manipulates files.

2. It manages the hardware and software resources of the system.

3. It provides a stable, consistent way for applications to deal with the hardware without having to know all the details of the hardware.

Q 7. How efficiency of operating system is measured?

Ans. The efficiency of operating system and the overall performance of a computer system are measured in following terms.

1. **Throughput :** Throughput is the amount of work that the system is able to do per unit time.

2. **Response time :** It is the interval time of a submission of a job to the system for processing to the time the first response for a job.

3. **Turnaround time :** It is the interval from the time of submission of a job to the system for processing to the time of completion of the job.

Q 8. What is multiprogramming? What are the factors affecting the degree of multiprogramming?

(PTU, Dec. 2007)

OR

Write short note on Multiprogramming.

(PTU, Dec. 2016, 2003)

Ans. Multiprogramming : The most important aspect of job scheduling is the ability to multiprogramming. The jobs are classified as CPU-Bound jobs and I/O bound jobs. A single uses cannot, in general keep either the CPU or I/O devices busy at all the times. Multiprogramming increases CPU utilization by organizing jobs so that the CPU always has one job to execute. In the early days of computing, CPU time was expensive and peripherals were very slow. When the computer ran a program that needed access to a peripheral, the CPU would have a stop executing program instructions while the peripheral process the data. This was deemed very inefficient multiprogramming does not give any guarantee that a program will run in timely manner.

In deed the very first program may very well ran for hours without needing access to a peripheral. Multiprogramming greatly reduced the working time. In some multiprogramming system only a fixed number of jobs can be processed concurrently while in other the number of jobs can vary.

Factors affecting degree of multiprogramming :

1. Jobs may have different sizes, therefore, memory management is needed to accommodate them in memory.
2. Many jobs may be ready to run on the CPU which implies that we need CPU scheduling.
3. It seems that many programs are alloted to CPU almost simultaneously.

Q 9. State and discuss the differences between Multiprogramming and Multitasking.

(PTU, May 2009)

Ans. Multiprogramming : The most important aspect of job scheduling is the ability to multiprogramming. The jobs are classified as CPU-Bound jobs and I/O bound jobs. A single uses cannot, in general keep either the CPU or I/O devices busy at all the times. Multiprogramming increases CPU utilization by organizing jobs so that the CPU always has one job to execute. In the early days of computing, CPU time was expensive and peripherals were very slow. When the computer ran a program that needed access to a peripheral, the CPU would have a stop executing program instructions while the peripheral process the data. This was deemed very inefficient multiprogramming does not give any guarantee that a program will run in timely manner.

In deed the very first program may very well ran for hours without needing access to a peripheral. Multiprogramming greatly reduced the working time. In some multiprogramming system only a fixed number of jobs can be processed concurrently while in other the number of jobs can vary.

Multitasking : In multitasking system the computing is done by multitasks, also known as processes, share common processing resources such as CPU. The CPU executes multiple- tasks by switching between them. But the CPU switching occurs so frequently that the users may interact each program while it is running. The act of reassigning a CPU from one task to another one is called context switch. When context switches occur frequently enough the illusion of parallelism is achieved. The following two types of schemes that multitasking operating system program uses :

1. Cooperative multitasking
2. Preemptive multitasking

Q 10. What is difference between multitasking and multiprocessing?
(PTU, May 2011 ; Dec. 2004)

Ans. Multitasking : It is a procedure by which multiple tasks, sharing resources like CPU are performed. The CPU executes multiple tasks by switching between them.

Multiprocessing : In multiprocessing there are more than one processor in close communication, sharing the computer bus, the clock and sometimes memory.

Q 11. What are the various features of networked operating system? (PTU, May 2004)

Ans. A network operating system is a software applicaiton that provides the functionality of an individual computer and for multiple computer within interconnected network. Basically network operating system controls other software and computer hardware to run applications, share resources protect data and establish communication. Individual computer runs client operating system while network system creates the software infrastructure for wireless, local and wide area network functions.

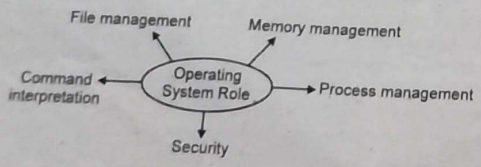
Basic Operating Features :

1. It supports the basic underlying operating feature of network. These include support for processors and the various protocols that allow computer to share data. Many network operating systems can detect hardware within the system to allow for asset discovery within the network.
2. Network operating system supports a number of security features that control access to the network. These include authorization and permission to access to the network with specific control of features. Such as user management logon control and passwords.
3. A network operating system is the platform on which computer networking takes place. Basic features allow for file, print and internet connections data backup and replication functions are controlled through the network operating system. The management of connective systems for local and wide area network. Such as routing, switches and other parts are configured and managed through network operating system.
4. One of the features of operating system is that it has an administrative interface that allows a network administrator to monitor and maintain a system. This interface will have a menu that allows the administrator to perform functions such as formatting hard drives and setting up.

Security protocols for both the system and the individual users. He can also configure security and data backup requirement for individual computers or the network as a whole.

Q 12. What is the role of an operating system?

Ans. When the power to a computer is turned-on, the first program that runs is usually a set of instructions kept in the computer's read only memory that examines hardware, S/W and resources is done by operating system.



Q 13. What are the main functions of an operating system?
(PTU, May 2009, 2007)

Ans. 1. File management : The file management means take care of file related activities such as organization, storing, retrieval, etc.

2. Memory management : The memory management means take care of allocation and deallocation of memory.

3. Security : Security means protect the resources and information from unauthorized users.

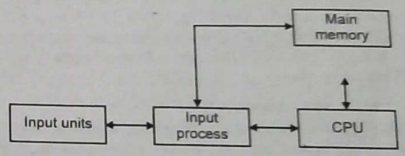
4. Process management : It means take care of creation and deletion of processes, scheduling of various resources.

Q 14. What is process management?

Ans. A process (also called job) is a program in execution. The main objective of the process management module of an operating system is to manage the processes submitted to the system in a manner to minimize the idle time of the various processors.

Process management ways :

(i) Batch processing
(ii) Multiprogramming
(iii) Multitasking
(iv) Multiprocessing



Architecture of computer system how process

Q 15. What is the main goal of memory management? **(PTU, Dec. 2006)**

Ans. It is the job of the operating system to coordinate how memory to be used. The part of the operating system that manages the memory hierarchy is called the memory manager. Its job is to keep track of which parts of memory are in use and which parts are not in use, to allocate memory to processes when they need it and deallocate it when they are done and to manage swapping between main memory is too small to hold all the processes.

Q 16. What is file management?

Ans. A file is a collection of related information. Every file has a name, its data and attributes. A file's data is its contents. The contents of a file is sequence of bits, bytes, lines, or records. The file management module of an operating system takes care of file related activities, such as accessing, sharing and protection of files.

File access methods :

1. Sequential access method
2. Random access method

File operations :

1. Create

2. Delete
3. Open
4. Close
5. Read
6. Write
7. Seek
8. Get attributes.

Q 17. What is bootstrap loader? **(PTU, Dec. 2011)**

Ans. Most computer systems can only execute code found in the memory (RAM or ROM). Modern operating systems are stored on hard disks or occasionally on line CDs, USB, flash drivers or other non-volatile storage devices then a computer is first powered on, it performs complex actions such as loading a program from disk, so on apparent paradox exists: to load the operating system into memory, one appears to need to have an operating system already loaded. The solution is to use a special small program, called a bootstrap loader. This program's only job is to load other software for the operating system to start. Often, multiple-stage boot loaders are used in which several small programs of increasing complexity summon each other, until the last of them loads the operating system. The name bootstrap loader comes from the image of one pulling oneself up by one's bootstraps. It derives from the very earliest days of computers and is possibly one of the oldest processes of computer technology in common use easily programmable computers had a row of toggle switches front and to allow the operator to manually enter the binary boot instructions into memory before transferring control to CPU. The boot loader would then read the operating system in from an outside storage medium such as paper tape, punched card or a disk drive.

Q 18. What is a real time operating system? **(PTU, May 2009)**

Ans. It is used when rigid time requirements have been placed on the operation of a processor or the flow of data ; thus, it is often used as a central device in a dedicated application. Sensors bring data to the computer. The computer must analyze the data and possibly adjust controls to modify the sensor inputs, systems that control scientific experiments; medical imaging systems, industrial control systems and certain display systems are real time systems. Some automobile engine fuel injection system, home appliance controllers and weapon system are also real time systems. A real time system has well defined fixed time constraints. Processing must be done within the defined constraints or the system will fail.

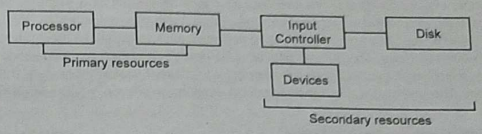
Q 19. What does an operating system do?

Ans. 1. As a resource manager.
2. As an implementor of virtual computers.

Q 20. Role of operating system as a resource management. **(PTU, Dec. 2018)**

Ans. The operating system manages the hardware resources of computer system. The hardware resources in a computer system are :

1. Processor
2. Memory
3. Input/output controllers
4. Disk Devices



Operating manages and controls these resources.

Q 21. What is resource management?

Ans. Resource management is :

- **Transforming :** Creating a new resource from an existing resource. The created resource will act as a substitute for the existing resource but will be easier to use.
- **Multiplexing :** Creating the illusion of several resources from one resource.
- **Scheduling :** Deciding which programs should get each resource and when they should get it.

Q 22. Write some popular operating systems.

Ans. UNIX : Unix is a multiuser, time sharing operating system.

MS-DOS : MS-DOS stands for Microsoft Disk Operating System. It is a single user operating system for IBM.

Microsoft Windows : It was developed by microsoft to overcome the limitation of MS DOS. Its native interface is GUI.

Microsoft Windows NI : It is also native interface of GUI. It supports multiprogramming.

LINUX : It is an open-source operating system.

Q 23. What is the principal disadvantage of too much multiprogramming? **(PTU, May 2007)**

Ans. Multiprogramming requires content switching since several programs sequentially share the CPU. This typically introduces processing delay compared to a dedicated unshared process. When we have too much multiprocessing, these delays could be significant if random access memory resource limits are reached and page faults occur requiring memory swaps to disc.

Q 24. What are computer system resources that operating system handle?

Ans. There are two types of resources :

1. Primary resources : Processor, memory.

2. Secondary resources : Input/Output controllers, disk, devices.

Q 25. Give at least three different views of Operating System. **(PTU, May 2018, 2016, 2010 ; Dec. 2003)**

Ans. Following are the different views of operating system :

1. User View : The user view of the computer varies by the interface being used. Most computer users sit in front of a PC, consisting of a monitor, keyboard, mouse and system unit. Such a system is designed for one user to monopolize its resources to maximize the work that the user is performing. In this case, the operating system is designed mostly for case of use, with some attention paid to performance and none paid to resource utilization.

2. System View : From the computer point of view, the operating system is a program that is

most intimate with the hardware. We can view an operating system as a resource allocator. The operating system acts as the manager of all hardware and software resources.

3. Efficient Use : It is easier to define an operating system by what it does than by what it is, but even this can be tricky. The primary goal of some operating system is convenience for the user operating systems exist because they are supposed to make it easier to compute with them than without them. The primary goal of other operating systems is efficient operation of the computer system.

Q 26. Write the functions of kernel. **(PTU, May 2017, 2015)**

Ans. The functions of kernel are :

1. Process scheduling
2. Memory allocation
3. Device allocation
4. Add new components or modify existing one.

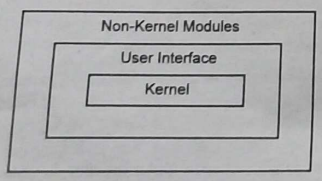
Q 27. Explain the mechanisms of operating system kernel.

Ans. Mechanisms of operating system kernel :

1. Interrupt processing mechanisms
2. Scheduling mechanisms
   - ❑ Program dispatching
   - ❑ Program preemption
3. Memory management mechanisms
   - ❑ Memory protection
   - ❑ Memory swapping mechanisms e.g.,
   - ❑ Swapping in/out
4. Input/output mechanisms
   - ❑ Input/output initiation
   - ❑ Input/output completion
5. Communication mechanisms
   - ❑ Interprocessing communication mechanisms
   - ❑ Networking mechanisms.

Q 28. Define operating system layers.

Ans.



O.S. layer

Q 29. Explain microkernel.

Ans. *As the unix operating system expanded, the kernel becomes large and difficult to manage.*

In the mid 1980 researches at Earnegtic Imellon University developed an operating system called much that modularizes the kernel using microkernel approach. The main function of the microkernel is to provide a communication facility between the client program and the various services that are also running in use space. For example, if the client program wishes to access a file, it must interact with the file server. The client program and the service never interact directly. Rather they communicate by exchanging message with the microkernel. The benefit of the microkernel approach includes the case of extending the operating system.

Q 30. Define shell. **(PTU, May 2018, 2017)**

Ans. A shell is a piece of software that provides an interface for users to an operating system which provides access to the services of a kernel.

Operating system kernel or shell generally falls into one of the two categories :

1. Command Line Interface (CLI)
2. Graphical Shell Interface (GUI)

Q 31. Define merits of CLI and GUI.

Ans. The relative merits of CLI and GUI based shells are often debated. CL interface proponents clear that certain operations can be performed much faster under command line interface.

GUI proponents advocate the instability and simplicity of graphical user shells.

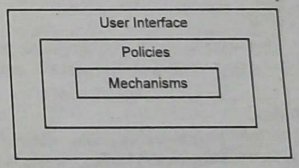
Q 32. Explain operating system structure.

Ans. Operating system design strongly depends on two factors :

- ❑ Architectural features of the computer on which it operates.
- ❑ Features of its application domain.

1. Dependence on architectural features is caused by the need to exercise complete controls overall functional units of the system. Hence, the operating system needs to know the **addressing structure,** interrupt structure. Input output organization and memory protection features of the computer system.

2. Application domain : The CPU scheduling policy depends on whether the operating system will be used for timesharing or for real time applications. The dependence on these two factors is very important.

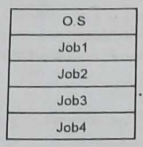


Policies : Policies governing the use of resources.
Mechanisms : Mechanisms to implement the policy.

Q 33. How multiprogramming and timesharing is important in operating system structure?

Ans. An operating system provides the environment within which programs are executed. One of the most important aspects of operating systems is the ability to multiprogram. A single user cannot, in general, keep either the CPU or the input/output devices busy at all times. Multiprogramming

increases CPU utilization by organizing jobs so that the CPU always has one to execute. The operating system keeps several jobs in memory simultaneously.



Memory layout of multiprogramming

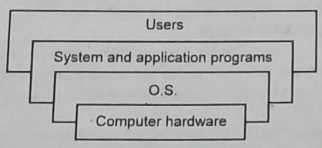
A timeshared operating system allows many users to share the computer simultaneously, since each action or command is in a timeshared system.

A timesharing operating system uses CPU scheduling and multiprogramming to provide each user with a small portion of **timeshared computers**.

Q 34. Components of computer system.

Ans. ❑ Operating system

❑ Computer hardware



❑ System and application programs

❑ Users

Q 35. Explain user's view.

Ans. *The user's view of the computer varies according to interface being used. The goal is to maximize the work that the user is performing. The main goal is to ease of use not paid to resource utilization.*

Q 36. Explain system view.

Ans. *The operating system is the program most intimately involved with the hardware. In system view, we can view an operating system as a resource allocation. A computer system has many resources that may be required to solve a problem. CPU time, memory space, file-storage space, input/output devices and so on. The operating system acts as resource manager.*

Q 37 What is real time processing? **(PTU, May 2008)**

Ans. *Real time computing system is a number of possible external activities needed to be controlled by a single processor system. A real time system is used when there are rigid time requirements on the operation of a processor or flow of data. In such system hierarchical interrupt*

system was coupled with process prioritization to ensure that key activities were given share of available process time.

Q 38. Explain the term system call and software interrupt. **(PTU, May 2004)**

Ans. A software interrupt occurs at the request of programmer. They are used to implement system calls. Software interrupt is considered to be an exception. As far as LINUX is concerned, software interrupts are handled by the CPU as trap so you might see somewhere else that system calls are implemented by trap.

Signals are part of the intercommunication. They are just like asynchronous message that one process can send to the another one.

Q 39. What is multitasking system? **(PTU, May 2010)**

Ans. In multitasking system the computing is done by multitasks, also known as processes, share common processing resources such as CPU. The CPU executes multiple tasks by switching between them. But the CPU switching occurs so frequently that the uses may interact each program while it is running. The act of reassigning a CPU from one task to another one is called content switch. The following two types of schemes that multitasking operating system program uses :

1. Cooperative multitasking
2. Preemptive multitasking.

Q 40. What are the main advantages of the multiprogramming?
(PTU, Dec. 2012 ; May 2010)

Ans. Advantages of Multiprogramming : Multiprogramming increases CPU utilization by organizing jobs so that the CPU always has one to execute. In this, the operating system keeps several jobs in memory simutaneously. This set of job is a subset of the jobs kept in the job pool, since the number of jobs that can be kept simultaneously in memory is usually much smaller than the number of jobs that can be in the job pool. The operating system picks and begins to execute one of the jobs in the memory. In a non-multiprogrammed system, the CPU would sit idle. In multiprogramming system, the operating system simply switches to and executes another job.

Q 41. What is a time sharing system? **(PTU, May 2010 ; Dec. 2004)**

Ans. Time sharing System : The timesharing is a logical extension of multiprogramming. The CPU executes multiple jobs by switching among them, but the switches occur too frequently that the users can interact with each program while it is running. An interactive computer system provides direct communication between the user and the system. A timeshared operating system allows many users to share the computer simultaneously. Since each action in a timeshared system tends to be short, only a little CPU time is needed for each user. As the system switches rapidly from one user to the next, each user is given the impression that the entire computer system is dedicated to her use, even though it is being shared among many users.

Q 42. What is job pool?

Ans. Timesharing and multiprogramming require several jobs to be kept simultaneously in memory. Since in general main memory is too small to accommodate all jobs, the jobs are kept initially on the disk in the job pool.

Q 43. What is job scheduling?

Ans. The job pool consists of all processes residing on disk awaiting allocation of main memory. If several jobs are ready to be brought into memory, and if there are not enough room for all of them, then job scheduling is done means the operating system selects a job from the job pool.

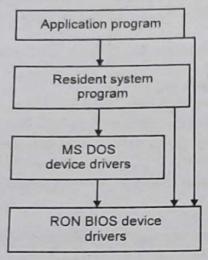
Q 44. What is CPU scheduling?

Ans. If several jobs are ready to run at the same time, the system must choose among them,

running multiple jobs concurrently requires that their ability to affect one another be limited in all phases of the operating system.

Q 45. Explain simple operating system structure.

Ans. MS-DOS is an example of simple operating system structure.



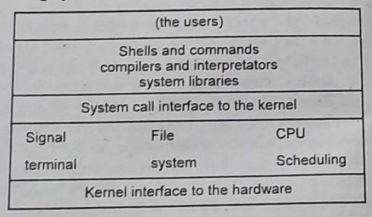
It was written to provide the most functionality and the least space.

In MS-DOS, the interfaces and levels of functionality are not well separated.

Q 46. Explain layered approach of operating system structure.

Ans. With proper hardware support, operating system can be broken into pieces that are similar and more appropriate than those allowed by the original MS-DOS or UNIX systems.

The operating system can then retain much greater control over the computer and over the applications that make use of that computer. Implementation have more freedom in changing the inner workings of the system.



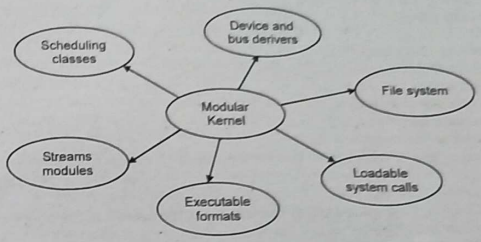
UNIX System Structure

Q 47. Define operating system services.

Ans. 1. User interface
2. Program execution
3. Input/output operation
4. File-system manipulation
5. Communications
6. Error detection
7. Resource allocation
8. Accounting
9. Protection and security

Q 48. Explain modular kernel.

Ans. The best current methodology for operating system design involves using object-oriented programming techniques to create a modular kernel. Here the kernel has a set of core components and dynamically links in additional services either during boot time or run time.



Q 49. Explain resource allocation.

Ans. When there are multiple users or multiple jobs running at the same time, resources must be allocated to each of them. Many different types of resources are managed by the operating system. Such as CPU cycles, main memory and file storage.

Q 50. How operating system act as resource manager?

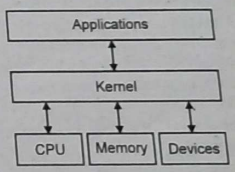
Ans. Resource management includes multiplexing in two ways – "in time" and "in space".

(i) When a resource is time multiplexed different programs or different users get their turn to use that resource, e.g., Printer.

(ii) When a resource is space multiplexed instead of taking turns, then resource is shared among them as each one gets a part of the resource, e.g., Sharing main memory.

Q 51. Explain kernel computing.

Ans. Kernel computing : A kernel connects the application software to the hardware of a computer.



In computing the kernel is the main component of most computer operating systems.

Q 52. Explain kernel's responsibilities.

Ans. The kernel's responsibilities include managing the system's resource, communication

between hardware and software components. It provides lowest level abstraction layer for the resources, manage system call, etc.

Q 53. Explain kernel's functions.

Ans. The kernel's primary function is to manage the computer's resources and allows other programs to run and use these resources.

Resources are :

1. CPU
2. The computer's memory
3. Input-output devices
4. Address space
5. Time or space multiplexed
6. Synchronization
7. Interprocess communication.

Q 54. What are the different objectives for the operating system to decide scheduling? **(PTU, Dec. 2017, 2009, 2005)**

Ans. Different objectives are :

1. CPU utlization : CPU utilization may range from 0 to 100 percent. In a real system it should range from 40 percent to 90 percent.

2. Through put : If the CPU is busy executing processes then work is being done. One measure of work is the number of processes completed per time unit called throughput

3. Turnaround time : From the point of view of particular process, the important criterion is how long it takes to execute that process. The interval from the time of submission of a process to the time of completion is the turnaround time.

4. Waiting time : The CPU scheduling algorithm does not affect the amount of time during which a process execute or does I/O. It affects one, the amount of time that a process spends waiting in the ready queue. Waiting time is the sum of the period spent waiting in the ready queue.

5. Response time : In an interactive system turn around time may not be her criterion often a process can produce some output fairly early and can continue computing new results while previous results are being output to the user. Thus another measure is the time from the submission of a request until the first response is produced. This measure is called response time.

Q 55. What is multiprocessing? Is it same as parallel processing? **(PTU, Dec. 2013, 2004)**

Ans. Multiprocessing : Multiprocessing is the use of two or more central processing units (CPU) with in a single computer system. The term refers to the ability of a system to support more than one processor and/or the ability to allocate tasks between them. There are many variations on the basic theme, and the definition of multiprocessing can vary with content mostly as a function of how CPUs are defined. It sometimes refers to the execution of multiple concurrent software processes in a system as opposed to a single process at any instant.

No, it is not same as parallel processing because in parallel processing the simultaneous use of more than one CPU to execute a program is there.

Q 56. What is meant by 16-bit operating system? **(PTU, Dec. 2004)**

Ans. Computer actually run "machine language". When user write a program in some programming language, it gets translated into machine language by the compiler. Each machine language command tells the computer to do one thing. An 8-bit operating system uses instructions that are 8 bit long. A 16 bit operating system uses instructions that are 16 bit long. There are 32 bit operating system and 64 bit operating systems.

Q 57. What is meant by concurrent processing? **(PTU, Dec. 2004)**

Ans. With an application composed of many concurrent process, we lose the convenience offered by the determination of sequential programs. The simultaneous execution of several interrelated computer programs. A sequential computer program consists of a series of instructions to be executed one after another. A concurrent program consists of several sequential programs to be executed in parallel. Each of the concurrently executing sequential programs is called a process.

Q 58. What are batch systems? **(PTU, Dec. 2004)**

Ans. Early computers were physically enormous machines run from a console. The common input devices were card readers and tape drives. The common o/p devices were line printers, tape drives and card punches. The user did not interact directly with the computer systems. The user prepaired a job, which consisted of the program, the data, and some control information about the nature of the job and submitted it to the computer operator. The job was usually in the form of punch cards. At some later time the output appeared. The output consisted of the result of the program.

Q 59. Open Source Software. **(PTU, May 2012)**

Ans. Open source refers to a program in which the source code is available to the general public for use and modification from its original design free of charge i.e. open. Open source is typically created as a collaborative effort in which programmers improve upon the code and share the changes within the community. Open source software in the technological community as a responses to proprietary software owned by corporations. The open source includes the concept of concurrent yet different agendas and differing approaches in production.

Q 60. Boot Process of a PC. **(PTU, May 2012)**

Ans. Booting the PC is the process of loading the operating system from disk into memory. The first position of the boot sequence is controlled by programs such stored in system ROM and this portion will be the same no matter which operating system is installed on the PC.

1. The user switches on the system.
2. The power supply does a self check and then sends a power good design or signal to the CPU.
3. The CPU starts executing code stored in ROM on the motherboard at address FFFFO.
4. The routines in ROM test the central hardware, search for the video ROM, perform a checksum on the video ROM.
5. The routines in the motherboard ROM then.

Q 61. What is the difference between a hard real time system and soft real time system? **(PTU, Dec. 2012)**

Ans. A hard real time system guarantees that critical tasks complete on time. This goal requires that all delays in the system be bounded from the retrieval of the stored data to the time that it takes the operating system to finish any request made of it. A soft real time system where a critical real time task gets priority over other tasks and retains that priority until it completes.

As in hard real time systems kernel, delays need to be bounded. A non real time system is one that cannot guarantee a response time in goal of a hard real time system is to ensure that deadlines are met.

Q 62. Define the term semaphore.
(PTU, Dec. 2012, 2009, 2005 ; May 2012, 2011, 2008)

Ans. A semaphore is a service very often offered by real time operating systems to allow programmers to perform one of two major functions ; synchronize 2 tasks or control sharing of resources between 2 or more tasks.

Types of semaphores :
1. Binary semaphores
2. Counting semaphores
3. Mutexes.

Q 63. What is a context switch? **(PTU, Dec. 2019, 2018, 2011)**

Ans. To give each process on multi programmed machine a fair share of the CPU, a hardware clock generates interrupts periodically. This allows the operating system to schedule all processes in main memory to run on the CPU at equal intervals. Each time a clock interrupt occurs, the interrupt handler checks how much time the current running process has used, if it has used up its entire time slice, then the CPU scheduling algorithm picks a different process to run. Each switch of the CPU from one process to another is called a context switch.

Q 64. What do you understand by FAT? Explain. **(PTU, May 2011)**

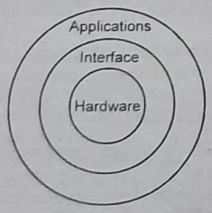
Ans. A table that the operating system uses to locate files on a disk. Due to fragmentation a file may be divided into many sections that are scattered around the disk. The FAT keeps tracks of all these pieces.

In DOS system, FATs are stored just after the boot sector. The FAT system for older version of Windows 95 is called FAT16 and the one for new versions of Windows 95 and Window 98 is called FAT 32. The FAT volume is divided into four areas :

1. The boot record 2. File allocation tables 3. The root directory 4. The data area.

Q 65. What is the overall structure of an OS? Draw a neat architecture and explain.
(PTU, May 2012)

Ans.



Operating System Concept :

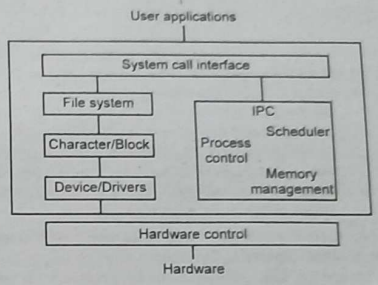
1. Hardware : The hardware is obviously the physical hardware and particularly intersecting to us in this.

2. Kernel : The kernel of an operating system is the bottom-most layer of software present on a machine and the only one with direct access to the hardware.

3. Outer operating system : Surroundings the kernel are other parts of the operating system. These perform less critical functions.

4. Interface : The interface provides a mechanism for you to interact with the computer.
5. Applications : There are what do the actual work they can be complex.

Architecture of operating system :



Q 66. What are the differences between user level threads and kernel level threads? Under what circumstances one is better than other? **(PTU, Dec. 2013)**

Ans.

User level threads	Kernel level threads
1. User level threads are faster to create and manage.	1. Kernel level threads are slower to create and manage.
2. Implementing by a thread library at the user level.	2. Operating system support directly to kernel threads.
3. User level thread can run on any operating system.	3. Kernel level threads are specific to the operating system.
4. Multithread application can't take advantage of multiprocessing.	4. Kernel routine themselves can be multithreaded.

Circumstances :

1. If the thread in a process are not performing the blocking system call. The user level thread can be used.

2. If a thread perform a blocking system call. The kernel can schedule another thread for location.

Q 67. Explain the architecture of UNIX.

Ans. Unix architecture is a computer operating system architecture that embodies the UNIX philosophy. It may adhere to standards such as the single UNIX specification (SUS) or similar POSIX 12LL standard describes all UNIX architecture computer operating systems that is an part of legacy of the Unixwars.

There are many systems which are UNIX-like in their architecture. Notable among these are the GNU/UNIX distributions. The distinctions between the UNIX and UNIX like systems have been the subject of heated legal battles, and the holders of the UNIX brand.

Q 68. What are the process states in UNIX?

Ans. Following are the process states :

1. New
2. Running
3. Waiting
4. Ready
5. Terminated.

Q 69. Features of LINUX file system. **(PTU, Dec. 2016)**

Ans. Features of Linux file system : Linux file systems and their main features :

File system	Features
Ext 2	Stable, general use, can be shrunk or expanded.
Ext 3	Stable, general use, quick recovery, can be shrunk or expanded.
Ext 4	New, general use, quick recovery, improves on ext 3
XFS	Stable, general use, quick recovery, can be expanded online.
JFS	Stable, general use, quick recovery.

Q 70. Write two advantages of LINUX Operating System. **(PTU, May 2019 ; Dec. 2016)**

Ans. Advantages of LINUX Operating System :

1. Reliability : Linux is a stable operating system. Linux servers are not shut down for years together. This means that users on the Linux work consistently with the Linux server without reporting any operating systems failure.

2. Backward compatibility : Linux has excellent support for hardware as well as old features of Linux are supported by newer versions.

Q 71. Explain the file system in UNIX.

Ans. A file system is a method of storing and organizing computer files and their data. Essentially, it organizes these files into a database for the storage organization. Manipulation and retrieval by computer's operating system.

File systems are used on data storage devices such as a hard disk or CD-ROMs to maintain the physical location of the files. Beyond this, they might provide access data on a file server by acting as clients for a network protocol, as they may be virtual or exists only as an access method for virtual data.

Q 72. Explain the directory structure of UNIX. **(PTU, Dec. 2004)**

Ans. Directory structure of unix : Following generalized overview of common locations of files in a unix system :

/(root) : The / notes the "root" of the file system where the entire system is contained.

/bin : Contains fundamental utilities needed by a system administrator.

/etc : Contains configuration files and some system databases.

/dev : Short for devices contains file representations of every peripheral device attached to the sytem. It also contains :

/dev/null : This virtual file discards all contents written to it.

/dev/random : This virtual file contains random numbers.

/dev/urandom : Except it always returns random numbers, even if there is not enough entropy in the system noise available.

/home : Contains home directories for the users.

/mnt : default location to mount external devices like hard disk drives, pen drives, etc.

/lib : Depository of all integral unix system libraries.

/tmp : a place for temporary files.

/usr : It how holds executables, libraries and shared resources that are not system critical.

/var : A place for files that may change after, such as the storage to a database, the contents of a database, log files, email stored on a server, etc.

Q 73. What is the booting process of UNIX? **(PTU, Dec. 2016)**

Ans. The boot program is stored in a standard location on a bootable device. For a normal boot from disk, for example, the boot program might be located in block 0 of the root disk or less commonly, in a special partition on the root disk. In the same way, the boot program may be the second file on a bootable tape or on a designed location on a remote file server in the case of a network boot of diskless workstation.

There is usually more than one bootable device on a system. The firmware program may include logic for selecting the device to boot from, other in the form of a list of potential devices to examine.

Q 74. Write the features of Unix.

Ans. Unix is a computer operating system originally developed in 1969 by a group of AT/T employees at Bell Labs, including Ken Thompson, Dennis Ritchie, Brean Kornighan, Douglas Mitlroy and Joe Ossanna. Today's Unix systems are spit into various branches, developed overtime by AT and T as well as various commercial vendors and non-profit organisations.

The open group, an industry standards consortium, owns the "Unix" trademark. Only systems fully complaint with and certified according to the single Unix specification are qualified to use the trademark ; others may be called "Unix system like" or "Unix like". However, the term "Unix" is often used informally to denote any operating system that closely resembles the trademarked system.

During the late 1970s and early 1980s the influence of Unix is acedamic circles led to large. Scale adoption of Unix by commercials stateups, the most notable of which are solaris, HP-UX and AIX. Today, in addition to certified unix systems such as those already mentioned, unix like are commonly encountered. Mac OS Xes also a unix system developed by Apple Inc. The term characteristic of either version 7 Unix or Unix System V.

Q 75. Write short note on Linux operating system. **(PTU, Dec. 2005)**

Ans. Linux operating system : The linux kernel is an operating system kernel used by the linux family of unix like operating system. It is one of the most prominent examples of free and open source software.

The linux kernel is released under the GNU general public license version 2 plus proprietary licenses.

Linux supports true preemptive multitasking, virtual memory, shared libraries, demand loading, shared copy on write executable, memory management and the internal protocol suit and threading.

Q 76. Give the architecture of LINUX operating system.

(PTU, Dec. 2010, 2008, 2004)

Ans. LINUX operating system : The Linux kernel is an operating system kernel used by the Linux family of unix like operating systems. It is one of most prominent examples of free and open source software.

The Linux Kernel is released under the GNU general public License Version 2 plus proprietary licenses. Linux supports true preemptive multitasking, virtual memory, shared libraries.

Architecture : Linux is a monolithic kernel, device drivers and kernel extensions run in kernel space with full access to the hardware, although some exceptions run in user space. The graphics system most people use with Linux doesn't run in the kernel, in contrast to that found in Microsoft Windows.

Kernel mode preemption allow device drivers to be preempted under certain conditions. This feature was added to handle hardware interrupts correctly and improve support for symmetric multiprocessing.

Q 77. Brief about the initial process sequence while system boot's up.

(PTU, Dec. 2006)

Ans. For a computer to start running, for instance, when it is powered up or rebooted it needs to have an initial program to run. The initial bootstrap program tends to be simple. It initializes all aspects of the system, from CPU registers to device controllers and the content of main memory, and then starts the operating system. To do so, the bootstrap program finds the operating system kernel on disk, loads the kernel into memory and jumps to a initial address to begin the operating system execution.

For most computers, the bootstrap is stored in read-only memory (ROM). This location is convenient, because ROM needs no initialization and is at a fixed location that the processor can start executing when powered up or reset. The full bootstrap program can be changed easily: A new version is simply written onto the disk. The full bootstrap program is stored in a partition called the boot blocks, at a fixed location on the disk. A disk that has a boot partition is called a boot disk.

The code in the boot ROM instructs the disk controller to read the boot blocks into memory and then starts executing that code. The full bootstrap program is more sophisticated than the bootstrap loader in the boot ROM ; it is able to load the entire operating system for a non-fixed location on disk and to start the operating system running. Even so, the full bootstrap code may be small, e.g., MS-DOS uses one 512 byte block for its boot program.

Q 78. Explain key features of UNIX.

Ans. 1. Time sharing operating system, which uses round robin CPU scheduling, for user processes.

2. Highly portable and is thus available for a variety of processor classes ranging from microprocessors to to supercomputers.

3. Has hierarchical file system that spans volume boundaries.

4. To reduce physical I/O and effective disk access time.

5. Use signals for process - synchronization while pipes provide for communication between related processes.

Q 79. Describing the booting process for a Window XP system.

Ans. As with other windows operating systems, when you turn on your PC, it goes through an elaborate boot up process. It begins when the computer performs the post followed by the post for each adapter card that has a BIOS. For example, your video card.

Q 80. What are the responsibilities of object manager?

Ans. The object manager acts like a dedicated client except that it represents multiple clients instead of just one client. With this in mind, the object manager needs to understand how to display things on the screen, what rules to enforce etc. To do this, the object manager reads the srf file on an as needed basis. The object manager also manages client specific information. As a rule of thumb

each client will add about 4mb of memory usage to the object manager on top of all the object definition memory. This is a rule of thumb and can vary greatly depending on usage.

As thin clients log in out of server the object manager usage will go up and down. If all users log off the object manager processes my stick around for 20-30 minutes while time outs are reached. Once all the user connections are cleaned up the object SIEBMTSH will shut down. The "Minimum Multi-threaded servers" parameter value may prevent SIEBMTSH processes from shutting down.

MIN MT servers govern how many processes to keep survey where there is no activity for a component. One reason to do this to avoid having a login delay when a thin client tries to connect and a new process is started to handle the request.

Q 81. How does XP Window allocates user memory windows allocates reburces according to its settings and manages devices and programs accordingly. However, you can use the system item in control panel to change these performance options and how windows manage them?

Ans. Manual steps to set performance options in Window XP : If you would rather not install and run guided help. You can set performances options in Window XP manually. You must be logged on to windows by using a computer administration account in order to set many of these performance options.

Q 82. What is the reason for popularity of LINUX operating system?

(PTU, May 2007)

Ans. There are many reasons for the popularity of Linux operating system. Following are some of them :

1. Security : You should run linux is the security that Linux operating system provides. There are no viruses for Linux and there are no anti virus programs for Linux.

2. Cost : All Linux distributions are free open source software. And that is not all, the majority of the software is also free of charge.

3. Stability : Linux is also known for its stability. Its best suited for servers and the large computers. But it runs on small and handheld devices as well as on servers.

4. Works well on older hardware : Computers with older hardware have no or little problem running Linux. There is a great support for hardware and devices. The Linux kernel is constantly upgraded to support new devices and drivers are always upgraded to give better support for existing devices.

5. Administration : You can have as little as no administration which involves only clicking on install button to install new updates. It works out of the box and on the other hand you have a full set of administration tool you can use to do whatever you want.

6. User Interface : Users can control a linux-based system through a command line interface, a graphical user interface or through controls attached to the associated hardware. For desktop systems, the default mode is GUI, where the CLI is available through terminal emulator windows or on a separate virtual console.

7. Programming on Linux : Most Linux distributions support dozens of programming languages. The most common collection of utilities for building both Linux applications and operating system programs is found within the GNU toolchain, which includes GNU compiler collection (GCC) and the GNU build system. GCC provides compilers for Ada, C, C++, Java and Fortran. Most distributions also include support for PHP, Perl, Ruby, Python and other dynamic languages.

Q 83. What level of security that Windows NT meets?

Ans. The starting point for strong internet security is the operating system of any machine connected to it. Fortunately for the organizations using IIS 4.0, strong levels of security were built into the core of Windows NT in order to meet and exceed certifiable security standards, i.e. the C2 security guidelines required by the US department of Defence evaluation criteria. Windows NT security standards was originally only required for government organizations. However, many commercial organizations are demanding the same level of security and they recognize the value that such standards offer. The main requirements for C2 compliance are :

- User identification and authentication
- Discretionary access control
- Auditing capabilities
- Safe object reuse
- System integrity.

Q 84. What is SMP?

Ans. In computing, symmetric multiprocessing or SMP involves a multiprocessor computer architecture where two or more identical processors are connected to a single shared main memory and are controlled by a single OS instance. Most common multiprocessor systems today use an SMP architecture. In case of multi-case processors, the SMP architecture applies to the cases, treating them as a separate processors, the memory and the disk arrays. Mesh architectures avoid these bottlenecks and provide nearly linear scalability to much higher processor counts.

SMP : SMP systems allow any processor to work on any task no matter where the data for that task are located in memory, provided that each task in the system is not in execution on two or more processors at the same time. With proper operating system support, SMP systems can execute easily more tasks between procesors to balance the work load efficiently.

Q 85. What is an operating system?

Ans. Operating system : Operating system is a software that provides an interface between the hardware and other software. This operating system is responsible for management and coordination of processes and allocation and sharing of hardware resources such as RAM and disk space and acts as a host for computing applications running on the operating system. An operating system may also provide orderly accesses to the hardware by compelling software routines.

Q 86. What is kernel? **(PTU, Dec. 2010 ; May 2009)**

Ans. The kernel is the hub of the operating system. It allocates time and memory to programs and handles the file store and communications in response to system calls. As an illustration of the way that the shell and the kernel work together suppose a user type rm my file. The shell searches. The files store for the file containing the programs and then request the kernel through system calls to execute the program on my file.

Q 87. Explain how UNIX has a better policy to handle smaller files than the larger files? Explain how UNIX is booted. Show inode structure in UNIX. **(PTU, Dec. 2005)**

Ans. File specification provides the system with the needed information to uniquely identify a file. To avoid problems refrain from using characters that have other meanings. It is best to use one, letters, numbers and periods.

A full path name consists of/user/users 3/grad/home id is

where
/leading = Root of file system when it is the first character in the path name.
users = System directory one level below root in the hierarchy
/subsequent = Slash that delimits the directory names
user 3 igrad = Further subdirectory names
home dis = user's home directory

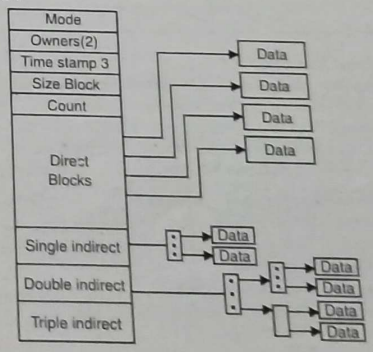
Example :

1. %l pr chad (1 – 3ad)
   Print all files that start with chap1, chap2, chap3, chapa
2. % cut?
   Type all files that consist of one character name if you wish to suppress the special meaning of *?, etc, enclose the entire argument in a single quota (e.g. cat '?')

Boot procedure : Bootstrapping is the process of starting up a computer from a halted or powered down condition. When the computer is switched on, it activates the memory resident code which resides onto CPU board. The normal facilities of the operating system are not available at this stage and the computer pulls itself up by its own boot straps so to speak. This procedure is often referred to as boot strapping also known as cold boot. Although the boot strap procedure is very hardware dependent it typically consists of following steps.

1. The memory resident code
   (a) Runs self test
   (b) Probes bus far the boot device
   (c) Reads the boot program from the boot device.
2. Boot program reads the kernel and passes control to it.
3. Kernel identify and configures the device.
4. Initializes the system and starts the system process.
5. Brings up the system in single user mode.
6. Runs the appropriate start up scripts.
7. Brings up the system for multi user operation.

Unix 1 mode structure



Q 88. Explain the components of a linux system. **(PTU, Dec. 2018)**

Ans. The three main bodies of code compose the linux system, in line with most traditional unix implementations.

1. Kernel : The kernel is responsible for maintaining all the important abstractions of the operating system, including such things as virtual memory and processes.

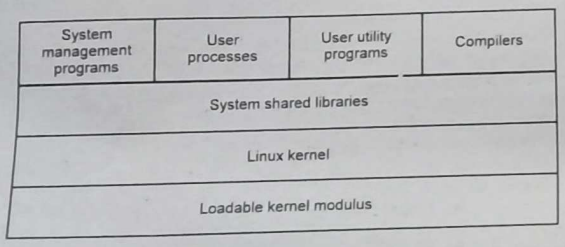
2. System libraries : The system libraries define a standard set of functions through which applications can interact with the kernel. These functions implement much of the operating system functionality that does not need the full privileges of kernel code.

3. System utilities : The system utilities are programs that perform individual, specialized management tasks. Some system utilities may be invoked just once to initialize and configure some aspect of the system. Others known as demons in UNIX terminology - may run permanently handling such tasks as responding to incoming network connections, accepting logon requests from terminates, and updating log files.

Q 89. Explain the diagram of components of the Linux system.

Ans. The Linux system is composed of three main bodies of code, in line with most traditional UNIX implementations :

1. Kernel
2. System libraries
3. System utilities



Q 90. What is Linux licensing?

Ans. Linux is free software, however, in the sense that people can copy it, modify it, use it in any manner that want, and give away their own copies, without any restrictions.

The main implications of Linux's licensing terms are that nobody using Linux or creating her own derivative of linux can make the derived product proprietary software released under the general public license, you must make source code available alongside any binary distributions.

Q 91. What are the various states of a process? **(PTU, Dec. 2006, 2004, 2003)**

Ans. As a process executes, it changes states. The state of a process is defined in part by the current activity of that process. Each process may be in one of the following states :

1. **New :** The process is being created.
2. **Running :** Instructions are being executed.
3. **Waiting :** The process is waiting for some event to occur.

4. **Ready :** The process is waiting to be assigned to a processor.
5. **Terminated :** The process has finished execution. These state names are arbitrary. Only one process be running on any processor at any instant, although many processes may be ready and waiting.

Q 92. Give comparison between NTFS, FAT and FAT32.

Ans.

Criteria	NTFS	FAT	FAT32
Operating System	Windows 2000 Windows XP Windows 2003 Server Windows Vista Windows 7	Windows CE 6.0 Windows Vista SP Windows 7	Dos V7 and higher windows Windows 98 Windows ME Windows 2000 Windows **XP** Windows **2003 Server** Windows Vista Windows 7

Limitations :

	NTFS	FAT	FAT32
Max Volume size	2TB	64ZB	32GB for all OS
Max file size	hinted only be volume size	16ZB	2TB for some OS 4194304
Max file size volume	Unlimited	4294967295	4177918

Q 93. Comment on merits and demerits of bat files OS MS DOS with comparison to batch processing. **(PTU, Dec. 2007)**

Ans. Merits :

1. Fewer keystrokes required to perform computer operations.
2. Major time savings.
3. Less chance of making typing errors.
4. Short commands are easier to remember.
5. One command executes an extended chain of complicated operations.

Demerits :

1. Bat files can make decisions to perform operations if certain conditions exist or do not exist.
2. Comments in batch files are usually placed in lines starting with REM. If you have many lines REMed out, this may slow down command. com processing of the batch file.

Q 94. Explain memory management in UNIX operating system.

Ans. When the UNIX kernel is first loaded into memory (at boot time), it sets aside a certain amount of RAM for itself as well as for all systems and user processes. Main categories into which RAM is divided are :

1. Text : To hold the text segments of the running processes.

2. Data : To hold the data segments of the running processes.

3. Stack : To hold the stack segments of the running processes.

4. Standard memory : This is an area of memory, which is available to all the running programs, if they need it.

5. Buffer cache : All the read and writes to the file system are cached here first.

6. Text segment : The text segment contains the machine instructions that form the program's executable code. The compiler or the assembler produces it by translating the C, Pascal, or other program into machine code. The text segment is normally read only.

7. Data segment : The data segment contains storage for the program's variables, strings, arrays and other data. It has two parts, the initialized data and the uninitialized data.

Q 95. What is Window NT operating system?

Ans. Window NTs development period, from 1988 to 1993. Microsoft built support in NT for DOS. Throughout NTs development period, most PCs used Intel's X86 processor. Its developers gave NT a multitasking preemptive scheduling system. It supports kernel mode (monitor mode) and user mode. User mode is the least-privileged mode NT supports and it has no direct access to hardware and only restricted access to memory.

Q 96. How is the NTFS namespace organized? Describe.

Ans. All file systems supported by windows use the concept of files and directories to access data stored on a disk or device. Windows developers working with the Windows Apis for file device I/O should understand the various rules, conventions and limitations of names for files and directories.

Data can be accessed from a disk, devices and network shares using file I/O.

Files and directories along with namespaces, are part of the concept of a path, which is string representation of where to get the data regardless if it is from a disk or a device or a network connection for a specific operation.

Q 97. What type of operating system is Windows XP?

Ans. Windows XP is an operating system produced by Microsoft for use on personal computers, including home and business desk-tops, laptops and media centres. It was first released in August 2001 and is currently one of the most popular versions of Windows. The name "XP" is short for "experience".

Windows XP is the successor to both Windows 2000 and Windows Me, and is the first consumer-oriented operating system produced by Microsoft to be built on the Windows NT Kernel and architecture. Windows XP was released for retail sale on October 25, 2001 and over 400 million copies were in use in Jaunary 2006, according to an estimate in that month by an IDC analyst. It was succeeded by Windows Vista, which was released to volume license customers on Nov. 8, 2006 and world wide to the general public on January 30, 2007. Direct OEM and retail sales of Windows XP ceased on June 30, 2008 Microsoft continued to sell XP through their system builders program until Jan. 31, 2009. XP may continue to be available as these sources run through their inventory or by purchasing Windows 7 ultimate, Windows 7 pro, Windows Vista ultimate or Windows Vista Business and then downgrading to Windows XP.

Q 98. What is Process in Linux? Explain '&' and 'kill' in detail. **(PTU, May 2009)**

Ans. Process in Linux : Linux can manage the processes in the system, each process is represented by a task-struct data structure. The data vector task-struct data structure. The task vector is an array of pointers to every task-struct data structure in the system.

This means that the maximum number of processes in the system is limited by the size of task

vector ; by default it has 512 entires. As processes are created, a new task-struck is allocated from system memory and added into the task vector. As well as the normal type of process, Linux support real time processes. As a process executes it changes state according to its circumstances

Linux processes have following states :

1. Running : The process is either running or it is ready to run.

2. Waiting : The process is waiting for an event or for a resource. Linux differentiates two types of waiting process ; interruptible and uninterruptible.

3. Stopped : The process has been stopped, usually by receiving a signal.

4. Zombie : This is a halted process, which for some reason, still has a task-struct data structure in the task vector. It is what it sounds like, a dead process.

Scheduling information : The scheduler needs this information in order to fairly decide which process in the system most deserves to run.

Kill : The command kill sends the specified signal to the specified process or process group. If no signal is specified, the term signal is sent. The term signal will kill processes which do not catch this signal. For other processes, it may be necessary to use the kill signal, since this signal cannot be caught.

Syntax

Kill [–S signal|–P] [–a] [– –] pid .....
Kill –l [signal]

The –a and –p options and the possibility to specify pid by command name is a local extension,

options :

pid .....

specify the list of processes that kill should signal. Each pid can be one of five things :

- n – where n is larger than 0. The process with pid n will be signaled.
- 0 – All processes in the current process group are signaled.
- – 1 – All processes with pid larger than, will be signaled.
- – n – where n is larger than 1. All processes in process group n are signaled.
- – S signal – specify the signal to send.
- – l – print a list of signal names
- – a – Do not restrict the commandance to pid conversion to processes with the same uid as the present process.
- – p – specify that kill should only print the process id of the named processes and not send any signals.

Q 99. SOLARIS OS. **(PTU, May 2012)**

Ans. Solaris is a unix operating system originally developed by Sun Microsystems. It superseded their earlier sun operating system in 1993. Oracle solaries as it is now known, has been owned by oracle corporation.

Solaris is known for its scalability, especially on SPARC system. Solaries used a common code base for the platform it supports. Solaries can be installed from physical media or a network for use on a desktop or server.

Q 100. SCO-UNIX. **(PTU, May 2012)**

Ans. SCO open server, previously SCO UNIX and SCO open desktop (SCO ODT) is a closed source version of the UNIX computer operating system developed by Santa Cruz Operation (SCO) group and now owned by UNIXs.

SCO UNIX was the successor to SCo Xenis.

Q 101. Security features in Linux. **(PTU, May 2012)**

Ans. 1. User accounts
2. File permissions
3. Data verification
4. Encrypted storage
5. Secure remote access
6. Software management
7. Host integrity testing
8. System recovery
9. Resource allocation controls
10. The system firewalls.

Q 102. What is a inode? **(PTU, Dec. 2004, May 2004)**

Ans. An inode is a data structure in UNIX operating systems that contains important information pertaining to files within a file system. When a file system is created in UNIX, a set amount of inode is created as well. Usually about 1 per cent of the total file system disk space is allocated to the inode table.

Q 103. What is the drawback of MVT?

Ans. 1. It does not have the features like ability to support multiple processors virtual storage source level debugging.
2. This arrangement suffer from external fragmentation.

Q 104. Explain the features of Linux system. **(PTU, Dec. 2012)**

Ans. Linux operating system : The Linux kernel is an operating system kernel used by the linux family of unix like operating systems. It is one of most prominent example of free and open source software.

Features of Linux :

- **Multitasking :** Several programs running at the same time.
- **Multiuser :** Several users on the same machine at the same time.
- **Multiplatform :** Runs on many different CPUs, not just Intel.
- **Multiprocessor/Multithreading :** It has native kernel support for multiple independent threads of control within a single process memory space.
- It has memory protection between processes, so that one program can't bring the whole system down.
- **Demand loads executables :** Linux only reads from disk those parts of a program that are actually used.
- Shared copy-on-write pages among executables. This means that multiple process can use the same memory to run it.
- Many networking protocols like TCP, IPV4, IPV6, AX,25 etc.
- Support several common file system, including minix, xenix and all the common system V file system.

Q 105. How the files are stored on the secondary memory storage? Discuss in context of DOS and UNIX partitioning and file management schemes. **(PTU, May 2012)**

Ans. It depends on the location whether or selected C:/. D:/ and E:/ by default it will stores in documents. This section will be looking at the various forms of secondary storage devices.

Seek time : The average time taken from requesting data to starting the read the requested data.

Cost : How much it costs for megabyte.

Magnetic media : It stores data by assigning a magnetic charge to metal. This metal is then processed by a read head, which converts the charges into ones and zeros.

- Hard disk
- Magnetic take drivers
- Optimal media.

DOS and UNIX partitioning and file management : With DOS, OS/2, Microsoft Windows, the standard partitioning scheme is to create a single active primary partition, the C: drive, where the operating system, utilities, applications, user data and page/swap file all reside for UNIX based operating system such as Linux the creation of separation partitions for /boot, /Home, /tmp all remaining files under "/" root directory is possible. Such a scheme has number of potential advantages. The same is true for sun operating system.

Q 106. Explain the security and protection mechanism of LINUX operating system. **(PTU, May 2011)**

Ans. Protection mechanism : The following mechanisms are commonly used for protecting files containing programs and data.

(i) Access control lists (ACL's)
(ii) Capability list (C-lists)

These lists are used to ensure that uses only access files which are explicitly authorized access. These files includes

(i) files created by a user himself/herself.
(ii) files owned by others, for which a user process explicit access privileged granted by other owners.

C-list : A capability is a file access privilages concerning capabilities possessed by a user is stored in a capability list. A c-list is a set of pairs {(file.id, access privileges—)}

Security mechanisms : Authentication is the primary security mechanisms. Authentication is the act of verifying the identity of user. Authentication is typically performed through passwords at login time. The system stores the password information in a system as set as pair of form (user id, password info). The password information is protected by encryption.

Q 107. What are various components of a file system? Explain the structure of UNIX file system. **(PTU, Dec. 2013)**

Ans. Various component of a file system :

(a) Device drivers
(b) Basic file system or physical input/output system.
(c) Basic input/output supervisor.
(d) Basic logical input/output.
(e) Different access method.

Structure of unix file system : The unix file system is composed of two types of entities:

(i) File
(ii) Directories

Every file and every directory has a pathname which indicates where that file or directory lines.

(i) File : File store information. There are many types of files. Two or more common type are text files (which you read) or executable files (which you run).

If you choose the option "Save as ..." at the bottom of this window. A dialog box will appear and prompt you for a "Name for saved document". If you type in a name and choose "OK", a text file will be created under the name you provided.

Files with names starting with a period (i.e. .xstartup) are invisible in normal unix use.

You can see then by using a variation of the ls command. Filename in unix generally do not certain embedded spaces. If you end up with a filename with embedded spaces, you can refer to it by enclosing the name in double queues.

(ii) Directories : Directories organize files and other directories, creating the tree structure of unix. When you login to your account, you arrive in your personal home directory, the root of your personnal section of the unix file system tree. You home directory in there in a subdirectory within other directories in the file system.

Pathnames : Files and directories are specified by pathnames. Pathname can be defined as a sequence of component names where each supported by slash character. The pathname of a file indicates its location in the file system. Two files can have the same name as long as their pathname are different.

If you choose the option "Save As". "The dialog box asking you for a filename will automatically fill in the pathname of your home directory.

For example : A pathname for a user's home directory is /use/jpower. The pathname for the user's mail directory might then be /user/jpower/mail.

When you tell UNIX to look for a text file, it checks for that file in the current directory. If the file is in other directory, you must supply the pathname to the file.

Q 108. What is the difference between Online and Real-time systems?
(PTU, Dec. 2016, 2013)

Ans. Online activity may be real time sometimes but there are online systems that are not real time.

Real time systems are those where the user gets immediate response to his reaction and there is no time delay.

If you are typing sometime and it appears on the screen of another person online after a lag of flow micro seconds, it is online but not real time.

Q 109. Explain the history of operating systems.

OR

Explain the various generations of operating systems.

Ans. The following are the various generations of operating system :

1. The First generation (1940's to early 1950's) : When electronic computers were first introduced in the 1940's they were created without any operating system. All programming was done in absolute machine language, often by wiring up plug boards to control the machine's basic functions. During this generation, computers were generally used to solve simple math calculations, operating systems were not necessarily needed.

2. The Second generation (1955–1965) : The first operating system was introduced in the early 1950's. It was called GMOS and was created by General Motors for IBM machine the 701 operating system in the 1950's were called single stream batch processing systems because the data was submitted in groups. These new machines were called mainframes and they were used by

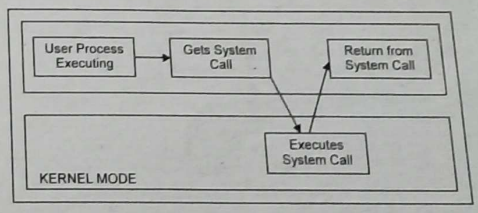
professional operators in large computer rooms. Since there was such a high price tag on these machines, only government agencies or large corporations were able to afford them.

3. The Third generation (1965 – 1980) : By the late 1960's operating systems designers were able to develop the system of multiprogramming in which a computer program will be able to perform multiple jobs as the same time. The introduction of multiprogramming was a major part in the development of operating system because it allowed a CPU to be busy nearly 100 percent of the time that it was in operating. Another major development during the third generation was the phenomenal growth of minicomputers, starting with the DEC PDP – 1 in 1961. The PDP – 1 had only 4k of 18 bit words, but at $1,20,000 per machine. It sold like hotcakes. These micro computers help create a whole new industry and development of more PDP's. Thus PDP's helped lead to the creation of personal computers which are created in the fourth generation.

4. The Fourth Generation (1980 – Present Day) : The fourth generation of operating system saw the creation of personal computing. Although these computers were very similar to the mini computers developed in the third generation, personal computers cost a very small fraction of what minicomputer cost. A personal computer was so affordable that is made it possible for a single individual could be able to own one for personal use while minicomputers were still at such a high price that only corporations could afford to have them. One of the major factors in the creation of personal computing was the birth of Microsoft and the windows operating system. Windows went on to become the largest operating system used. In technology today with release of Windows 95, Windows 98, Windows XP and their newest operating system Windows 7. Along with microsoft Apple is the other major operating system created in the 1980s. Windows development throughout the later years were influenced by the Macintosh. Today all of our electronics devices run off of operating system from our computers and smartphones, to ATM machines and motor vehicles and as technology advances so do operating systems.

Q 110. What are system calls in operating system ?

Ans. The interface between a process and an operating system is provided by system calls. In general, system calls are available as assembly language instructions. They are included in the manuals used by the assembly level programmers. System calls are usually made when a process in user mode requires access to a resource. Then it requires the kernal to provide the resource via a system call. The following figure represents the execution of the system call.



As can be seen from above diagram, the processes execute normally in the user mode until a system call interrupts this. Then the system call is executed on a priority basis in the kernel mode.

After the execution of the system call, the control returns to the user mode and execution of user processes can be resumed. In general system calls are required in the following situations :

- It a file system requires the creation or delection 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.

Q 111. What are the various types of system calls?

Ans. There are mainly five types of system calls. These are explained as follows :

1. Process control : These system calls deal with processes such as process creation, process termination etc.

2. File management : The system calls are responsible for device manipulation such as creation a file, reading a file, writing into a file etc.

3. Device managements : These system calls are responsible for device manipulation such as reading from devices buffers, writing into device buffers etc.

4. Information maintenance : These system calls handle information and its transfer between the operating system and the user program.

5. Communication : The system calls are useful for interprocess communication. They also deal with creating and delecting a communication connection.

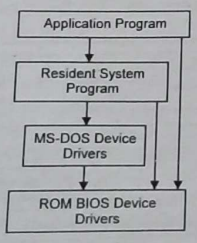
Q 112. What is a virtual machine ? What are its uses ?

Ans. A virtual machine is an operating system or application environment that is installed on software, which imitates dedicated hardware. The end user has the same experience on a virtual machine as they would have on dedicated hardware.

VMs have multiple uses, but in general they are deployed when the need for different operating systems and processing power are needed for different applications running simultaneously. For example If an enterprise wants to test multiple web servers and small databases at the same time. Similarly, If an enterprise wants to use the same server to run graphics intensive gaming software and customers service database.

Q 113. Explain structure of operating system.

Ans. 1. Simple structure :

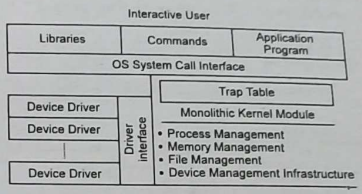


- *MS–DOS, application may by pass the operating system*
- *Operating systems such as MS–DOS and the original UNIX did not have well defined structures.*

- There are no CPU execution mode and so errors in applications could cause the whole system to crash.

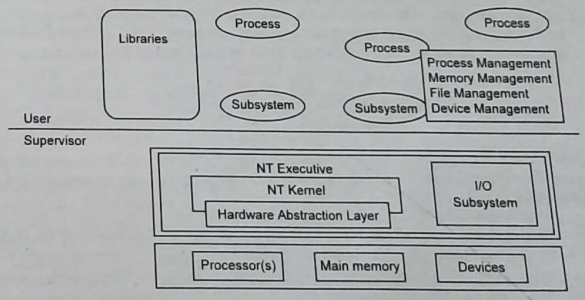
2. Monolithic Approach :

- Functionality of the OS is invoked with simple function calls within the kernel which is one large program.
- Device drivers are loaded into the running kernel and become part of the kernel.



A monolithic kernel, such as linux and other unix systems

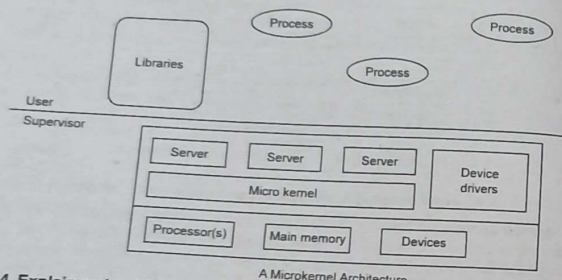
3. Layered approach : This approach breaks up the operating system into different layers. This allows implementation to change the inner working and increases modularity. As long as the external interface of the routines don't change, developers have more freedom to change the inner workings of the routines. With the layered approach, the bottom layer is the hardware, while the highest layer is the user interface. The main advantage of this approach is simplicity of construction and debugging but its main disadvantage is that the OS tends to be less efficient than other implementation.



The Microsoft windows NT operating system. The lowest level is a monolithic kernel but many OS components are at a higher level but still part of the OS.

4. Microkernels : This structures the operating system by removing all non essential portions of the kernel and implementing them as system and user level programs. Generally they provide minimal process and memory management and a communication facility. The benefits of the microkernal are as follows :

1. Extending the operating system becomes much easier.
2. Any changes to the kernel tend to be fewer, since the kernel is smaller.
3. The microkernel also provides more security and reliability. Here main disadvantage is poor performance due to increased system overhead from message passing.



A Microkernel Architecture

Q 114. Explain major difference between system and user mode. **(PTU, May 2014)**

Ans. Kernel mode, also referred to as system mode

(1) Processes in user mode can access their own instructions and data but not kernel instructions and data. Processes in Kernel mode can access kernel and user addresses.

(2) Some machine instructions are privileged and result in an error when executed in user mode.

- The operating system keeps internal records to distinguish the many processes executing on the system. The kernel distinguish between processes A,B,C and D on the horizontal axis, and the hardware distinguishes the mode of execution on the vertical axis.

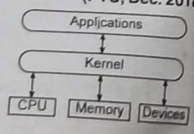
	Processes			
	A	B	C	D
Kernel mode	K			K
User mode		U	U	

- The system executes in one of two modes, the kernel runs on behalf of a user process. The kernel is not a separate set of processes that run in parallel to user processes, but it is part of each user process.

Q 115. What is the Kernel in operating system ? **(PTU, Dec. 2018)**

Ans. A kernel is the central part of an operating system. It manages the operations of the computer and the hardware must notably memory

- There are two type of kernel
- **Micro Kernel :** Which only contains basic functionality.
- **Monolithic Kernel :** Which contains many Device drivers.



Q 116. Under what circumstances, it is better using time-sharing system rather than a personal computer or single-user workstations. **(PTU, Dec. 2015)**

Ans. When there are few other users, the task is large, and the hardware is fast, time-sharing makes sense. The full power of the system can be brought to bear on the user's problem. The problem can be solved faster than on a personal computer. Another case occurs when lots of other users need resources at the same time. A personal computer is best when the job is small enough to be executed reasonably on it and when performance is sufficient to execute the program to the user's satisfaction.

Q 117. What are the main parts of the Unix Operating system ? **(PTU, Dec. 2015)**

Ans. 1. UNIX Kernel 2. Shell 3. Tools and applications

Q 118. What is software trap and how is it used in operating system design ? **(PTU, May 2016)**

Ans. A software trap occurs on any of the following conditions :

- A system call is executed.
- A page fault occurs.
- A privileged instruction is executed in non privileged mode.
- A memory protection violation occurs. To handle a software trap from a guest operating system.
- The trap goes to the host OS kernel.
- The host OS kernel determines that the trap came from a virtual machine and passes the trap to the VMM.

Q 119. Write two advantages of Windows based Operating System. **(PTU, May 2017)**

Ans. Stability : Windows 7 is the most stable Windows operating system.

Software Library : Windows O/S has the largest sized software library than any other O/S.

Q 120. Differentiate between UNIX and Windows based operating systems. **(PTU, Dec. 2017 ; May 2018, 2015)**

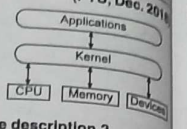
Ans.

UNIX	Window based OS
1. Unix is a command line. User interface operating system.	1. Window is a graphical user interface operating system.
2. Unix is a command based.	2. Window is a menu based.
3. This feature is absent in unix.	3. Window is a event driven.
4. Unix support following file system i.e. STD.ERR, STD.IO	4. Window support FAT32 and NTFS file system.
5. In term of security Unix is more secure.	5. In term of security window base OS is less sccure than Unix.
6. Plug and play features is not available in Unix	6. Window support plug and play feature.
7. Unix is a free source operating system.	7. Window is a licensed operating system.
8. Unix is a multithreading and multitasking operating system.	8. Window is not a multiprocessing operating system.
9. Unix is customizable.	9. Window is not customizable.
10. Unix can be booted from eithes primary or logical portion.	10. Window must boot from the primary position.
11. In Unix file names are case-sensitive.	11. In windows file names are not case sensitive.

Q 121. What is the Kernel in operating system ? **(PTU, Dec. 201**

Ans. A kernel is the central part of an operating system. It manages the operations of the computer and the hardware must notably memory and CPU time. There are two type of kernel

- **Micro Kernel :** Which only contains basic functionality.
- **Monolithic Kernel :** Which contains many Device drivers.



Q 122. Explain the Linux 'cd' command options along with the description ? **(PTU, Dec. 201**

Ans. The 'cd' command also known as change directory is an command line OS shell comman used to change the current working directory in operating system such as LINUX, UNIX, DOS Following are some options used along with 'cd' command

- cd by itself cd ~ – will always put you in your home directory.
- cd. – will leave you in the same Directory you are currently in.
- cd username – will put you in username's home directory.
- cd dir (without a/) will put you in a subdirectory.
- cd.. – will move you up one directory.
- cd- – will switch you to the previous directory.

Q 123. What are the advantages and disadvantages of multiprocessor systems ? **(PTU, Dec. 2018**

Ans. Advantages of multiprocesser system are as following :

(i) It improves the performance of the computer by allowing parallel processing of segments o program. Better performance is direct!y reflected by increased throughput and lowered turnaround t such system.

(ii) It helps in utilising all the system resources more efficiently and does not allow task to wa unnecessarily long.

(iii) If one of the CPU fails, the other can take its place till the damage corrected. This provide an inbuilt backups.

Disadvantages of Multiprocessing system :

(i) It require more skills and technical know how to develop such hardware organisation.

(ii) A large anount of memory will be required because more CPU can consume more Dat simultaneously.

(iii) Due to the special skill required to develop such system and special hardware requirements these system are expensive.

❑❑❑

Chapter 2

Processes

Contents

Definition, Process Relationship, Different states of a Process, Process State transitions, Process Control Block (PCB) Context swtiching.

Thread : Definition, Various states, Benefits of threads, Types of threads, Concept of multithreads.

Process Scheduling : Foundation and Scheduling objectives, Types of Schedulers, Scheduling criteria, CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time ; Scheduling algorithms : Pre-emptive and Non-pre-emptive, FCFS, SJF, RR ; Multiprocessor scheduling : Real Time Scheduling : RM and EDF.

POINTS TO REMEMBER

- A process is a program in execution. Our command interpreter is a process.
- Process states are new, ready, waiting, running, terminate, etc.
- CPU scheduling is the task of selecting a waiting process from the ready queue and allocating the CPU to it. The CPU is allocated to the selected process by the dispatcher.
- First come first served scheduling is the simplest scheduling algorithm but it can cause short processes to wait for very long processor.
- Shortest job first scheduling is provably optimal providing the shortest average waiting time.
- Round Robin scheduling is more appropriate for a time shared system RR scheduling allocates the CPU to the first process in the ready queue for q units, where q is the time quantum.
- The FCFS algorithm is non-preemptive. The RR algorithm is preemptive. The SJF and priority algorithm may be either preemptive or non-preemptive.
- Long term scheduling is the selection of processes to be allowed to contend for the CPU. Normally long term scheduling is highly influenced by resource allocation considerations, especially memory management.
- Short term scheduling is the selection of one process from the ready queue.
- Each process is represented in the operating system by a process control block also called task control block.
- The states of PCB are process state, program counter, CPU register, CPU scheduling information, memory management information.
- There are various different concurrency control schemes to guarantee serializability by either delaying an operation associated with it are executed to completion or now are performed.

- When several transactions overlap their execution the resulting may no longer be equivalent to an execution where these transactions executed automatically.
- Each process is represented by a PCB and the PCB can be linked together to form ready queue
- When several processes access and manipulate the same data concurrently and the outcome of the execution depends on the particular order in which the access takes place is called each condition.

QUESTION-ANSWERS

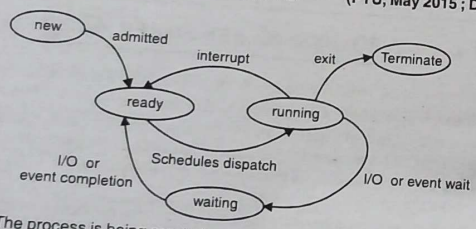
Q 1. Define process. **(PTU, May 2015)**

Ans. A process is a program in execution. The environment we interact with, when we use a computer is built out of processes.

- Our command interpreter is a process.
- When we execute a program, we just compiled, the operating system generates a process to run the computer.

Q 2. Define process states and their transitions. **(PTU, May 2015 ; Dec. 2019, 2003)**

Ans.



- **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 to a process.

Q 3. Explain operations on process.

Ans. A process is usually sequential and consists of a sequence of actions that take place one at a time. The set of operations on processes are :

1. Create a process
2. Destroy a process
3. Run a process
4. Suspend a process
5. Get process information
6. Set process information.

Q 4. Define CPU scheduling.

Ans. It is the process of determining which processes will actually run when there are multiple runable processes.

Q 5. Define CPU scheduler.

Ans. The scheduler will always switch the CPU to another process when one is busy with the input/output operation. This is actually the fundamental way in which multi-programming should work. The basic aim that scheduler should achieve, is that, CPU should never be allowed to sit idle.

- When process switches from running to blocked state.
- When process switches from running to ready state.
- When process switches from blocked to ready state.
- When a process terminates.

Q 6. Explain the difference between a program and a process. **(PTU, Dec. 2015, 2009 ; May 2009)**

Ans. A process is more than a program code. A process is an active entity as opposed to program which is considered to be a passive entity. Program is an algorithm expressed in some suitable notation, being a passive, a program is only a part of process. Process on the other hand includes :

1. Current value of program counter
2. Contents of the processor registers
3. Value of the variables.

Q 7. Explain types of CPU scheduling.

Ans. There are two types of scheduling algorithms : Pre-emptive or non-preemptive.

1. Non-preemptive : Once the CPU has been allocated to a process, the process keeps the CPU until its termination or its transition to the blocked state. This means that once CPU is allocated to a process, this process can use the CPU for its own execution till it willingly surrenders or leaves the CPU.

2. Preemptive scheduling : Here, even if the CPU has been allocated to a certain process, it can be snatched from this process any time either due to time constraint or due to priority reasons.

Q 8. Explain CPU scheduling criteria. **(PTU, Dec. 2018 ; May 2018, 2017)**

Ans. The possible measures are :

1. CPU utilization : CPU should remain as busy as possible.

2. Throughput : Throughput is defined as the number of processes that completed per unit of time and it should be as high as possible.

3. Turnaround time : Turnaround time is defined as interval between the time of submission and completion of jobs.

4. Waiting time : It is the sum of the time intervals for which the process has to wait in the ready queue.

5. Response time : In an interactive system, response time is the best metric. It is defined as the time interval between the job submission and the first response produced by the job.

Q 9. Explain dispatcher.

Ans. It is the component of CPU scheduling. It is a program responsible for assigning the CPU to the process, which has been selected by the short-term scheduler.

Q 10. Distinguish between Preemptive and Non-preemptive scheduling policies. **(PTU, Dec. 2018, 2010, 2009, 2005, 2003 ; May 2014, 2012, 2010, 2004)**

Ans. Preemptive scheduling : Preemptive scheduling incurs a cost. Consider the case of two processors sharing data. One may be in the midst of updating the data when it is preempted and the

second process is run. The second process may try to read the data, which are currently in an inconsistent state.

Non-preemptive : Non-preemptive scheduling is strictly unliked in computer. Because in a centre there is one server and other works like client. If one system is using server then other may be not be able to use server. And for some important preemptive we need to preemptive currently.

Q 11. Write about CPU Scheduling Algorithm. **(PTU, Dec. 2006 ; May 2004)**

OR

What is CPU scheduling? What is its need? List various scheduling algorithms. **(PTU, May 2018 ; Dec. 2011, 2009)**

OR

Discuss different factors which are taken into account while selecting a CPU scheduling algorithm. **(PTU, Dec. 2010)**

Ans. CPU scheduling algorithms : CPU scheduling deals with the problem of deciding which of the processes in the ready queue is to be allocated the CPU. These are :

1. First come, first served : In this, the process that requests the CPU first is allocated the CPU first. When a process enters the ready queue, its PCB is linked on to the tail of the queue. When the CPU is free, it is allocated to the process at the head of the queue.

2. Shortest job first : This algorithm associates with each process the length of the latter's next CPU burst. When the CPU is available, it is assigned to the process that has the smallest next CPU burst. If two processes have the same length next CPU brust, FCFS scheduling is used to break the tie.

3. Priority scheduling : A priority is associated with each process and the CPU is allocated to the process with the highest priority. Equal priority processes are scheduled in FCFS order.

4. Round Robin scheduling : It is designed for timesharing systems. A small unit of time called a time quantum is defined. The ready queue is treated as circular queue. The CPU scheduler goes around the ready queue, allocating the CPU to each process for a time interval of upto 1 time quantum.

5. Multilevel queue scheduling : It partitions the ready queue into several separate queues. The processes are permanently assigned to one queue, generally based on some property of the process, such as memory size, process priority or process type.

6. Multilevel feedback queue scheduling : It allows a process to move between queues. If a process uses too much CPU time, it will be moved to a lower priority queue. Similarly, a process that waits too long in a lower priority queue may be moved to a higher priority queue.

Q 12. What would be the effect, using the FCFS scheme, if the running process got stuck in an infinite loop? **(PTU, Dec. 2003)**

Ans. Using the FCFS scheme, if the running process got stuck in an infinite loop the system got stuck in deadlock as in FCFS scheme, the operating system is allocated to the first process and it remains to that process till that pocess is not completed its execution. If execution of this process completes, then the OS is allocated to next process. But if, the process to which OS allocated is stuck into infinite loop, then obviously the next processes will enter in waiting queue for OS allocation.

Q 13. Advantages of first come first served scheduling.

Ans. Advantages :

1. It is simple to understand and code.
2. Suitable for batch systems.

Disadvantages :

1. Waiting time can be larger if short requests wait behind the long ones.
2. It is not suitable for timesharing systems where it is important that each user should get the CPU for an equal amount of time interval.

Q 14. Advantages and disadvantages of shortest-job-first scheduling.

Ans. Advantages : This is usually considered to be an optimal algorithm, as it gives the minimum average waiting time.

Disadvantages : The problem is to know the length of time for which CPU is needed by a progress.

A prediction formula can be used to predict the amount of time for which CPU may be required by a process.

Q 15. What are various entries in process control block? **(PTU, May 2004)**

Ans. Each process is represented in the operating system by a PCB also called task control block. A PCB contains many pieces of information with a specific process including :

1. Process state
2. Program counter
3. CPU register
4. CPU scheduling information
5. Memory management information.

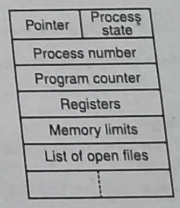
Q 16. What is Process Control Block? **(PTU, Dec. 2019, 2010, 2009 ; May 2010, 2007)**

OR

What is process control block? Explain the contents in process control block. **(PTU, Dec. 2016, 2012, 2008, 2005, 2004 ; May 2015, 2014, 2004)**

Ans. Each process is represented in the operating system by a process control block also called task control block. A PCB is shown in figure. It contains many pieces of information associated with a specific process including.

1. Process state
2. Program counter
3. CPU register
4. CPU scheduling information
5. Memory management information.



1. Process state : The state may be new, ready, running, waiting, halted.

2. Program counter : The counter indicates the address of next instruction to be executed for this process.

3. CPU registers : The registers vary in number and type, depending upon the computer architecture. They include accumulators, input registers, TR and so on.

4. CPU scheduling information : This information includes a process priority pointers to scheduling queues.

5. Memory management information : This information may include such information as the value of base and limit registers.

6. Accounting information : This information includes the amount of CPU and real time used, time limits, account numbers.

7. Input/Output status information : It includes the list of input/output devices allocated to the process.

Q 17. Comment on principal disadvantages of each of the following : FCFS, SJF, Round Robin. **(PTU, Dec. 2003)**

Ans. Disadvantages of FCFS :

1. Waiting time can be large if short requests wait behind the long ones.
2. It is not suitable for timesharing systems.

Disadvantages of SJF : The problem is to know the length of time for which CPU is needed by a process.

Disadvantages of Round Robin : Performances depend heavily on the size of time quantum.

Q 18. Explain shortest-job scheduling algorithm. **(PTU, Dec. 2017)**

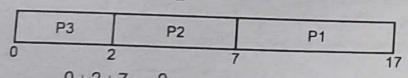
Ans. "CPU is allocated to the process with least CPU-burst time."

Amongst the processes in the ready queue, CPU is always assigned to the process with least CPU burst time.

If there are 2 processes with same CPU burst, the one which arrived first, will be taken up first by the CPU.

Example

Process	CPU-burst
P_1	10
P_2	5
P_3	2



Average waiting time = $\frac{0+2+7}{3} = \frac{9}{3} = 3$ milliseconds.

Q 19. Explain advantages and disadvantages of multilevel feedback queue algorithm. **(PTU, Dec. 2017)**

Ans. Advantages :

1. Multilevel feedback is more flexible.
2. A process that waits too long in a lowerpriority queue may be moved to a higher priority queue, this form of aging prevents starvation.
3. It allows to process to move between queues. This is fair for input/output bound processes, they have to wait too long.

Disadvantages :

1. It requires some means of selecting values for all the parameters to define the best scheduler.
2. It is the most complex algorithm.

Q 20. Difference between static and dynamic scheduling.

Ans. Static scheduling was primarily to minimize execution time, dynamic scheduling in a multiprocessor system seeks to :

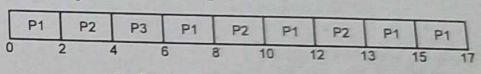
(i) Maximize resource utilization
(ii) Maximize processor utilization
(iii) Maximize execution time.

Q 21. What is round robin scheduling? Explain taking any example. Can it be useful for a single user system? If yes, then explain. If no, then why not? **(PTU, May 2013)**

Ans. It gives response to the users in a responsable time is known as Round-Robin scheduling algorithm. The basic purpose of this algorithm is to support timesharing systems. This algorithm is similar to FCFS algorithm but now it is a preemptive FCFS scheduling.

Example

Process	CPU burst time
P_1	10
P_2	5
P_3	2

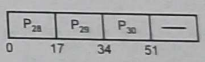


Average waiting time = $\frac{7+8+4}{3} = \frac{19}{3} = 6.33$ milliseconds

Waiting time of $P_1 = 0 + (6-2) + (10-8) + (13-12) = 4+2+1 = 7$
Waiting time of $P_2 = 2 + (8-4) + (12-10) = 2+4+2 = 8$
Waiting time of $P_3 = 4$.

No it can't be useful for a single user system because in this Round Robin time sharing system is used. In RR time quantum is given i.e. process is preempted from one user according to time quantum and so it is not useful.

e.g.



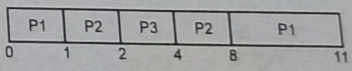
Multiple users.

Q 22. Explain priority scheduling algorithm.

Ans. Each process in the system if given a priority, then the scheduling must be done accordingly to the priority of each process. A higher priority job should get CPU whereas lower priority job can be made to wait.

Example :

Process	Burst time	Priority	Arrival time
P_1	10	3	0
P_2	5	2	1
P_3	2	1	2



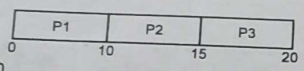
Average waiting time = $\frac{0+(8-1)+1+(4-2)+2}{3}$

$= \frac{7+1+2+2}{3} = \frac{12}{3} = 4$ milliseconds.

Q 23. Explain first come first served scheduling algorithm.

Ans. It is the simplest of all the scheduling algorithms. The major concept is allocate the CPU in the order in which the process arrives. It assumes that ready queue is managed as FIFO (first in first out).

Process	CPU-Burst
P_1	10
P_2	5
P_3	5



Waiting time for $P_1 = 0$
Waiting time for $P_2 = 10$
Waiting time for $P_3 = 15$

Average waiting time = $\frac{0+10+15}{3} = \frac{25}{3} = 8.33$ milliseconds.

Q 24. In what way a shortest job first scheduling (SJF) just a particular form of priority scheduling. **(PTU, May 2007)**

Ans. Shortest job first is a particular way of priority scheduling. When CPU burst time of process is used to calculate the priorities of processes. With less CPU burst time, a process will be given higher priority as compared to process with more CPU burst time.

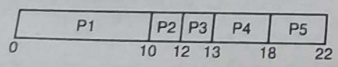
Q 25. Find the waiting time of :

(i) FCFS (ii) SJF (iii) RR (iv) Priority scheduling. The processes are assumed to be arrived in order P1, P2, P3, P4, P5.

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

(PTU, Dec. 2005)

Ans. (i) FCFS : *Gantt Chart*

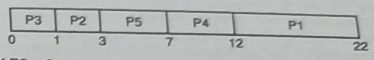


Waiting time of P1 = 0 milliseconds
Waiting time of P2 = 10 milliseconds

Waiting time of P3 = 12 milliseconds
Waiting time of P4 = 13 milliseconds
Waiting time of P5 = 18 milliseconds

Average waiting time = $\frac{0+10+12+13+18+22}{5}$ = 25 milliseconds.

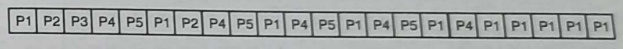
(ii) SJF : Gantt chart of SJF is :



Waiting time of P3 = 0
Waiting time of P2 = 1
Waiting time of P5 = 3
Waiting time of P4 = 7
Waiting time of P1 = 12

Average waiting time = $\frac{0+1+3+7+12}{5}$ = 4.6 milliseconds.

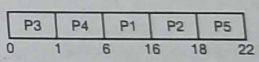
(iii) Round Robin : quantum is 1



If we use a time quantum of 1 millisecond, then process P_1 gets the first 1 millisecond. Since it requires another 9 milliseconds, it is preempted after the first time quantum and CPU is given to next quantum or process.

Average waiting time = $\frac{153}{5}$ = 30.6 milliseconds

(iv) Priority scheduling :



$\frac{0+1+6+16+18}{5}$ = 8.2 milliseconds.

Q 26. Explain concurrency. **(PTU, May 2009)**

Ans. Multitasking operating systems, especially real-time operating systems, need to maintain the illusion that all tasks running on top of them are all running at the same time, even though only one or few tasks really are running at any given moment due to the limitations of the hardware the operating system is running on. Such multitasking is fairly simple. However, when several tasks try to use same resource, it can lead to confusion and inconsistency. The task of concurrency computing is to solve that problem.

Q 27. How concurrency can be achieved in CPU and input/output devices?

Ans. The concurrent execution of CPU and input/output devices are :

(i) Hardware parallelism : CPU can be computing while one or more input/output devices are

running at the same time. This is the one, which is actually implemented, in multiprogramming environment.

(ii) Pseudo parallelism : Rapid switching of the CPU among processes is known as pseudo parallelism.

(iii) Real parallelism : Actual parallelism is achieved by having multiple CPUs, each executing a different process or part of a process simultaneously.

Q 28. Differentiate between heavyweight and lightweight processes. **(PTU, Dec. 2016)**

Ans. Difference between heavyweight and lightweight processes :

A heavy-weight process is one that is typically independently managed by the target operating system and so encompasses its own address space.

A light weight process usually lives with in a single operating system process along with other light weight processes, which share the same address space.

Q 29. What do you mean by preempting a process? **(PTU, Dec. 2011)**

Ans. Preempting a process means that the dispatcher removes the process from the processor and put it either into the ready queue, or puts it into the I/O queue if the process was interrupted by an I/O event/system call.

Q 30. What is short-term scheduler? **(PTU, Dec. 2004)**

Ans. Short-term scheduler : It selects among the processes that are ready to execute and allocate the CPU to one of them. Short-term scheduler on the other hand, must have to work very often. A process must be executing for only a very short duration before it gets blocked.

Q 31. What is CPU scheduling? Explain FCFS, SJF and priority scheduling with the advantages and disadvantages. **(PTU, May 2007)**

Ans. CPU scheduling : CPU scheduling deals with the problem of deciding which of the processes in the ready queue is to be allocated to the CPU.

FCFS : FCFS acts like normal queues. The first process to arrive is the first one to get CPU. After completion of processing of first process, CPU will be allocated to next process in the queue.

Advantages :

1. Simple
2. Easy to understand
3. First come first served.

Disadvantages :

1. This is non-preemptive, the process will run until it finishes.
2. Because of non-preemptive, short processes which are at the back of the queue have to wait for the long process at the front to finish.

Shortest job first : This algorithm associates with each process the length of the latter's next CPU burst. When the CPU is available, it is assigned to the process that has the smallest next CPU burst. If two processes have the same length next CPU burst, FCFS scheduling is used to break the tie.

Priority Scheduling : In this each is given a priority and higher priority methods are executed first come first served.

Advantages : An important process can be given high priority and it will execute first as highest priority.

Disadvantages : A process sometimes becomes starved, live blocked. One way to fix this is by aging priority, so that the longer process has waited for CPU time, higher its priority is.

Q 32. Comment on the following context of interacting processes in unix : "Implementation of wait call using PCB data structures". **(PTU, Dec. 2007)**

Ans. In unix, each process is identified by its process identifier, which is a unique integer. A new process is created by the fork system call. This mechanism allows the parent processor to communicate easily with its child process. Both processes continue execution at the instruction after the fork system call. The parent creates a child process using the fork system call. The parent can create more children or if it has nothing else to do while the child runs, it can issue a wait system call to move itself off the ready queue until the termination of the child. The child process overlays its address space with the unix command /bin/ls using the child process to complete with the wait system call. When the child process completes, the parent process resumes from the call to wait where it completes using the exit system call. The wait system call returns the process identifier of a terminated child, so that the parent can tell which of its possibly many children has terminated.

Q 33. What is meant by process scheduling? **(PTU, Dec. 2004)**

Ans. Scheduling is a concept of computer multitasking, multiprocessing operating system and real time operating systems.

Process scheduling : Select a process from ready queue for execution.

Evaluation metrices :

- ❑ CPU/device utilization
- ❑ System throughput.

Q 34. What do you mean by scheduling? Explain with example the Shortest Job First (SJF) scheduling and Round robin scheduling algorithm. **(PTU, Dec. 2008)**

OR

List different scheduling algorithms. Explain Preemptive Shortest Job First (SJF) and Round robin scheduling algorithms with the help of suitable examples. **(PTU, May 2019 ; Dec. 2010, 2004)**

Ans. Scheduling : CPU scheduling deals with the problem of deciding which of the processes in the ready queue is to be allocated the CPU. These are :

1. First come, first served : In this, the process that requests the CPU first is allocated the CPU first. When a process enters the ready queue, its PCB is linked onto the tail of the queue. When the CPU is free, it is allocated to the process at the head of the queue.

2. Shortest job first : This algorithm associates with each process the length of the latter's next CPU brust. When the CPU is available, it is assigned to the process that has the smallest next CPU brust. If two processes have the same length next CPU brust, FCFS scheduling is used to break the tie.

3. Priority scheduling : A priority is associated with each process and the CPU is allocated to the process with the highest priority. Equal priority processes are scheduled in FCFS order.

4. Round Robin scheduling : It is designed for time sharing systems. A small unit of time called a time quantum is defined. The ready queue is treated as circular queue. The CPU schedular goes around the ready queue, allocating the CPU to each process for a time interval of upto 1 time quantum.

5. Multilevel queue scheduling : It partitions the ready queue into several separate queues. The processes are permanently assigned to one queue, generally based on some property of the process, such as memory size, process priority or process type.

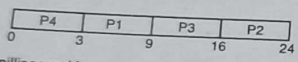
6. Multilevel feedback queue scheduling : It allows a process to move between queues. If a

process uses too much CPU time, it will be moved to a lower-priority queue. Similarly, a process that waits too long in a lower priority queue may be moved to a higher priority queue.

A different approach to CPU scheduling is the shortest job first scheduling algorithm. This algorithm associates with each process the length of the latter's next CPU burst. When the CPU is available, it is assigned to the process that has the smallest next CPU burst. If two processes have the same length next CPU burst, FCFS scheduling is used to break the tie. As an example consider the following set of processes, with the length of the CPU burst time given in milliseconds.

Process	Burst time
P1	6
P2	8
P3	7
P4	3

Using SJF scheduling we would schedule these processes according to the following Gantt Chart.



The waiting time is 3 millisecond for process P1, 16 milliseconds for process P2, 9 miliseconds for process P3 and 0 milliseconds for process P4. Thus the average waiting time is (3 + 16 + 9 + 0)/ 4 = 7 milliseconds. If we were using the FCFS scheduling scheme, then the average waiting system waiting time would be 10.25 milliseconds.

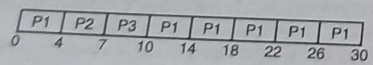
Round Robin Scheduling : It is designed especially for the time sharing system. It is similar to FCFS scheduling, but preemption is added to switch between processes. A small unit of time, called a time quantum is defined. A time quantum is generally from 10 to 100 milliseconds. The ready queue is treated as a circular queue. The CPU schedular goes around the ready queue, allocating the CPU to each process for time interval of upto 1 time quantum.

To implement RR scheduling we keep ready queue as a FIFO queue of processor. New processes are added to the tail of the ready queue. The CPU schedular picks the first process from the ready queue, set a timer to interrupt after 1 time quantum and dispatches the process.

The average waiting time under the RR policy, however is often quite low. Consider the following set of processor that arrive at time 0, with the length of the CPU burst time given in milli seconds.

Process	Burst time
P1	24
P2	3
P3	3

If we use time quantum of 4 milliseconds, then process P1 gets the first 4 milliseconds, since it requires another 20 milliseconds, it is preempted after the first time quantum and the the CPU's given to the next process in the queue, process P2. Since process and P2 does not need 4 milliseconds, it quits before its time quantum expires. The CPU is then given to the next process P3. Once each process has received 1 time quantum, the CPU is returned to process P1 for an additional time quantam. The resulting RR schedule is



The average waiting time is $\frac{17}{3}$ = 5.66 milliseconds.

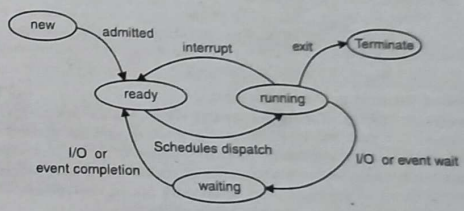
Q 35. Explain the concept of process. Discuss various process states.

(PTU, Dec. 2004)

Ans. Initially, a process is a program in execution. A process is more than the program code, which is sometimes known as the text section. It also includes the current activity, as represented by the value of the program counter and the contents of the processor's registers. In addition a process generally includes the process stack, which contains temporary data, and a data section, which contains global activity.

Process State : As a process executes, it changes state. The state of a process is defined in part by the current activity of that process. Each process may be in one of the following states :

- **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 to a processor.
- **Terminated :** The process has finished execution.



Q 36. Round Robin scheduling.

(PTU, May 2012)

Ans. Round Robin : With respect to round robin scheduling, the time quantum to be large with respect to the control switch time. The performance of RR algorithm depends heavily on the size of the time quantum. At one extreme, if the time quantum is very small, the RR approach is called processor sharing and appears to the users as through each of n processes has its own processor running at 1/N the speed of the real processor.

Q 37. What is the disadvantage of FCFS scheduling algorithm?

(PTU, Dec. 2012)

Ans. 1. This is non preemptive, the process will run until it finishes.

2. Because of non-preemptive, short processes which are at the back of the queue have to wait for the long process at the front to finish.

Q 38. What you understand by process scheduling? Write a note.

(PTU, May 2012)

Ans. Scheduling is a key concept of computer multitasking, multiprocessing operating system and real time operating system designs.

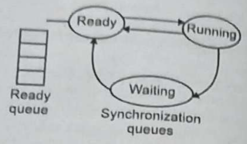
Process scheduling : Select a process from ready queue for execution.

Evaluation metrics :

- CPU/device utilization
- System throughout
- Waiting time
- Response time

Scheduling Policies :

- First come first served scheduling.
- Round Robin.
- Shortest job first.
- Priority scheduling.
- Multilevel queues.



Q 39. Write a note on Preemptive and non-preemptive scheduling. **(PTU, May 2012)**

Ans. Preemptive and non-preemptive scheduling :

Preemptive : The preemptive scheduling is prioritized, the highest priority process should always be the process that is currently utilized.

- It can cease to be ready to execute.
- It can get pre-empted by a high priority thread.
- It can explicity call a thread scheduling method such as wait or suspend.

Non-preemptive : When a process enters the state of running, the state of that process is not deleted from the scheduler until it finishes its service time.

Q 40. Write short notes on Two-phase locking. **(PTU, May 2008)**

Ans. Two-phase locking : One way to ensure serializability is to associate with each data item a lock and to require that each transaction follow a locking protocol that governs how locks are acquired and released. One protocol that ensures serializability is the two phase locking protocol. This protocol requires that each transaction issue lock and unlock requests in two phases.

- **Growing Phase :** A transaction may obtain locks, but may not release any lock.
- **Shrinking Phase :** A transaction may release locks but may not obtain any new locks.

Initially a transaction is in the growing phase. The transaction acquires locks are needed. Once the transaction release a lock, it enters the shrinking phase and no more lock request can be issued.

Two phase locking protocol ensures conflict serializability. It does not however ensure freedom from deadlock.

Q 41. What is the difference between turnaround time and response time? **(PTU, Dec. 2015, 2011)**

Ans. Turnaround time : Amount of time to execute a particular process.

Response time : Amount of time it takes from when a request was submitted until the first response is produced. It is also called Response Ratio.

Q 42. Discuss the need of process scheduling and types of schedulers. **(PTU, Dec. 2004)**

OR

[illegible] differences among short-term, medium-term, and long-term scheduling. **(PTU, Dec. 2011)**

Ans. Informally, a process is a program in execution. A process is more than the program code, which is sometimes known as the text section.

Process scheduling : The objective of multiprogramming is to have some process running at all times, so as to maximize the CPU utilization. The objective of time sharing is to switch the CPU among processes so frequently that users can interact with each program while it is running. A uniprocessor system can have only one running process. If more processes exist, the rest must wait until the CPU is free and can be rescheduled.

Schedulers : A process migrates between in the various scheduling queues throughout its life time. The operating system must select, for scheduling purposes, processes from these queues in some fashion. The selection process is carried out by the appropriate scheduler.

Types of schedulers :

- Short term scheduler
- Long term scheduler
- Medium term scheduler

Short term scheduling : The jobs that are ready to run immediately are kept in **Ready Queue.** Scheduling or choosing processes from this queue is called short term scheduling.

Long term scheduling : There might also be a queue of jobs on the disk that need to be loaded into memory and placed in the ready queue. Scheduling this transfer into the Ready Queue is called long term scheduling.

Medium term scheduling : Finally, some jobs may be moved out of the ready queue into some immediate queue. The purpose is to remove excess multiprogramming. Scheduling such jobs is called medium term scheduling because, presembly, these processes have priority over jobs in the long term queue.

Q 43. What is the difference between short term scheduler and long term scheduler? **(PTU, Dec. 2012)**

Ans. Long term scheduling performs a gate keeping function. It decides whether there's enough memory, or room, to allow new programs or jobs into the system.

The short term scheduler taps jobs from the "ready" line and gives them the green light to run. It decides which of them can have resources and for how long. The short term scheduler runs the highest-priority jobs first.

Q 44. What is a thread in operating system ?

Ans. A thread is a single sequence stream with in a process. Threads have same properties as of the process so they are called as light weight processes. Threads are executed one after another but gives the illusion as if they are executing in parallel. Each thread has different states. Each thread has.

1. A program counter
2. A register set
3. A stack space.

Threads are not independent of each other as they share the code, data, OS resources etc.

Q 45. Explain the similarities and dissimilarities between threads and processes?

Ans. Similarities between threads and processes :

1. Only one thread or process is active at a time.
2. Within process both execute sequential.
3. Both can create children.

Dissimilarities between threads and processes :

1. Threads are not independent, processes are
2. Threads are designed to assist each other, processes may or may not do it.

Q 46. What are the different types of threads ? Also explain their advantages and disadvantages.

Ans. The various types of threads are as follows :

1. User Level thread (ULT) : ULT is implemented in the user level library they are not created using the system calls. Thread switching does not need to call OS and to cause interrupt the Kernel. Kernel doesn't know about the user level thread and manages them as if they were single-threaded processes.

Advantages of ULT :

1. Can be implemented on an OS that doesn't support multithreading.
2. Simple representation since thread has only program counter, register set, stack space.
3. Simple of create since no intervention of kernel.
4. Thread switching is fast since no OS calls need to be made.

Disadvantages of ULT :

1. No or less co-ordination among the threads and kernel.
2. If one thread causes a page fault, the entire process blocks.

2. Kernel level thread (KLT) : Kernel knows and manages the threads. Instead of thread table in each process, the kernel itself has thread table (a master one) that keeps track of all the threads in the system. In addition kernel also maintains the traditional process table to keep track of the processes. OS kernel provides system call to create and manage threads.

Advantages of KLT :

1. Since kernel has full knowledge about the threads in the system scheduler may decide in the system scheduler may decide to give more time to processes having large number of threads.
2. Good for applications that frequently block.

Disadvantages of KLT :

1. Slow and inefficient.
2. It requires thread control block so it is an overhead.

Q 47. Explain the difference between process and thread.

Ans.

S.No.	Process	Thread
1.	Process is heavy weight or resource intensive	Thread is light weight, taking lesser resources than a process.
2.	Process switching need interaction with operating system.	Thread switching does not need to interact with operating system.
3.	In multiple processing environments, each process executes the same code but has its own memory and file resources.	All threads can share same set of open files, child processes.
4.	If one process is blocked, than no other process can execute until the first process is unblocked.	While one thread is blocked and waiting a second thread in the same task can run.
5.	Multiple processes without using threads use more resources.	Multiple threaded processes use fewer resources.
6.	In multiple processes each process operates independently of the others	One thread can read, write or change another thread's data.

Q 48. What are the various advantages of thread.

Ans. The various advantages of thread are as follows :

1. Threads minimize the context switching time.
2. Use of threads provides concurrency within a process
3. Efficient communication.
4. It is more economical to create and context switch threads.
5. Threads allow utilization of multiprocessor architectures to a greater scale and efficiency.

Q 49. What is dispatcher? How it is different from scheduler? **(PTU, Dec. 2013)**

Ans. Dispatcher : It is the component of CPU scheduling. It is a program responsible for assigning the CPU to the process, which has been selected by the short-term scheduler.

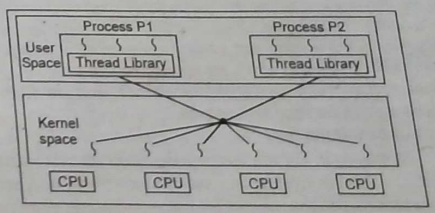
1. The dispatcher chooses which process to terminate. The scheduler chooses the process to use the CPU.
2. The dispatcher terminates resource access whereas the scheduler determine which process use which hardware resource.
3. The dispatcher dispatches, whereas the scheduler schedules the scheduler is used to create a process from a program.
4. The dispatcher works with process as they run. The scheduler is used to create a process from a program.
5. The dispatcher makes a process obey that schedule whereas the scheduler is creates a schedule of times a process may use the CPU.

Q 50. What is multithreading ? What are the various types of multithreading models?

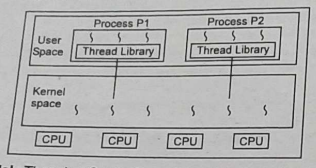
Ans. Some operating system provide a combined user level thread and kernel level thread facility. Salaris is a good example of this combined approach. In a combined system, multiple thread with in the same application can run is parallel on multiple processors and a blocking system call need not block the entire process. Multithreading models are three types :

1. Many to many relationships.
2. Many to one relationship.
3. One to one relationship.

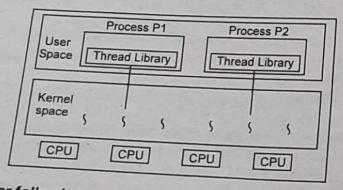
1. Many to Many Model : The many to many model multiplexes any number of user threads onto an equal or smaller number of kernel threads. The following diagram shows the many to many threading model where 6 user level threads are multiplexing with 6 kernel level threads. In this model, developers can create as many user threads as necessary and the corresponding kernel threads can run in parallel on a multiprocessor machine. This model provides the best accuracy on concurrency and when a thread performs a blocking system call, the kernel can schedule another thread for execution.



2. Many to one model : Many to one model maps many user level threads to one kernel level thread. Thread management is done in user space by the thread library. When thread makes a blocking system call, the entire process will be blocked. Only one thread can access the kernel at a time, so multiple threads are unable to run in parallel on multiprocessers. If the user level thread libraries are implemented in the operating system in such a way that the system does not support them then the kernel threads use the many to one relationship modes.



3. One to one model : There is a One to One relationship of user level thread to the kernel level thread. This model provides more concurrency than the many to one model. It also allows another thread to run when a thread makes a blocking system call. It supports multiple threads to execute in parallel on microprocessors. Disadvantages of this model is that creating user thread requires the corresponding kernel thread. OS/2 Windows NT and Windows 2000 use one to one relationship model.



Q 51. State whether following are true or false with justification :

(a) Multitasking is a kind of multiprogramming.

(b) Response times are more predictable in preemptive system than in non-preemptive system. **(PTU, Dec. 2013)**

Ans. (a) False, we can say that Multitasking is basically multiprogramming but in the context of a single user interactive environment, in which the operating system switches between several program in main memory so as to give the illusion that several are running at onces. The multitasking and multiprogramming often have differences which are context dependent.

(b) False, because in non-preemptive system response time are more predictable because incoming high priority jobs cannot displace waiting jobs.

Q 52. What is role of scheduler ? **(PTU, Dec. 2018)**

Ans. Scheduler : Scheduler in OS are the process which decides which task and process should be accessed and run at what time by the system resources. The scheduler in operating system are the algorithm which help in the system optimisation for maximum performance.

Q 53. What is the purpose of system call ? **(PTU, Dec. 2016)**

Ans. The purpose of system call is that system calls allow user-level processes to request services of the operating system.

- Process control
- File management
- Device management
- Information maintenance
- Communication between processes.

Q 54. Consider the following set of processes, with the length of the CPU-burst time give in milliseconds :

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

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

(a) Draw four Gantt charts illustrating the execution of these processes using FCFS, SJF, a non-preemptive priority (a smaller priority number implies a higher priority), and RR (quantum=1) scheduling.

(b) What is the turnaround time of each process for each of the scheduling algorithms in part a?

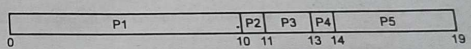
(c) What is the waiting time of each process for each of the scheduling algorithms in part a ?

(d) Which of the schedules in part a results in the minimal average waiting time (over all processes) ? **(PTU, May 2014 ; Dec. 2015, 2013)**

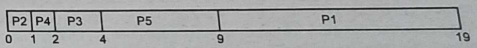
Ans. (The processes are assumed to have arrived in the order P1,P2,P3,P4,P5 all at time 0).

(a) The four Gantt charts are

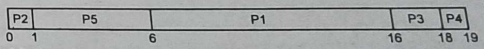
FCFS :



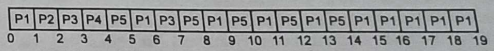
S.J.F.



Priority



Round Robin



(b) The turnaround time of each process for each of the scheduling algorithm in part a :

Turnaround time = Finish-time – Arrival-time

	FCFS	SJF	Priority	RR
P1	10	19	16	19
P2	11	1	1	2
P3	13	4	18	7
P4	14	2	19	4
P5	19	9	6	14

(c) The waiting time of each process for each of the scheduling algorithms in part a :

	FCFS	SJF	Priority	RR
P1	0	9	6	9
P2	10	0	0	1
P3	11	2	16	5
P4	13	1	18	3
P5	14	4	1	9
Average waiting time	9.6	3.2	8.2	5.4

(d) Ave.Wait	FCFS	SJF	Priority	RR
	9.6	3.2	8.2	5.4

So the answer is shortest Job first.

Q 55. What are the differences between load-time and run-time dynamic linking ?
(PTU, May 2016, 2014)

Ans. In load-time dynamic linking an application makes explicit calls to exported DLL functions like local functions. To use load-time dynamic linking provide a header (.h) file and an import library (.lib) file when you compile and link the application. When you do this, the linker will provide the system with the information that is required to load the DLL and resolve the exported DLL function locations at load time.

In run-time dynamic linking an application calls either the Load Library function or the LoadLibraryEx function to load the DLL at run time. After the DLL is successfully loaded, you use the GetProcAddress function to obtain the address of the exported DLL function that you want to call when you use run-time dynamic linking, you do not need an import library file.

Q 56. Show why and how is busy waiting supported in hardware ? **(PTU, May 2014)**

Ans. Busy waiting is a process is waiting for an event to occur and it does so by executing instructions.

It is a technique in which a process repeatedly checks to see if a condition is true, such as whether keyboard or a lock is available.

If a lock is busy, the thread keeps testing it until it changes value. In the process it consumes both processor and memory bandwidth. This overhead can be mitigated by the thread scheduler. In hardware - multithreaded core, the waiting thread may be deactivated, or its execution priority may be lowered. In some cases a lock may suspend the running thread (block it) so that it is removed from the list of ready-to-run threads in the core. In some machines this may mean deallocating the thread context.

The difficulty in implementing high-level synchronization operations in hardware is not the acquire or the release component but the waiting algorithm. Thus, it makes sense to provide hardware support for the critical aspects of the acquire and release method.

Q 57. Suppose that the following processes arrive for execution at the times indicated. Each process will run the listed amount of time. In answering the questions, use non-preemptive scheduling and base all decisions on the information you have at the time the decision must be made.

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

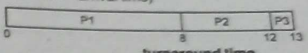
(a) What is the average turnaround time for these processes with the FCFS scheduling ?

(b) What is the average turnaround time for these processes with the SJF scheduling algorithm ?

(c) The SJF algorithm is supposed to improve performance, but notice that we chose to run process P1 at time 0 because we did not know that two shorter processes would arrive soon. Compute what the average turnaround time will be if the CPU is left idle for the first 1 unit and then SJF scheduling is used. Remember that processes P1 and P2 are waiting during this idle time, so their waiting time may increase. This algorithm could be known as future-knowledge scheduling. **(PTU, May 2014)**

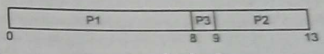
Ans. (a) Average turnaround time with FCFS : We need to draw the Gantt chart to find the finish time then calculate the turnaround time

(Turnaround time = finish time – arrival time)



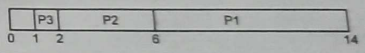
	turnaround time
P1	8 – 0 = 8
P2	12 – 0.4 = 11.6
P3	13 – 1 = 12
Average turn around time	10.53

(b) Average turnaround time with SJF :



P1	8 – 0 = 8
P2	13 – 0.4 = 12.6
P3	9 –1 = 8
Average turn around time	9.53

(c) Average turnaround time with future knowledge scheduling : We need to draw the Gantt chart to find the finish time then calculate the turnaround time



P1	14 – 0 = 14
P2	6 – 0.4 = 5.6
P3	2 – 1 = 1
Average turn around time	6.86

Q 58. What causes a process/thread to change the state ?
(i) From running to ready ?
(ii) From ready to running ?
(iii) From running to blocked ?
(iv) From blocked to ready ? **(PTU, May 2014)**

Ans. Steps :

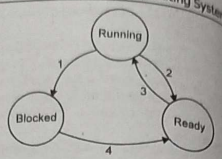
1. Process blocks for input.
2. Scheduler picks another process
3. Scheduler picks this process
4. Input becomes available.

(i) From running to ready.
Scheduler picks another process

(ii) From ready to running.
Scheduler picks this process

(iii) From running to blocked.
Process blocks for input.

(iv) From blocked to ready.
Inputs becomes available.



The process manager uses the state diagram to determine the type of service to provide to the process (process/thread).

1. If the process is in ready state, it is competing for the processor. The only transition out of the ready state is for the process to change to the running state when it is allocated the CPU.
2. If the process is in running state, it can complete execution, in which case the process manager will release the resources it holds and destroy the process. Alternatively, the process may request a resource when it is in the running state; for example, by requesting an I/O operation.

If a running process requests and receives a resource without having to wait for it, the process is allowed to continue the running state.

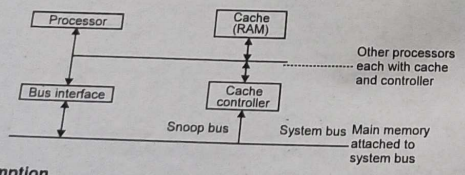
Otherwise, the process manager removes the process from the processor, marks its state as blocked, and notifies the resource manager that the process is waiting for units of its resource, The OS then calls the scheduler to allocate the processor to the next selected process in the ready state. From the blocked state, the process can transition to the ready state when the resource manager allocates the requested resource. The process is then once again competing for the CPU.

Q 59. Snoop bus mechanism. **(PTU, Dec. 2014)**

Ans. For a single bus structure, snoop bus mechanism often used.

In the snoop bus mechanism, a "bus watcher" unit with each processor/cache observes the transactions on the bus and in particular monitors all memory write operations. If a write is performed to a location which is cached locally, this copy is invalidated; needs a protocol.

Could invalid based upon only index (not compare tags)



Q 60. Preemption. **(PTU, Dec. 2014)**

Ans. Preemption is the act of temporarily interrupting a task being carried out by a computer system, without requiring its cooperation, and with the intention of resuming the task at a later time. Such a change is known as a context switch. It is normally carried out by a privileged task or part of the system known as a preemptive scheduler, which has the power to preempt, or interrupt, and later resume, other tasks in the system. Preemption as used with respect to operating systems means the ability of the operating system to preempt (i.e. stop or pause) a currently scheduled task in favour of a higher priority task. The resource being scheduled may be the processor or I/O, among others.

Q 61. Describe terms long-term, medium-term and short-term thread scheduling. **(PTU, May 2016)**

Ans. Long term scheduling : The long term scheduler decides which jobs or processes are to be admitted to the ready queue (in main memory).

Medium term scheduling : The medium term scheduler temporarily removes processes from main memory and places them in secondary memory or vice versa.

Short term scheduling : The short term scheduler is also known as CPU scheduler. It decides which of the ready in-memory processes is to be executed.

Q 62. What are the three important levels of scheduling ? **(PTU, Dec. 2015)**

Ans. 1. High-level scheduling or long term scheduling.
2. Medium level scheduling
3. Low-level scheduling.

Q 63. Compare round robin scheduling and multilevel queue scheduling ? **(PTU, Dec. 2014)**

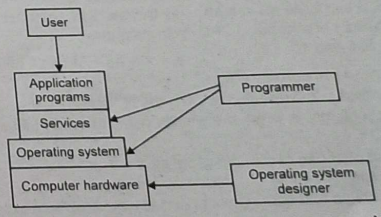
Ans. The multilevel scheduling policy combines priority-based scheduling and round robin scheduling to provide a good combination of system performance and response time. A multilevel scheduler maintains a number of ready queue. A priority and a time slice are associated with each ready queue, and round robin scheduling with time slicing is performed with in it.

Round Robin scheduling is especially designed for time-sharing system and is used for forground process

	Round-Robin	Multilevel Queue
CPU Utilization	High	High
Throughput	Medium	High
Turnaround time	Medium	Medium
Response time	High	Medium
	Premptive	Non-Preemptive

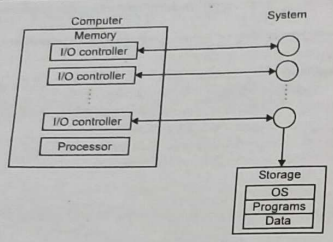
Q 64. Explain the role of Operating system as a resource manager. **(PTU, May 2018, 2015)**

Ans. Operating system as a resource manager : Computers have a number of resources such as I/O devices, CPU time, memory space, file storage space, etc. An operating system is a resource allocator and a resource manager. It decides that to which program which resource is to be allocated at a specific time and the time duration of the allocation.



An operating system interface between the programmer and the computer

A portion of an operating system is in the main memory (Kernel). The allocation of this memory is done both by the operating system and the memory management hardware.

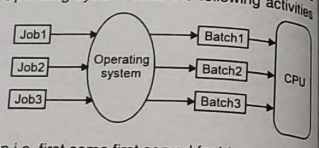


An operating system as a resource manager

Q 65. What is batch processing ? **(PTU, Dec. 2015)**

Ans. Batch processing is a technique in which operating system collects one programs and data together in a batch before processing starts. Operating system does the following activities related to batch processing :

- OS defines a job which has predefined sequence of commands, programs and data as a single unit.
- OS keeps a number of jobs in memory and executes them without any manual information.



- Jobs are processed in the order of submission i.e. first come first served fashion.
- When job completes its execution, its memory is released and the output for the job gets copied into an output spool for later printing or processing.

Q 66. Five jobs are in the ready queue waiting to be processed. Their estimated CPU cycles are as follows : 10, 3, 5, 6 and 2. Using SJN in what order they should be processed to minimize average waiting time ? **(PTU, May 2016)**

Ans. Five jobs are in the ready queue waiting to be processed. Their estimated CPU cycles are as follows 10, 3, 5, 6 and 2, so we can also say that the SJN (shortest job next) algorithm would schedule the jobs in order of ascending CPU cycle estimates :

2, 3, 5, 6 and 10

Q 67. Explain the term convoy effect. **(PTU, Dec. 2017)**

Ans. Convoy effect : Convoy effect refers to the phenomenon associated with the First Come First Serve (FCFS) algorithms, in which the whole operating system slows down due to few slow processor.

Q 68. What is the purpose of system program? **(PTU, Dec. 2017)**

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

Q 69. What is meant by the state of the process ? **(PTU, Dec. 2018)**

Ans. As a Process execute, it changes state. The state of a process is defined in part by the current activity of that process. Each process may be in one of the following states :

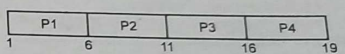
- New
- Running
- Waiting
- Ready
- Terminated

Q 70. Consider the following four processes, with the length of the CPU burst time given in milliseconds.

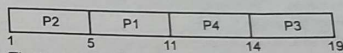
Process	Arrival Time (ms)	Burst Time (ms)
P1	1	6
P2	1	5
P3	2	5
P4	2	3

Find Average Waiting Time and Turnaround time for given Process using FCFS and SJF Algorithms ? **(PTU, Dec. 2018)**

Ans. FCFS



SJF



Turnaround Time : The turnaround time of each process for each of the algorithm in calculates as

Turnaround time = Finish Time – Arrival Time

For FCFS

Turnaround time for P_1 Process = 6 – 1 = 5ms
Turnaround time for P_2 Process = 11 – 1 = 10ms
Turnaround time for P_3 Process = 16 – 2 = 14ms
Turnaround time for P_4 Process = 19 – 2 = 17ms

$$\text{Averge Turnaround Time} = \frac{P_1+P_2+P_3+P_4}{4} = \frac{5+10+14+17}{4}$$

$$= \frac{46}{4} = 11.5\text{ms}$$

For SJF

Turnaround time for P_1 Process = 11 – 1 = 10ms
Turnaround time for P_2 Process = 5 – 1 = 4ms
Turnaround time of P_3 Process = 19 – 2 = 17ms
Turnaround time of P_4 Process = 14 – 2 = 11ms

$$\text{Average Turnaround Time} = \frac{10+14+17+11}{4} = \frac{42}{4} = 10.5\text{ms}$$

(ii) Waiting time of each process for each of the following algorithm

For FCFS

Waiting time for P_1 Process = 1 – 1 = 0ms
Waiting time for P_2 Process = 6 – 1 = 5ms
Waiting time for P_3 Process = 11 – 2 = 9ms

Waiting time for P_4 Process = 16 – 2 = 14ms

$$\text{Average Waiting Time} = \frac{0+5+9+14}{4} = \frac{28}{4} = 7ms$$

For SJF

Waiting time for P_1 Process = 5 – 1 = 4ms
Waiting time for P_2 Process = 1 – 1 = 0ms
Waiting time for P_3 Process = 14 – 2 = 12ms
Waiting time for P_4 Process = 11 – 2 = 9ms

$$\text{Average Waiting time} = \frac{0+4+12+9}{4} = \frac{25}{4} = 6.25ms$$

Q 71. What is a scheduler ? Explain the primary objectives of scheduling. **(PTU, Dec. 2019)**

Ans. Scheduler : *Refer to Q.No. 42*
Various objectives of Scheduling are as follows :
1. Optimizing the efficiency of labor.
2. Utilizing Equipment to the fullest extent.
3. Increasing profit and output.
4. Serv level Improvement
5. Manufacturing time Reduction
6. Production cost Minimization
7. Worker cost minimization
8. Inventory minimization

❑❑❑

Chapter 3

Inter-Process Communcation

Contents

Critical Section, Race Conditions, Mutual Exclusion, Hardware Solution, Strict Alternation, Peterson's Solution, The Producer/Consumer Problem, Semaphores, Event Counters, Monitors, Message Passing, Classical IPC Problems : Reader's & Writer Problem, Dinning Philosopher Problem etc.

POINTS TO REMEMBER

- Critical section is a segment of code in which the process may be changing common variables, updating a table and so on.
- Different algorithms exist for solving the critical section problem, with the assumption that only storage interlock is available.
- **Race Condition :** When several processes access and manipulate the same data concurrently.
- **Semaphore :** A binary semaphore is a semphore with an integer value that can range over or only between 0 and 1.
- Semaphores can be used to solve various synchronization problem and can be implemented efficiently especially if hardware support for atomic operation is available.
- **Event Counter :** Event counter is a datastructure that can be used for process synchronization.

QUESTION-ANSWERS

Q 1. Define critical section. **(PTU, Dec. 2019, 2016, 2011, 2009 ; May 2008)**

OR

What is critical section problem? How is it handled?

(PTU, Dec. 2016 ; May 2010)

Ans. Consider a system consisting of n processes (P_0, P_1, ... P_{n-1}). Each process has a segment of code, called a critical section, in which the process may be changing common variables, updating a table, writing a file and so on. The important feature of the system is that when one process is executing in its critical section, no other process is to be allowed to execute critical section.

A solution to the critical section problem must satisfy the following three requirements :

1. Mutual Exclusion : If process P_i is executing in its critical section, then no other processes can be executing in their critical sections.

2. Progress : If no process is executing in its critical section and some processes wish to enter

Waiting time for P_4 Process = 16 – 2 = 14ms

$$\text{Average Waiting Time} = \frac{0+5+9+14}{4} = \frac{28}{4} = 7ms$$

For SJF

Waiting time for P_1 Process = 5 – 1 = 4ms
Waiting time for P_2 Process = 1 – 1 = 0ms
Waiting time for P_3 Process = 14 – 2 = 12ms
Waiting time for P_4 Process = 11 – 2 = 9ms

$$\text{Average Waiting time} = \frac{0+4+12+9}{4} = \frac{25}{4} = 6.25ms$$

Q 71. What is a scheduler ? Explain the primary objectives of scheduling. **(PTU, Dec. 2019)**

Ans. Scheduler : *Refer to Q.No. 42*
Various objectives of Scheduling are as follows :
1. Optimizing the efficiency of labor.
2. Utilizing Equipment to the fullest extent.
3. Increasing profit and output.
4. Serv level Improvement
5. Manufacturing time Reduction
6. Production cost Minimization
7. Worker cost minimization
8. Inventory minimization

❑❑❑

Chapter 3

Inter-Process Communcation

Contents

Critical Section, Race Conditions, Mutual Exclusion, Hardware Solution, Strict Alternation, Peterson's Solution, The Producer/Consumer Problem, Semaphores, Event Counters, Monitors, Message Passing, Classical IPC Problems : Reader's & Writer Problem, Dinning Philosopher Problem etc.

POINTS TO REMEMBER

- Critical section is a segment of code in which the process may be changing common variables, updating a table and so on.
- Different algorithms exist for solving the critical section problem, with the assumption that only storage interlock is available.
- **Race Condition :** When several processes access and manipulate the same data concurrently.
- **Semaphore :** A binary semaphore is a semphore with an integer value that can range over or only between 0 and 1.
- Semaphores can be used to solve various synchronization problem and can be implemented efficiently especially if hardware support for atomic operation is available.
- **Event Counter :** Event counter is a datastructure that can be used for process synchronization.

QUESTION-ANSWERS

Q 1. Define critical section. **(PTU, Dec. 2019, 2016, 2011, 2009 ; May 2008)**

OR

What is critical section problem? How is it handled?

(PTU, Dec. 2016 ; May 2010)

Ans. Consider a system consisting of n processes (P_0, P_1, ... P_{n-1}). Each process has a segment of code, called a critical section, in which the process may be changing common variables, updating a table, writing a file and so on. The important feature of the system is that when one process is executing in its critical section, no other process is to be allowed to execute critical section.

A solution to the critical section problem must satisfy the following three requirements :

1. Mutual Exclusion : If process P_i is executing in its critical section, then no other processes can be executing in their critical sections.

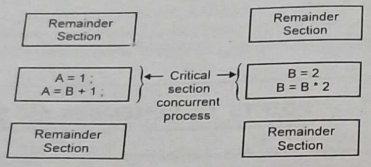
2. Progress : If no process is executing in its critical section and some processes wish to enter

their critical sections, then only those processes that are not executing in their remainder section participate in the decision on it which will enter its critical section next and this selection cannot postponed indefinitely.

3. **Bounded Waiting :** There exists a bound on the number of times that other processes allowed to enter their critical sections after a process has made a request to enter its critical sect and before that request is granted.

Q 2. Explain diagrammatically critical section.

Ans.



Q 3. Explain the structure of critical section.

Ans. Critical section problem is to design a protocol that the processes can use to co-operate. Each process must request permission to enter its critical sections at the same time. The section of code implementing this request is the entry section. The critical section may be followed by an exit section. The remaining code is the remainder section.

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

Q 4. Explain in brief about process synchronization. **(PTU, May 2018)**

Ans. Process synchronization : It is a technique which is used to coordinate the process that use shared data. It refers to the idea that multiple processes are to join up or handshake at a certain point, in order to reach an agreement or commit to a certain sequence of action. It occurs in an operating system among cooperating processes.

Q 5. What is binary semaphore? What is its use?

(PTU, Dec. 2014 ; May 2017, 2006)

Ans. The semaphore construct described in the previous sections is commonly known as counting semaphore, since its integer value can range over an unrestricted domain. A binary semaphore is a semaphore with an integer value that can range over or only between 0 and 1. A binary semaphore can be simpler to implement than a counting semaphore, depending on the underlying hardware architecture.

Let S be a counting semaphore. To implement binary semaphore


```
binary-semaphore S1, S2 ;
    int C ;
initially S1 = 1, S2 = 0
```

```
Wait (S1)
C .... ;
if (C > 0)
{
    Signal (S1) ;
    Wait (S2) ;
}
    Signal (S1) ;
```

The signal operation on the counting semaphore S can be implemented as follows :

```
Wait (S1)
C++ ;
if (C < = 0)
     signal (S2) ;
else
     signal (S1) ;
```

Q 6. What is cooperating process in process synchronization?

Ans. A cooperating process is one that can affect or be affected by other processes executing in the system. Cooperating process can either directly share a logical address space both code and data or be allowed to share data only through files or messages.

Q 7. Explain race condition. **(PTU, Dec. 2014)**

Ans. When both processes manipulate the variable counter concurrently. A situation like this, where several processes access and manipulate the same data concurrently and the outcome of the execution depends on the particular order in which the access takes place, is called a race condition.

Q 8. Explain Peterson's solution to critical section problem. What are the limitations of this solution and how it can be resolved ? **(PTU, May 2016)**

Ans. Peterson's solution is restricted to two processes that alternate execution between their critical section and remainder sections. The processes are numbered P_0 and P_1.

```
        int turn ;
    boolean flag [2] ;
```

The variable turn indicates whose turn is to enter its critical section. That is, if turn== i, then process P_i is allowed to execute its critical section.

The flag array is used to indicate if a process is ready to enter its critical section.

For example, if flag [i] is true, this value indicates that P_i is ready to enter its critical section.

```
do {
 flag [i] = true
 turn = j ;
 while (flag [j] && turn = = j) ;
      critical section
 flag [i] = false ;
      Remainder section
} while (true) ;
```

Limitation to Peterson's solution :

1. Strict order of execution
2. Variable updates (turn and flag)

Could still be problematic.

Q 9. How do processes communicate? **(PTU, Dec. 2011)**

Ans. Inter process communication, which in short is known as IPC, deals mainly with the techniques and mechanisms that facilitate communication between processes. The IPC mechanisms can be classified into the following categories as given below :

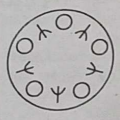
(a) Pipes
(b) FiFOs
(c) Shared memory
(d) Mapped memory
(e) Message queues
(f) Sockets.

Q 10. Explain spinlock.

Ans. The main disadvantage of semaphore is busy waiting. While 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 continual looping is a clear problem in a real multiprogramming system, when a single CPU is shared among many processes. Busy waiting waste CPU cycles called spinlock because the process "spins" while waiting for the lock.

Q 11. Give an example of producer-consumer problem, indicating the reasons for inconsistency that can arise due to race conditions. **(PTU, Dec. 2005)**

Ans. In this problem, five philosophers sit around a circular table eating spaghetti and thinking. In front of each philosopher is a plate and to the left of each philosopher plate. When a philosopher wishes to eat he picks up the forks to the right and to the left of his plate. When done puts both forks back on the table.



There are two issues first the deadlock must never occur and second on set of philosophers should be able to act to prevent philosopher from even eating.

```
While (true)
{
Wait (chopstick [i]) ;
Wait (chopstick [(i+1) %5]) ;
// eat
signal (chopstick [i]) ;
signal (chopstick [(i+1) %5])
// think
}
```

Q 12. What is synchronization? **(PTU, May 2017, 2006)**

Ans. A cooperating process is one that can affect or be affected by other processes executing in the system. Cooperating processes may either directly share a logical address space or be allowed to share data through files. Concurrent access to shared data may result in data inconsistency.

Consider a system consisting of n processes $\{P_0, P_1, \ldots\ldots P_{n-1}\}$. Each process has a segment of code, called a critical section, in which the process may be changing common variables, updating a table, writing a file and so on. The important feature of the system is that, when one process is executing in its critical section, no other process is to be allowed to execute in its critical section. Thus, the execution of critical sections by the processes is mutually exclusive in time. 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.

A solution to critical section problem satisfy the following requirements :

1. Mutual Exclusive : If process P_i is executing in its critical section, then no other processes can be executing in their critical sections.

2. Progress : If no process is executing in its critical section and some processes wish to enter their critical sections.

3. Bounded waiting : There exists a bound on the number of times that other processes are allowed to enter their critical sections after a process has made a request to enter its critical section and before that request is granted.

Synchronization Hardware : As with other aspects of software, hardware features can make the programming task easier and improve system efficiency.

The critical section problem could be solved simply in a uniprocessor environment if we could forbid interrupts to occur while a shared variable is being modified. In this manner, we could be sure that the current sequence of instructions would be allowed to execute in order without preemption. No other instructions would be run, so on unexpected modifications could be made to the shared variable.

Unfortunately, this solution is not feasible in a multiprocessor environment. Disabling interrupt on a multiprocessor can be time-consuming, as the message is passed to all the processors. This message passing delays entry into each critical section and system efficiency decreases. Also, consider the effect on system's clock, if the clock is kept updated by interrupts.

Q 13. How are critical regions and the principle of mutual exclusion related to each other? **(PTU, Dec. 2004)**

Ans. Mutual exclusion is a way of making sure that if one process is using a shared modifiable data, the other processes will be excluded from doing the same thing. And the critical region is the key to prevent trouble involving shared storage is find out some way to prohibit more than process from reading and writing the shared data simultaneously. Hence, critical region and the principle of mutual exclusion are related to each other.

Q 14. Explain producer consumer problem.

Ans. A producer process produces information that is consumed by a consumer process. For example, a print program produces characters that are consumed by the printer driver. A compiler may produce assembly code, which is consumed by an assembler. The assembler, in turn, may produce object modules which are consumed by the loader.

To allow producer and consumer processes to run concurrently, we must have available a buffer of items that can be filled by the produce and emptied by the consumer. A producer can produce one item while the consumer is consuming another item. The producer and consumer must be synchronized so that the consumer does not try to consume an item that has not yet been produced. In this situation, the consumer must wait until an item is produced.

The unbounded-buffer producer-consumer problem places no practical limit on the size of the buffer. The consumer may have to wait for new items, but the producer can always produce new

items. The bounded buffer producer-consumer problem assumes a fixed buffer size. In this case, the consumer must wait if the buffer is empty and the producer must wait if the buffer is full.

Q 15. What are the problems faced when the processes are to be synchronized? Explain. (PTU, Dec. 2019, 2004)

Ans. Following types of problems can be faced during synchronization of processes :

1. Producer and Consumer : This models a scenario where several processes produce items and several processes consume items. The items are stored in a buffer of a limited size. The synchronization problem requires that the buffer neither underflows nor overflows or in other words, that no producer attempts to put an item into a full buffer.

2. Readers and Writers : This models a scenario where several processes write shared data and shared processes read shared data. The synchronization problem requires that no two writers write the data simultaneously and that no reader reads the data while it is being written.

3. Dining Philosophers : This models a scenario where several philosophers alternatively think and dine at a round table. The table contains as many plates and forks as there are philosophers. A philosopher needs to pick two forks adjacent to his plate to dine. The problem approximates a situation where several processes complete for an exclusive use of resources with the possibility of a deadlock.

4. Sleeping Barber : This models a scenario where several customers visit a barber in a barber shop. The shop contains a limited number of seats for waiting customers. The barber serves customers one at a time or sleeps when there are no customers. A customer enters the shop and either wakes the barber to be served immediately, or waits in a seat to be served later or leaves when there are no free seats.

The problem approximates a situation where several processes queue to get served by another process.

Q 16. Write a note on the Interprocess communication. (PTU, Dec. 2019 ; May 2012)

Ans. Interprocess communication : In computing, inter-process communication (IPC) is a set of methods for the exchange of data among multiple threads in one or more processes. Process may be running on one or more computers connected by a network. IPC methods are divided into methods for message passing, synchronization, shared memory and remote procedure. The method of IPC used may vary based on the bandwidth and latency of communication between the threads, and the type of data communicated.

1. Information sharing
2. Computational speedup
3. Modularity
4. Convenience
5. Privilege separation.

Q 17. What are monitors in process synchronization ?

Ans. The monitor is one of the ways to achieve process synchronization. The monitor is supported by programming language to achieve mutual exclusion between processes. For example Java synchronized methods. Java provides wait( ) and notify( ) constructs.

1. It is a collection of condition variables and procedures combined together in a special kind of module or a package.
2. A process running outside the monitor can't access the internal variable of the monitor but can call procedures of the monitor.

3. Only one process at a time can execute code inside monitor.

Syntax

```
Monitor Demo
{
    variable;
    condition variables
    procedure p1 (.....)
    procedure p2 (.....)
}
    syntax of monitor.
```

Q 18. What do you mean by event counters?

Ans. An event counter is another data structure that can be used for process synchronization. Like a semaphore, it has an integer count and a set of waiting process identifications. Unlike semaphores, the count variable only increases. It is similar to the next customer number used in systems where each customer takes a sequentially numbered ticket and waits for that number to be called.

Q 19. What is role of scheduler ? (PTU, Dec. 2018)

Ans. Scheduler in OS are the process which decides which task and process should be accessed and run at what time by the system resources. The scheduler in operating system are the algorithm which help in the system optimisation for maximum performance.

Q 20. What are monitors and semaphores? How do they help in the process synchronization and inter process communication? (PTU, Dec. 2013)

Ans. Monitor : A monitor is a structured concurrent programming construct that handles requests that requires synchronization like the access to a critical section.

A monitor is a natural generalization of an object in object oriented programming, which encapsulates data operations. The monitor ensure that only one process can execute a monitor operation at any time.

Many different implementation of monitors exist, which behave slightly different.

Semaphores : A semaphores is an integer value that may be initialized to non-negative number where wait ( ) operation will increment semaphore value.

Semaphores and monitor ensures by various measurements that only one process can execute and other process have to wait and also insure of reducing of waiting time of different processes. This increase synchronization and interprocess communication (IPC) and thus prevent from the condition of deadlock.

Q 21. Provide the solution for Dinning Philosopher's problem using semaphores. (PTU, Dec. 2019 ; May 2014)

Ans. Philosophers spend their lives thinking and eating

Don't interact with their neighbors, occasionally try to pick up 2 chopsticks (one at a time) to eat from bowl.

Need both to eat, then release both when done.

In the case of 5 philosopheres :

- **shared data**

  Bowl of rice (data set)

  semaphore chopstick [5] initialized to 1

  Philosopher i :

  do {

```
P (chopstick [i])
P (chopstick [(i+1)%5])
....
//eat
....
V (chopstick [i]);
V (chopstick [(i+1)%5]);
....
//think
....
}while(1);
```

Q 22. What is meant by synchronization ? Explain various methods of is done in operating systems. **(PTU, Dec. 2014)**

Ans. Synchronization refers to one of two distinct but related concepts :

1. Synchronization of processes
2. Synchronization of data

Process synchronization refers to the idea that multiple processes are to join up or handshake at a certain point, in order to reach an agreement or commit to a certain sequence of action.

Data synchronization refers to the idea of keeping multiple copies of a dataset in coherence with one another, or to maintain data integrity.

Synchronization methods :

1. Hardware 2. Semaphores 3. Monitors

1. Hardware :

- Based on special instructions like test_and_set, exchange.
- Often in combination with interrupts.
- Sometimes used as basis for OS synchronization mechanisms.

2. Semaphores : It is used to solve synchronization problems among n processes. It is fundamental synchronization tool used in many operating systems.

Integer variable that can be accessed only via two atomic operations.

- Wait
- Signal

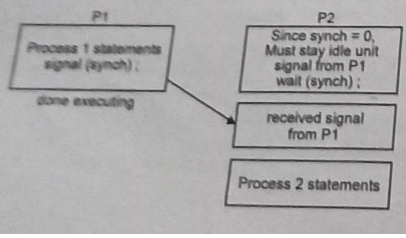
It is frequently used for mutual exclusion ; mutex as special semaphore.

Example :

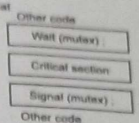
goal : force P2 to execute after P1

Common semaphore synch to synchronize the operations of the two concurrent processes

- wait, signal utilized to delay P2 until P1 is done
- synch initialized to 0.



Semaphore mutex



Busy waiting : If P2 has to wait for P1, P2 loops continuously until P1 is done. Most mutual exclusion solutions result in "busy waiting"

Sol. P2 blocks itself, "wakes up" via wakeup operation

Analysis

- each semaphore has an integer value and a list of associated processes.
- when a process blocks itself on a semaphore, it is added to a queue of waiting processes.
- The signal operation on a semaphore removes a process from the queue and wakes the process up.

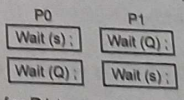
Declaration and operations

```
type semaphore = record
    value : integer ;
    L : list of process
end;
wait(s) : s.value : = s.value -1;
    if s.value < 0
    then begin  - - add this process to s.L ;
        block ;
        end ;
    signal(s) : s.value : = s.value +1;
    if s.value <= 0
    then begin - - remove next process from s.L;
    wakeup(P);
        end;
```

Critical section and semaphores

- Operation on semaphores must be atomic ; we must ensure that no two processes can execute wait and signal on the same semaphore at the same time.
- Uni-processor system; hardware support such that transactions are atomic and utilize interrupt during the time wait and signal are executing
- Multiprocessor system; hardware support is expensive for multiple CPUs and employ software algorithms if hardware instructions are not available.

Deadlock with semaphores : Semaphores must be used with care :



P0 is waiting for P1 to execute signal (Q)

P1 is waiting for P0 to execute signal (s)
Both processes are in a deadlock !

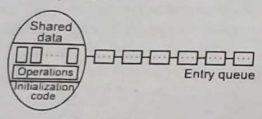
Binary Semaphores : Semaphores in which their integer value can only be either 0 or 1 or used by 2 or more processes to ensure only one can enter critical section.

Monitors :

- It is a high level synchronization construct; programmer defined operators.
- Access to internal data structures only via special procedures
- Provide a software solution to synchronization problems.

Properties :

- Ensure that only one process at a time can be active within the monitor.
- Programming language construct.
  Monitors procedures treated differently by the compiler.
- Compiler and OS are responsible for mutual exclusion, not the programmer.
  Less prone to errors than semaphores.
- Somewhat limited in its simple form.



Monitor diagram

Monitor syntax

```
type monitor_name = monitor
variable declarations
procedure entry P1 (. . .) ;
    begin (. . .)end;
procedure entry P2 (. . .) ;
    begin . . . end;
    :
    procedure entry Pn (. . .) ;
    begin . . . end;
```

A problem with monitors : Need a way for processes to block themselves whenthey cannot proceed. Solution is condition variable

Condition Variable : Variable that can be used with two operations :

1. x.wait suspends the process until it is invoked by another process, and
2. x.signal releases exactly one process from the affiliated waiting queue.

Monitor Drawbacks :

- Only usable in a few programming languages.
- Solves mutual exclusion problem only for CPU that all have access to common memory, not designed for distributed systems.

Q 23. List the techniques used for process synchronization. **(PTU, Dec. 2015)**

Ans. ❑ Semaphore ❑ Monitor

Q 24. Provide the solution for Reader/Writer problem using semaphores. **(PTU, May 2016)**

Ans. No reader will be kept waiting unless a writer has the object.

Writing is performed as soon as possible.
The reader processes share the semaphores mutex and wrt and the integer readcount. The semaphore wrt is also shared with the writer processes.
Mutex and wrt are initialized to 1 and readcount is initialized to 0.

```
Writer                          Reader
                                wait(mutex);
                                readcount : = read
                                count + 1;
                                if readcount = 1 then
Wait (wrt);                     wait(wrt);
.....                           signal(mutex);
Writing is performed            .....
....                            reading is performed
signal(wrt);                    ....
                                wait(mutex);
                                readcount : = readcount – 1;
                                if readcount = 0 then
                                signal(wrt);
                                signal(mutex);
```

Q 25. What is the difference between binary and counting semaphore ? **(PTU, Dec. 2016)**

Ans. Difference between binary semaphore and counting semaphore.

- A binary semaphore is a semaphore that can range over 0 and 1 whereas a counting semaphore can range over an unrestricted domain.
- A binary semaphore can be simpler to implement than a counting semaphore.

❑❑❑

Chapter 4

Deadlocks

Contents

Definition, Necessary and sufficient conditions for Deadlock, Deadlock Prevention, Deadlock Avoidance : Banker's algorithms, Deadlock detection and Recovery.

POINTS TO REMEMBER

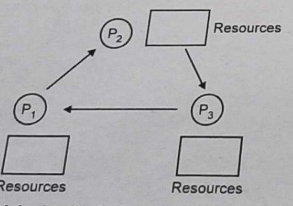
- *Deadlock is a situation where a coating process is never again able to change state.*
- ***Starvation :** Starvation occurs when one or more threads in your program are blocked for gaining access to a resource.*

QUESTION-ANSWERS

Q 1. What is deadlock? Explain with example.

(PTU, Dec. 2014, 2009 ; May 2015, 2006)

***Ans.** A deadlock is a situation in which two processes sharing the same set of resources are effectively preventing each other from accessing the resource, algorithm and resulting in both programs ceasing to function. A process requests resource and if resources are not available at that time, the process enters a waiting state. Sometimes, a waiting process is never again able to change state called deadlock.*



Deadlock state

$P_1 \rightarrow P_2$
$P_2 \rightarrow P_3$
$P_3 \rightarrow P_1$
(Request for resources)

Q 2. Explain deadlock characteristics.

***Ans.** If these four conditions occur simultaneously, then deadlock occurs.*

***1. Mutual Exclusion :** A resource can be used by only one process at a time.*

***2. Hold and Wait :** At least one process is holding a resource, and needs to acquire other resources that are being held by other processes.*

3. No Pre-emption : If a process holds a resource, it cannot be pre-empted.

4. Circular Wait : There must exist a set $\{P_0, P_1, \ldots P_n\}$ of waiting processes such that P_0 is waiting for a resource held by P_0.

Q 3. Explain deadlock prevention and avoidance. **(PTU, May 2009)**

Ans. Deadlock prevention can be done by following ways :

1. Elimination of mutual exclusion condition : The mutual exlusion condition must hold for non-sharable resources. That is several processes cannot simultaneously share a single resource. This condition is difficult to eliminate because some resources such as tape drive and printer are inherently non-sharable. Note that sharable resources like read only file do not require mutually exclusive access and thus cannot be involved in deadlock.

2. Elimination of hold and wait condition : There are two possibilities for elimination of the second condition. The first alternative is that a process request be granted all of the resources it needed once prior to execution. The second alternative is to disallow a process from requesting resources whenever it has previously allocated resources. This statergy requires that all of the resources a process will need must be requested at once.

3. Elimination of non-preemption condition : Forcing a process waiting for a resource that cannot immediately be allocated to relinguish all of its currently held resources so that other processes may use them to finish, can alleviate the non-preemption condition.

4. Elimination of 'circular wait' condition : The last condition 'the circular wait' can be denied by imposing a total ordering on all of the resource types and then forcing all processes to request the resources in order. This strategy imposes a total ordering of all resources types and to require that each process request resources.

In numerical order of enumeration with the rule, the resources allocation graph can never have a cycle.

Avoidance : For avoiding deadlock is to require additional information about how resources are to be requested. The various logarithms differ in the amount and type of information required. The simplest and most useful model required that each process declare the maximum number of resources of each type that it may need.

Given a prior information about the maximum number of resources of each type that may be requested for each process it is possible to construct an algorithm that ensures that the system will never enter a deadlock state. This algorithm defines the deadlock avoidance approach. A deadlock avoidance algorithm dynamically examines the resource allocation state to ensure that a circular wait condition can never exist.

Q 4. Explain dealing with deadlocks.

Ans. Once we have know that a deadlock exists in the system, we should now understand how deal with deadlock.

These methods are :

- **Deadlock prevention :** Do not let the deadlock occur.
- **Deadlock recovery :** Let the deadlock occur in the system and then attempt to recover the system from deadlock.
- Do not do anything for deadlock prevention and recovery. Let the deadlocks occur in the system and deteriorate the system performance, leading to a situation where system will have to be restarted manually.

Q 5. Write the difference between deadlock prevention and avoidance methods. **(PTU, May 20**

Ans.

Dead Lock Prevention	Deadlock Avoidance
1. Preventing deadlocks by constraining how requests for the resources can be node in the system and how they are handled (system design).	1. The system dynamically considers every request and decides whether it is safe to grant it at this point.
2. The goal is to ensure that at least one of the necessary conditions for deadlock can move hold.	2. The system requires additional appropriate information regarding the overall potential use of each resource for each process.
	3. Allows more concurrency. Similar to the difference between a traffic light and a police officer directing traffic.

Q 6. List the Coffman's conditions that lead to deadlock. **(PTU, Dec. 20**

Ans. 1. Mutual exclusion
2. Hold and wait
3. No preemption
4. Circular wait.

Q 7. List out some reasons for process termination. **(PTU, Dec. 2006 ; May 20**

Ans. A process terminates when it finishes executing its last statement. Its resources are returned to the system, it is purged from any system lists or tables and its process. Control block PCB is erased, i.e., the PCBs memory space is returned to a free memory pool.

1. Normal exit
2. Error
3. Killed by another process.

Q 8. Explain recovery from deadlock.

Ans. Once the deadlock has been detected in the system should now be recovered from the state. This is must because it will continuously degrade the system's performance and if no correction mechanism is applied, there is a possibility of total system failure. The ways how to recover deadlock

1. Process termination
2. Resource termination.

Q 9. When the system is in safe state? **(PTU, Dec. 20**

Ans. A state is safe if the system can allocate resources to each process in some order and still avoid a deadlock. More formally a system is in safe state only if there exists a safe sequence. A sequence of process (P_1, P_2 ..... P_n) is a safe sequence for the current allocation state if for each P_i the resource that P_i can still request can be satisfied by the currently available resources plus the resources held by all the P_j with $J < 1$. In this situation, if the resources that process P_i needs are not immediately available, then P_i can want until all P_j have finished. When they have finished P_i can obtain all of its needed resources, complete its designated task, return its allocated resources and terminate.

Q 10. Explain various strategies to deal with deadlock. How deadlock is detected and recovered? **(PTU, Dec. 2018, 2014, 2008)**

Ans. Various strategies to deal with deadlock are :

1. We can use protocol to prevent or avoid deadlock, ensuring that the system will never enter a deadlock state.
2. We can allow the system to enter a deadlock state, detect it and recover.
3. We can ignore the problem altogether and pretend that deadlocks never occur in the system. This solution is used by most operating system including unix.

If a system does not employ either a deadlock : Prevention or a deadlock avoidance algorithm, then a deadlock situation may occur. In this environment the system must provide.

- An algorithm that examines the state of the system to determine whether a deadlock has occurred.
- An algorithm never recovered from deadlock.

When a detection algorithm determine a deadlock exists, several alternatives exist. One possibility is to inform the operator that a deadlock has occurred and to let the operator that a deadlock has occurred and to let the operator deal with the deadlock manually. There are two options for breaking the deadlock automatically. One solution is simply to abort one or more processes to break the circular wait. The second option is to preempt, some resources from one or more of the deadlock processes.

Q 11. Explain the following table of resource requirements for four processes and determine the current allocation is safe state, with regard to avoiding deadlock.

Process	Max. need	Current usage
P1	7	3
P2	4	1
P3	6	2
P4	6	1

Total resources = 10, Available resources = 3 **(PTU, Dec. 2003)**

Ans. Process P_2 can immediately be allocated all its resources and then return them, then process P_1 can be allocated all its resources and after use P_1 will return all its 7 resources and then P_3 will allocated its resources and return them after P_3, the process P_4 will be allocated all resources and after use it return them. Hence, the current location is safe state.

Q 12. Which is the main limitation of resource allocation graph? **(PTU, May 2008)**

Ans. If resource allocation graph contains no cycle, then no process in the system is deadlocked. If the graph does contain a cycle, then a deadlock may exist. If each resource type has exactly one instance, then a cycle implies that a deadlock has occurred. If the cycle involves only a set of resource types, each of which has only a single instance, then a deadlock is occurred. Each process involved in the cycle is deadlocked.

Q 13. How can you prevent circular waiting situation in a deadlock? **(PTU, May 2008)**

OR

Briefly describe a deadlock prevention approach that ensures that the circular wait condition is never fulfilled? **(PTU, Dec. 2011)**

Ans. One condition for deadlock is the circular wait condition. One way to ensure that this

condition never holds is to impose a total ordering of all resource types, and to require that each process requests resources in an increasing order of enumeration.

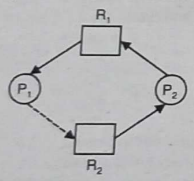
Let R = {R_1, R_2 ..... R_m} be set of resource types. We assign to each resource type a unique integer number, which allows us to compare two resources and to determine whether one precedes another in our ordering. Formally, we define a one-to-one function. F : R → N, where N is the set of natural numbers, e.g., if the set of resource types R includes tape drives, disk drives and printers, then the function F might be defined as follows :

F (tape drive) = 1
F (disk drive) = 5
F (printer) = 12

Now consider the following protocol to prevent deadlocks. Each process can request resources only in an increasing order of enumeration. That is, a process can initially request any number of instances of a resource type, say R_i. After that, the process can request instances of a resource type R_j if and only if F (R_j) > F (R_i). If several instances of the same resource type are needed, a single request for all of them must be issued.

Q 14. Explain how a deadlock can be represented graphically for two processes and two resources. Discuss the merits/demerits. **(PTU, Dec. 2005)**

Ans. Suppose that P_2 requests R_2. Although R_2 is currently free, we cannot allocate it to P_2 since this action will create a cycle in the graph. A cycle indicates that the system is an unsafe state. If P_1 requests R_2 and P_2 requests R_1, then deadlock will occur.



Methods of Recovery :

1. Process termination

Merits :

- It aborts all deadlock processes.
- It aborts one process at a time.

Until the deadlock cycle is eliminated.

Demerits :

- Aborting a process may not be easy. If the process was in midst of updating a file, terminating it will leave that file in a incorrect state.

2. Resource Preemption :

Merits :

- Selecting [illegible] and which processes are to be preempted? Cost factors, may include such parameters as the number of resources a deadlock process is holding and the amount of time a deadlock process has thus for consumed during its execution.

Demerits :

- In a system where victim selection is based primarily on cost factors ; it may happen that some process is always picked as a victim.

Q 15. What is the main advantage of using deadlock detection instead of prevention or avoidance? **(PTU, May 2010 ; Dec. 2003)**

Ans. If a system does not employ either a deadlock prevention or a deadlock avoidance algorithm, then a deadlock situation may occur. In this system must provide :

1. An algorithm that examines the state of the system to determine whether a deadlock has occurred.
2. An algorithm to recover from the deadlock.

Deadlocks occur only when some process makes a request that cannot be granted immediately. This request may be the final request that completes a chain of waiting processes. In the extreme, we could invoke the deadlock detection algorithm every time a request for allocation cannot be granted immediately. In this case, we can identify not only the set of processes that is deadlocked, but also the specific process that caused the deadlock. If there are many different resource types, one request may cause many cycles in the resource graph, each cycle completed by the most recent request and caused by one identifiable process.

Q 16. Explain the banker's algorithm. **(PTU, May 2011)**

Ans. The resource allocation graph algorithm is not application to a resource allocation system with multiple instances of each resource type. The deadlock avoidance algorithm that we describe next is applicable to such a system, but is less efficient than the resource – allocation graph scheme. This algorithm is commonly known as the banker's algorithm. The name was chosen because this algorithm could be used in a banking system to ensure that the bank never allocates its available cash such that it can no longer satisfy the needs of all its customers.

When a new process enters the system, it must declare the maximum no. of instances of each resource type that it may need. This number may not exceed the total number of resources in the system. When a user requests a set of resources will leave the system in a safe state. If it will, the resources are allocated, otherwise, the process must wait until some other process releases enough resources.

Several data structures must be maintained to implement the banker's algorithm. These data structure encode the state of resource allocation system. Let n be the number of processes in the system and m be the number of resource types. We need the following data structures:

Available : A vector of length m indicates the number of available resources of each type. If available [j] = K, there are K instances of resource type R_j available.

Max : An n×m matrix defines the maximum demand of each process. If max [i, j] = K, then process P_i may request at most K instances of resource type R_j.

Allocation : An n×m matrix defines the number of resources of each type currently allocated to each process. If allocation (i, j) K, then process P_i is currently allocated K instances of resource type R_j.

Need : An n×m matrix indicates the remaining resource need of each process. If need [i, j] = K, then process P_i may need K more instances of resource type R_j to complete its task. Note that need (i, j) = Max (i, j) – Allocation (i, j)

These data structures vary overtime in both size and value.

To simplify the presentation of the banker's algorithm, let us establish some notation. Let X

and Y be vectors of length n. We say that $X \leq Y$ if and only if $X[i] \leq Y[i]$ for all $i = 1, 2, \ldots n$. For example, if X = (1, 7, 3, 2) and Y = (0, 3, 2, 1), then $Y \leq X$. $Y < X$ if $Y \leq X$ and $Y \neq X$.

We can say that each row in the matrices allocation and need as vectors and refer to them as allocation and need, respectively. The vector allocation i specifies the resources current allocated to process P_i, the vector need i specifies the additional resources that process P_i may still request to complete the task.

Q 17. How would you kill a process? **(PTU, Dec. 2006)**

Ans. We can kill a process by using abort ( ) system call. A parent may want to abort its child's execution due to one of the following reasons.

1. Child has exceeded allocated resources.
2. Task assigned to a child is no longer required.
3. Parent is terminating and operating system does not allow a child to continue if its parent terminates.

Q 18. What for are Resource Allocating Graphs used? **(PTU, Dec. 2008 ; May 2008)**

Ans. It is used for deadlock detection. This graph consists of set of vertices V and a set of edges E. The set of vertices V is a partitioned into two different types of nodes $P = \{P_1, P_2, \ldots P_n\}$ the set consisting of all the active processes in the system and $R = \{R_1, R_2, \ldots R_{n-1}\}$ the set consisting of all resources types in the system.

A directed edge from process P_i to resource type R_J is denoted by $P_i \rightarrow R_J$; it signifies that the 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 is denoted by $R_J \rightarrow P_i$; it signifies that an instance of resource type R_J has been allocated to process P_i. A directed edge $P_i \rightarrow R_J$ is called a request edge ; a directed edge $R_J \rightarrow P_i$ is called an assignment edge.

Q 19. List out Coffman's conditions that lead to deadlock. **(PTU, Dec. 2006)**

OR

List and explain the conditions necessary and sufficient to produce a deadlock.

(PTU, May 2010, 2009, 2004 ; Dec. 2004, 2003)

Ans. 1. Necessary Conditions : A deadlock situation can arise if following conditions hold simultaneously :

(i) Mutual exclusion : At least one resource must be held in a non-sharable mode ; that is only one process at a time can use the resource.

(ii) Hold and wait : A process must be holding at least one resource and waiting to acquire additional resources that are currently being held by other processes.

(iii) No preemption : Resources cannot be preempted, that is, a resource can be released only voluntarily by the process holding it.

(iv) Circular wait : A set $\{P_0, P_1, \ldots P_n\}$ of waiting processes must exist such that P_0 is waiting for a resource held by P_1, P_1 is waiting for a resource held by P_2, ..., P_{n-1} is waiting for resource held by P_n and P_n is waiting for resource held by P_0.

Q 20. What is deadlock? Explain different conditions required to occur a deadlock. How deadlock prevention is possible? **(PTU, May 2018 ; Dec. 2015, 2012)**

OR

What is deadlock? List and explain four necessary conditions for dead lock to occur. Explain different algorithms for prevention and avoidance of deadlocks.

(PTU, Dec. 2010, 2009)

Ans. In a multiprogramming environment, several processes may compete for a finite number of resources. A process state request 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.

A process must request a resource before using it, and must release the resource after using it. A process may request as many resources as it requires to carry out its designed task. Obviously, the number of resource requested may not exceed the total number of resources available in the system. In other words, a process cannot request three printers if the system has only two.

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

1. Request : If the request cannot be granted immediately then the requesting process must wait until it can acquire the resource.

2. Use : The process can operate on the resource.

3. Release : The process release the resource.

A set of processes is in a deadlock state when they process in the set is waiting for an event that can be caused only by another process in the set. To illustrate a deadlock state, we consider a system with three tape drives. Suppose each of three processes holds one of these tape drives. If each process now requests another tape drive, the three processes will be in a dead lock state. Each is waiting for the event "tape drive is released" which can be caused only by one of other waiting processes.

Dead locks may involve different resource types e.g. consider a system with one printer and one tape drive. Suppose that process P_i is holding the tape drive and P_j is holding the printer. If P_i requests the printer and P_j requests the tape drive ; a deadlocks occurs.

Deadlock characteristics : Following are the features that characteristics deadlock :

1. Necessary conditions :

(i) Mutual exclusion : At least one resource must be held in a non-sharable mode ; that is only one process at a time can use the resource.

(ii) Hold and wait : A process must be holding at least one resource and waiting to acquire additional resources that are currently being held by other processes.

(iii) No preemption : Resources cannot be preempted, that is resource can be released only voluntarily by the process holding it.

(iv) Circular wait : A set $\{P_0, P_1, \ldots P_n\}$ of waiting processes must exist such that P_0 is waiting for a resource held by P_1, P_1 is waiting for a resource held by P_2, ... P_{n-1} is waiting for a resource held by in so on.

Deadlock prevention can be done by following ways :

1. Elimination of mutual exclusion condition : The mutual exclusion condition must hold for non-sharable resources. That is, several processes cannot simultaneously share a single resource. This condition is difficult to eliminate because some resources such as tape drive and printer are inherently non-sharable. Note that sharable resources like read only file donot require mutually exclusion access.

2. Elimination of hold and wait condition : There are two possibilities for elimination of the second condition.

The first alternative is that a process request be granted all of the resources it needed once prior to execution. The second alternative is to disallow a process from requesting resources whenever it has previously allocated resources.

3. Elimination of non-preemption condition : Forcing a process waiting for a resource that cannot immediately be allocated to relinquish all of its currently held resources.

4. Elimination of circular wait condition : The last condition the circular wait can be denied by imposing a total ordering on all of the resources types and then forcing all process to request the resource in order.

Avoidance : For avoiding deadlock is to require additional information about how resources are to be requested. The various logarithm differ in the amount and type of information required. The simplest and most useful model required that each process declare the maximum number of resources of each type that it may need.

Given a prior information about the maximum number of resources of each type that may requested for each process it is possible to construct an algorithm that ensures that the system will never enter a dead lock state. This algorithm defines the deadlock avoidance approach. A deadlock avoidance algorithm dynamically examines the resource allocation state to ensure that a circular wait condition can never exist.

Q 21. Consider a system consisting of m resources of the same type, being shared by n processes. Resources 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 :

(a) The maximum need of each process is between l and m resources.

(b) The sum of all maximum needs is less than m + n. **(PTU, May 2016)**

Ans. Suppose N = sum of all Need i,
A = sum of all allocation ;
M = sum of all Max i

Use contradiction to prove.

Assume this system is not deadlock free.

If there exists a deadlock state, then A = m because there's only one kind of resources and resources can be requested and released only one at a time. From condition b, N + A = M < m + n. So we get N + m < m + n so we get N < n. It shows that at least one process, that Need i = 0. From condition a, Pi can release at least 1 resource so there are n – 1 processes sharing m resources now, condition a and b still hold. Go on the argument, no process will wait permanently, so there's no deadlock.

Q 22. Define Starvation in deadlock. **(PTU, Dec. 2018)**

Ans. A system is fair when each threads gets enough access to limited resources to make reasonable progress. A fair system prevents starvation and deadlock. Starvation occurs when one or more threads in your program are blocked for gaining access to a resource and, as a result, cannot make progress. Deadlock, the ultimate form of starvation, occurs when two or more threads are waiting on a condition that cannot be satisfied.

❑❑❑

Chapter 5

Memory Management

Contents

Basic Concept, Logical and Physical address map, Memory allocation : Contiguous Memory allocation - Fixed and variable partition - Internal and External fragmentation and Compaction; Paging : Principle of operation - Page allocation - Hardware support for paging, Protection and sharing, Disadvantages of paging.

Virtual Memory : Basics of Virtual Memory - Hardware and control structures - Locality of reference, Page fault, Working Set, Dirty page/Dirty bit - Demand paging - Page Replacement algorithms : Optimal, First in First Out (FIFO), Second Change (SC), Not recently used (NRU) and Least Recently used (LRU).

POINTS TO REMEMBER

- Memory management means to place the programs in memory and fully utilize the memory and high degree of multiprogramming is there and share the memory between several processes.
- Every memory, address generated by CPU must be checked for legally and possibly mapped to physical address.
- A multiprogrammed system will generally perform more efficiently if it has higher level of multiprogramming.
- A simple base register or a pool of base and limit register is sufficient for the single and multiple partition schemes, whereas paging and segmentation need mapping tables to define the address map.
- LRU replacement is an approximation of optimal but even it may be difficult to implement.
- Virtual memory can be thought of as one level of hierarchy of storage level in computer system. Each level has its own access time, size and cost parameters.
- If process does not have enough memory for its working set it will thrash. Providing enough frames to each process to avoid thrashing may require process swapping and scheduling.
- An addition to page replacement algorithm, a frame allocation policy is needed. Allocation can be fixed suggesting local page replacement or dynamic suggesting global replacement.
- LRU replacement is an approximation of optimal but even it may be difficult to implement.
- Optimal page replacement requires future knowledge.
- FIFO page replacement is easy to program but suffers from Belady's anomaly.
- One solution to external fragmentation problem is compaction. Compaction involves shifting a program in memory without the program noticing the change.

- System with fixed sized allocation units is the single partition scheme and paging suffers from internal fragmentation.
- As the memory management algorithm becomes more complex, the time required to map a logical address to a physical address increases.
- If paging or segmentation is provided different sectors of a user program can be declared execute only, read only or read write.
- A multi-programmed system will generally perform more efficiently if it has higher level of programming.
- Pure demand paging never brings in a page unit that page is referenced.
- The operating system consults an internal table to determine where the page in form the backing store.

QUESTION-ANSWERS

Q 1. Explain memory management.

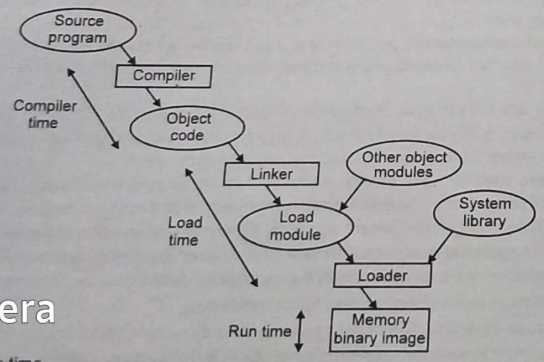
Ans. Memory management means to place the programs in memory and fully utilized the memory and high degree of multiprogramming is there and share the memory between several processes. Memory consists of large array of words or bytes, each with its own address. It increases the performance and several processes can be kept in memory.

Q 2. Explain instruction-execution cycle where memory interacts.

Ans. 1. First fetches an instruction from memory.
2. The instruction is then decoded and operands to be fetched from memory.
3. After the executed instruction on the operands and then operand is stored on to memory.

Q 3. Explain address binding in memory.

Ans. Program is a set of instructions and each instruction of a program is saved in RAM, when it is executed and thus each instruction has its location/address in the memory.



1. Compile time
2. Load time
3. Run time

Q 4. What is spooling? **(PTU, May 2010, 2008 ; Dec. 2008)**

Ans. The introduction of disk technology operating system to keep all jobs on a disk rather than in serial card reader. Rather than the cards being read from the card reader directly into memory and then the job being processed, cards are read from the card reader on the disk.

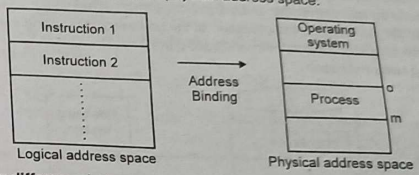
Q 5. Explain the operating system responsibility for memory management.

Ans. 1. Keeping track of which parts of memory are currently being used and whom.
2. Deciding which processes and data to move into and out of a memory.
3. Allocating and deallocating memory space as needed.

Q 6. Explain logical and physical address space. **(PTU, May 2017, 2015, 2012)**

Ans. A logical address is the address of an instruction of or data as used by a program.

A physical address is the effective memory address of an instruction or data, i.e., it is the address obtained after binding of logical addresses. Set of logical address is called logical address space and set of physical address is called physical address space.



Q 7. What is difference between cache memory and auxiliary memory? **(PTU, Dec. 2007)**

Ans. Cache memory : With increasing clock speeds for modern CPUs the disparts between the CPU speed and the access speed of RAM has grown substantially. In order to get information from RAM, CPU has to wait for many clock cycles, a considerable waste of time then cache is used to increase CPU speed and fetch the record. It is expensive.

Auxiliary memory : A high speed memory bank used to maintain frames and super- computers. It is not directly addressable by the CPU rather it functions like a disk.

Q 8. What is thrashing? Why is TLB used in paging scheme ?

(PTU, Dec. 2016 ; May 2015, 2014, 2011, 2008)

Ans. Suppose any process does not have enough frames, it is possible to reduce the number of allocated frames to the minimum and there is some number of pages are in active use. If the process does not have this number of frames, it will quickly page fault. At this point, it must replace some page. Since all its pages are in active use, it must replace a page that will be needed again. It quickly faults again and again. This process continues to fault, replacing pages for which it then faults and brings back in right away. This high paging activity is called thrashing.

A fast associative memory called translation look-aside buffer (TLB) is used to hold most recently used page table entries. When the CPU needs to access a particular page, the TLB is accessed first. If desired page table entry is present in the TLB, it is called TLB hit and then the frame number is retrieved from the table to get the physical address in main memory. If the desired page table entry is not present in the TLB, it is called TLB miss and then the CPU searches the original page table in main memory for the desired and this information (entry) is also loaded in the TLB. This ensures that translation information pertaining to a future reference is confined to the TLB.

Q 9. Explain best fit and worst fit algorithm of allocation. **(PTU, May 2009)**

Ans. Best fit algorithm : Best fit algorithm chooses the block that is closest in size to the request. It gives poor performance when it has to search complete list.

Worst fit algorithm : It chooses the block, i.e., largest in size and it is not useful because of memory wastage.

Q 10. Judgement field in the page table. **(PTU, May 2012)**

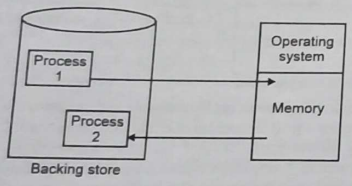
Ans. The judgement field decides if a page has to be moved back to the secondary memory or not.

The page map table has two additional columns : status and judgement. Initially implying that all the pages are in secondary. As when a page is loaded from secondary to primary the status is updated available otherwise judgement is there.

Q 11. Describe the concept of swapping. **(PTU, Dec. 2012, 2010, 2006, 2004)**

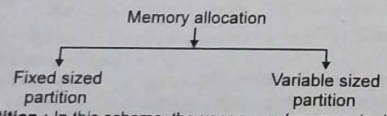
Ans. Swapping is simple memory/process management technique used by operating system. Swapping is one of the efficient regular and authentic approach of memory management. It is used by the opreating system to increase the utilization of the processor by moving some blocked process from the main memory to secondary memory (hard disks).

Swapping two processes :

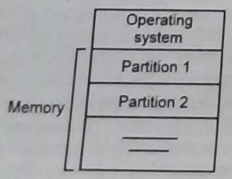


Q 12. Explain memory allocation.

Ans.

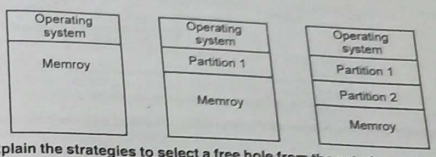


Fixed sized partition : In this scheme, the user area of memory is divided into a number of fixed sized partitions. This partition may be equal or unequal size, but the size of each partition is fixed.



Variable sized partition : Because of the fixed size partition, 50% of memory may remain unused due to reason. To overcome this problem, the scheme variable number of memory partitions was introduced. In this scheme, the number, size and location of the partitions vary dynamically as processes come and go.

e.g.



Q 13. Explain the strategies to select a free hole from the set of available holes.

Ans. 1. First fit : Allocate the first hole that is big enough. In this case searching of block can start either at the beginning or where the first-fit search ended. We stop searching as soon as we find a free hole that is largest enough.

2. Best fit : In this case, allocate the smallest hole that is big enough for process. This strategy produces the smallest leftover hole.

3. Worst fit : In this case, allocate the largest hole. Again, we must search the entire list unless it is sorted by size.

Q 14. What is garbage collection? **(PTU, May 2008)**

Ans. Garbage collection is reclaiming memory from objects no longer in use and returning it to system. Garbage collection is also used in general graph directory. We use garbage collection scheme to determine when the last referenced has been detected and the disk space can be reallocated. Garbage collection involves traversing the entire file system, making everything that can be accessed. Garbage collection for a disk-based file system, however, is extremely time-consuming and is thus seldom attempted.

Q 15. Explain fragmentation.

Ans. Memory fragmentation can be internal or external as well. The memory allocated to a process may be slightly larger than the requested memory. The difference between these two numbers is internal fragmentation. External fragmentation exists when there is enough total memory space to satisfy a request, but the available spaces are not contiguous; storage is fragmented into a large number of small holes.

Q 16. What is logical address space? **(PTU, May 2011, 2008 ; Dec. 2004)**

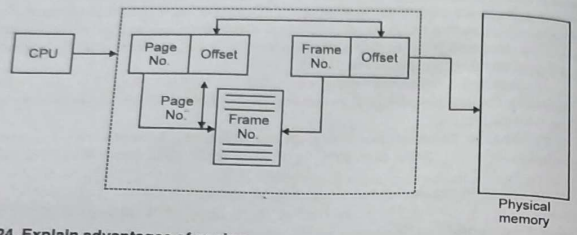
Ans. An address generated by CPU is referred to as logical address. Logical addresses are also known as virtual address. The set of all logical addresses generated by a program is logical address space. The set of all physical addresses corresponding to these logical address is physical address space. Thus in execution time address binding scheme M the logical and physical address spaces differ.

Q 17. Explain non-contiguous storage allocation.

Ans. To resolve the problem of external fragmentation or to enhance multiprogramming allocating non-contiguous physical address space of a process so that a process could be allocated memory wherever it was available.

Q 23. Explain a diagrammatic approach to mapping a logical address to a physical address.

Ans.



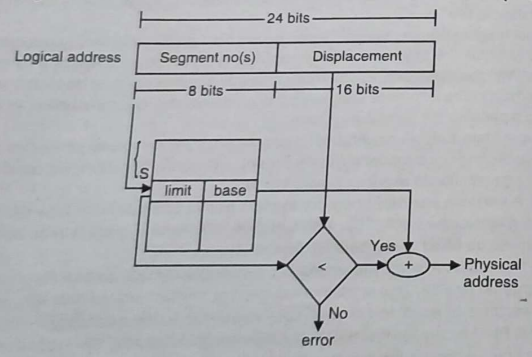
Q 24. Explain advantages of paging.

Ans. 1. There is never a possibility of external fragmentation, it really achieves a higher degree of multiprogramming as compared to contiguous storage allocation strategies.

2. Sharing a common code among the processes is now possible. Single copy of some commonly used program such as compilers, editors, database systems can be kept.

Q 25. Describe using a diagram how a logical address consisting of 24 bits could be converted to a segment address, supporting upto 256 segments. What would be the maximum size of each segment? (PTU, Dec. 2003)

Ans.

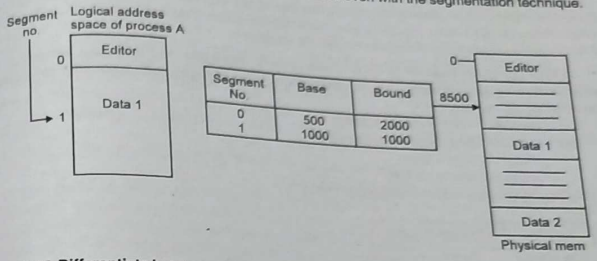


Maximum size of each segment depends on the available physical memory.

Q 26. Explain advantages of segmentation.

Protection bits are associated with segments, which check that attempt to write a read only segment should fail. They can also be used to check that attempt to use an execute-only segment as a data segment should be prohibited.

2. As with paging, sharing of code or data is possible even with the segmentation technique.



Q 27. Differentiate between a page and a frame. (PTU, Dec. 2009)

Ans. Physical memory is broken into fixed sized blocks called frames. Logical memory is also broken into blocks of the same size called pages. When a process is to be executed its pages are loaded into any available memory frames that from the backing store. The backing store is divided into fixed sized blocks that are of same size as the memory frames.

Q 28. Why pages are of sizes always powers of 2?

(PTU, Dec. 2012, 2010, 2008, 2005 ; May 2017, 2016, 2007)

Ans. The pages size is defined by the hardware. The size of the page is typically a power of 2, varying between 512 bytes and 16 MB a page, depending on the computer architecture. The selection of power of 2 as a page size makes the translation of a logical address into a page number and page offset particularly easy. If the size of logical address space is 2^m, and a page size is 2^n addressing units.

Q 29. List various free space management techniques and explain them.

(PTU, May 2010)

Ans. Disk space is limited, we need to reuse the space from deleted files for new files, if possible. To keep track of free disk space, the system maintains a free-space list. Following are the techniques for managing free-space.

1. Bit Sector : The free space list is implemented as a bit map. Each block is represented by 1bit. If the block is free, the bit is 1, if the block is allocated, the bit is 0. The main advantage of this is simplicity and efficiency in finding the first free block or consecutive free blocks on the disk.

2. Linked List : Another approach is to link together all the free disk blocks, keeping a pointer to the first free block in a special location on the disk and catching it in memory. The first block contains a pointer to the next free disk block and so on. However, this scheme is not efficient to traverse the list, we must read each block, which requires substantial I/O time.

3. Grouping : In this store the addresses of n free blocks in the first free block. The first n-1 of these blocks are actually free. The last block contains the addresses of another n free block and so on.

4. Counting : In this keep the address of first free block and the number n of free contiguous blocks that follow the first block. Each entry in the free-space list then consist of a disk address and a count. Although each entry requires more space than would a simple disk address, the overall list will be shorter as long as the count is generally greater than 1.

Q 30. A computer uses an 18-bit address system, with 6 bits used as a page address and 12 bits used as a displacement. Calculate the total number of pages and express the following address as a paging address.

001111000000111000 (PTU, Dec. 2003)

Ans. Computer uses 18 bit address system

Page address = 6 bit

Displacement is 12 bits

In a paging address = 001111000000111000

001111	00000111000
Page number	Displacement

And page size is always power of 2

Here page size is $2^n = 2^{12} = 4096$ bytes.

Q 31. Explain the following terms :

(a) Page fault

(b) Demand paging

(c) Resident page set

(d) Working set **(PTU, Dec. 2003)**

Ans. (a) Page Fault : If a process tries to access a page that was not brought into memory or access to a page marked invalid causes a page-fault trap. The paging hardware, in translating address through the page table, will notice that the invalid bit is set, causing a trap to operating system. The procedure for handling this page fault is :

1. Check internal table for this process, to determine whether the reference was a valid or invalid memory access.
2. If the reference was invalid, terminate the process. If it was valid, but not gets brought in that page.
3. Find a free frame.
4. Schedule a disk operation to read the desired page into the newly allocated frame.
5. When the disk read is complete, modify the internal table kept with the process and the page table to indicate that the page is now in memory.
6. Then start the instruction that was interrupted by integral address trap. The process can now access the page as though it had always been in memory.

(b) Demand Paging : A demand paging is similar to a paging system with swapping. Processes reside on secondary memory. When we want to execute a process, we swap it into memory. Rather than swapping the entire process into memory, we use a lazy swapper. A lazy swapper never swaps a page into memory unless that page will be needed. Since we are now viewing a process as a sequence of pages, rather than as one large contiguous address space, use of swap is technically correct.

(c) Resident page set : Operating system have memory areas that are "pinned down", i.e.,

cannot be swapped out to secondary storage, e.g., interrupt, the page tables are usually not pageable. Data buffers that are accessed outside of the CPU, timing dependent kernel/application areas cannot tolerate the varying response time caused by paging.

(d) Working Set : The working set is based on the assumption of locality. This model uses a parameter Δ, to define the working set window. The idea is to examine the most recent Δ page references. The set of pages in the most recent Δ page references is the working set. If a page is in active use, it will be in the working set. If it is no longer being used, it will drop from the working set Δ time units after it has reference. Thus, the working set is an approximation of the program's locality.

Q 32. Give or write short note on virtual memory. **(PTU, Dec. 2009, 2005, 2003)**

OR

What is virtual memory? What are its various advantages?

(PTU, Dec. 2017, 2016, 2015 ; May 2011, 2009, 2006)

Ans. Virtual memory : It is a technique that allows the execution of processes that may not be completely in memory. Virtual memory refers to the concept whereby a process with large size than available memory can be loaded and executed by loading to process in parts. The main advantage of this concept is that program can be larger than physical memory.

Advantages :

1. The size of user's program would be larger be considered by the available size of physical memory. Users would be able to write program for very large virtual address space, simplifying the programming task.
2. Since each user utilizes less physical memory, more users can keep their programs in memory simultaneously which will cause increase in CPU utilization and throughput.

Q 33. What is page fault? **(PTU, May 2014)**

Ans. In computer storage technology, a page is a fixed length block of memory that is used as unit of transfer between physical memory and external storage like a disk and a page fault is an interrupt (or exception) to the software raised by the hardware, when a program accesses a page that is mapped in address space but not loaded in physical memory.

The hardware that detects this situation is the memory management unit in a processor. The exception handling software that handles the page fault is generally part of an operating system. The operating system tries to handle the page fault by making required page accessible at a location in a physical memory or kills the program in case it is an illegal access. Containing to what their name might suggest page faults are not errors and are common and necessary to increase the amount of memory available to programs in any operating system that utilizes virtual memory including Microsoft Windows, Mac OS X, Linux and Unix. Note that microsoft uses the term hard fault in its Resource Monitor to mean 'page faults' (if Resource view help in Microsoft Operating Systems).

Q 34. What is page table?

Ans. A page table is the data structure used by a virtual memory in a computer operating system to store the mapping between virtual addresses and physical addresses. Virtual addresses are those unique to the accessing process. Physical addresses are those unique to the CPU, i.e., RAM.

Role of page table : In operating systems – that virtual memory, every process have been paged out to a backup storage. When a process required or requested access to its memory, it is the responsibility of the operating system to map the virtual address where that memory is stored.

Q 35. What is demand paging? **(PTU, Dec. 2014)**

Ans. In computer operating systems, demand paging is an application of virtual memory. In a system that uses demand paging, the operating system copies a disk page into physical memory only if an attempt is made to access it (i.e., if a page faults occur). It follows that a process begins execution with none of its pages in physical memory and many page faults will occur until. Most of processes working set of pages is located in physical memory. This is an example of lazy loading techniques.

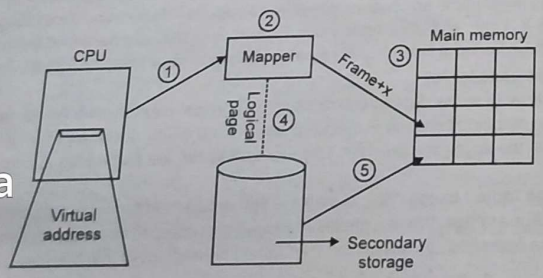
Demand paging follows that pages should only be brought into memory if the executing process demands them. This is often referred to as lazy evaluation as only those pages demand by the process are swapped from secondary storage to main memory contrast this to pure swapping, where all memory for a process is swapped from secondary storage to main memory during process start up.

When a process is to be swapped into main memory for processing, the pages guesses which pages will be used prior to the process being swapped out again. The pages will only load these pages into memory. This process avoids loading pages that are unlikely to be used and focuses on pages needed during the current process execution period. Therefore not only is unnecessary page load during swapping avoided but we also try to per cent which pages will be needed and avoid loading pages during execution. Commonly, to achieve this process a page table implementation is used. The page table maps logical memory to physical memory. The page table uses a bitwise operator to make if a page is valid or invalid. A valid page, is one that current resides in main memory. An invalid page is one that currently resides in secondary memory. When a process tries to access a page, the following steps are generally followed :

1. Atttempt to access page.
2. If page is valid the continue processing instruction as normal.
3. If page is invalid then a page fault trap occurs.
4. Check if the memory reference is a valid reference to a location on secondary memory. If not, the process is terminated (illegal memory access). Otherwise, we have to page in the required page.
5. Schedule disk operation to read the desired into page memory.
6. Restart the instruction that was interrupted by operating system trap.

Q 36. Diagrammatic approach of virtual memory.

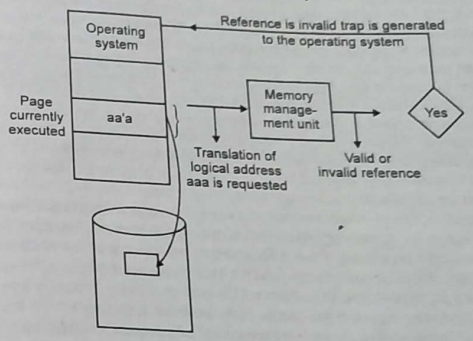
Ans.



Q 37. Explain the performance of demand paging.

Ans. In virtual memory system, demand paging is a type of swapping in which pages of programs are not copied from disk to main memory until they are needed for execution.

In demand paging, virtual address (logical address) is accessed by CPU, the corresponding page number is looped up in the page table and if it shows that currently this page is not in main memory, then this page must be brought into the main memory.



A valid-invalid bit in the page table may be used to find out whether page in memory or in disk. Page fault is also there if page is not found in memory. Page fault must be followed by swapping in the page from disk to main memory or trap should be generated to the operating system.

Q 38. What is page fault? Why does it occur? **(PTU, Dec. 2007)**

OR

What do you mean by page-faults? When do page-faults occur? Describe the action taken by the O.S when page fault occurs.

(PTU, Dec. 2019, 2010, 2009 ; May 2008)

Ans. Page Fault : If a process tries to access a page that was not brought into memory or access to a page marked invalid causes a page-fault trap. The paging hardware, in translating address through the page table, will notice that the invalid bit is set, causing a trap to operating system. The procedure for handling this page fault is :

1. Check internal table for this process, to determine whether the reference was a valid or invalid memory access.
2. If the reference was invalid, terminate the process. If it was valid, but not get brought in that page.
3. Find a free frame.
4. Schedule a disk operation to read the desired page into the newly allocated frame.
5. When the disk read is complete, modify the internal table kept with the process and the page table to indicate that the page is now in memory.

Q 39. What is the translation looked aside buffer (TLB)?

Ans. A translation look aside buffer (TLB) is a CPU cache that memory management hardware uses to improve virtual address translation speed. It has the first cache introduced in processors. All current desktop and server processors use a TLB.

The TLB is typically implemented as content-addressable memory (CAM). The CAM search key is the virtual address and the search result is a physical address. If the requested address is present in TLB, the CAM search yields a match quickly and the retrieved physical address can be used to access memory. This is called a TLB bit. If the requested address is not in the TLB, it is a miss and the translation proceeds by looking up the page table in a process called a page walk. The page walk is an expensive process, as it involves reading the contents of multiple memory locations and using them to compute the physical address. After the physical address is determined by the page walk, the virtual address to physical address mapping are entered into the TLB.

A TLB has a fixed number of slots that contain page table entries, which map virtual addresses to physical addresses. The virtual memory is the space seen from a process. The space is segmented in pages of a prefixed size. The page table (generally loaded in memory) holds trace of where the virtual pages are loaded in the physical memory. The TLB is a cache of page table, that is only a subset of its content are stored.

The TLB references physical memory address in its table. It may reside between the CPU cache and primary storage memory or between levels of a multi-level cache. The placement determines weather the cache uses physical or virtual addressing. If the cache is virtually addressed, requests are sent directly from the CPU to the cache, and the TLB is accessed only on and cache miss. If the cache is physically addressed, the CPU does a TLB look up on every memory operation and the resulting physical address is sent to the cache. There are pros and cons to both implementations. Caches using virtual addressing do not require access to TLB but must be fused every context switch in a multiprocessing environment. In a Havard architecture or hybrid thereof, a separate virtual address space or memory access hardware may exist for instructions and data. This can lead to distinct TLBs for each access types.

A common optimization for physically addressed caches is to perform the TLB loop up in parallel with the cache access. The low order bits of any virtual address (e.g., in a virtual memory) system having 3 KB pages, the lower 12 bits of virtual address) represent the offset of the desired address within the page and thus they do not change in the virtual to physical translation. During a cache access, two steps are performed : an index is used to find an entry in the cache's data store and then the tags for the cache access, two steps are performed: an index is used to find an entry in the cache's data store, and then the tags for the cache line found are compared. If the cache is structured in such a way that can be indexed using only the bits that do not change in translation, the cache can perform its "index" operation while the TLB translates the upper bits of the address. Then, the translated address from the TLB is passed to cache. The cache performs a tag comparison to determine if this access was a hit or miss. It is possible to perform the TLB look up in parallel with the cache access even if the cache must be indexed using some bits that may change upon address translation. See the address translation section in the cache article for more details about virtual addressing as it pertains to caches and TLBs.

Q 40. Explain pure demand paging.

Ans. *Pure demand paging is the form of demand paging in which not even a single page is loaded into memory, initially. Thus, very first instruction causes a page fault. This type of demand paging may decrease the performance of a computer system by generally executed or increasing the effective access time of memory.* **ma** *is a memory access time*

P is the probability of occurrence of page fault.

[Effective access time = $(1 - P) \times ma + P \times$ page fault time].

Q 41. What is overlay? Give one example.

Ans. An overlay network is a computer network which is built on top of another network. Nodes in the overlay can be thought of as being connected by virtual or logical links, each of which corresponds to a path, perhaps through many physical links in the underlying network. For example, distributed systems such as could computing, peer to peer network and client server.

Q 42. What is the page map table? **(PTU, Dec. 2004)**

Ans. Page map table : Every address generated by the CPU is divided into two parts: (a) page number (b) and a page of offset (d).

The page number is used as an index into a page table. The page table contains the base address of each page in physical memory. Thus, base address is combined with the page offset to define the physical memory address, that is, sent to the memory unit.

Q 43. Explain different page replacement algorithms in virtual memory. List all page replacement policies you know. **(PTU, May 2016, 2007)**

Ans. Following are the various page replacement algorithms in virtual memory :

1. Optimal page replacement algorithm : In this, at the occurs, some set of pages is in memory. One of these pages will be referenced on the very next instruction. Each page can be labelled with the number of instructions that will be executed before that page is first referenced. The optimal page algorithm simply says that the page with the highest label should be removed. If one page will not be used for 8 million instructions and another page will not be used for 6 million instructions removing the former pushes the page fault that will fetch it back as far into the future as possible.

2. Not recently used page replacement algorithm : In order to allow the operating system to collect useful statistics about which pages are being used and which ones are not, most computers with virtual memory have two status bits associated with each other. R is set whenever the page is referenced. M is set when the page is written to. The bits are contained in each page table entry. It is important to realize that these bits must be updated on every memory reference. The not recently used algorithm removes a page at random from the lowest numbered non empty class. Implicit in this algorithm is that it is better to remove a modified page that has not been referenced in at least one clock tick than a clean page that is in heavy use.

3. First-in, First-out page replacement algorithm : The operating system maintains a list of all pages currently in memory, with the page at the head of the list the oldest one and the page at the tail the most recent arrival. On a page fault, the page at the head is removed and the new page added to the tail of the list.

4. Least recently used page replacement algorithm : A good approximation to the optimal algorithm is based on the observation that pages that have been heavily used in the last few instructions will probably be heavily used again in the next few. Conversely, pages that have not been used for ages will probably remain unused for a long time. In this algorithm, when a page fault occurs, throw out the page that has been unused for the longest time. This strategy is called LRU paging.

5. Working set page replacement algorithm : Many paging systems try to keep track of each process working set and make sure that it is in memory before letting, the process run. This approach

is called working set model. It is designed to greatly reduce the page fault rate. Loading the pages before letting processes run is also called prepaging. Note that the working set changes overtime.

Page replacement policies :

1. Optimal page replacement policy
2. First-in-First-out (FIFO) page replacement policy.
3. Least Recently Used (LRU) page replacement policy.

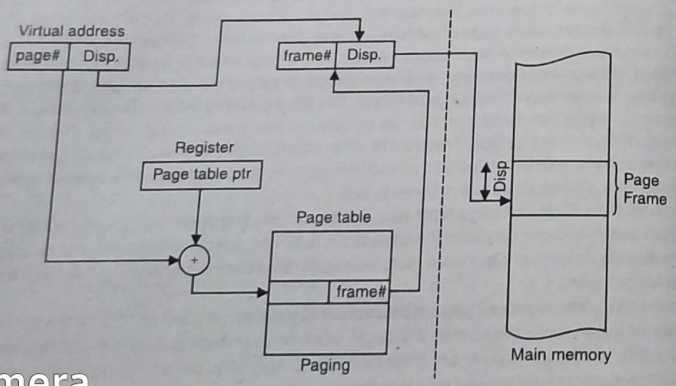
Q 44. What is a paged memory management technique? How logical and physical pages are used in address calculation? Explain. **(PTU, May 2017 ; Dec. 2004)**

Ans. Paged memory management technique : Paging is a memory management scheme that permits the physical-address space of a process to be non-contiguous. Paging avoids the considerable problem of fitting the varying sized memory chunks onto the backing store, from which most of memory management schemes suffer. In this physical memory is broken into fixed sized blocks called frames. Logical memory is also broken into blocks of the same size called pages. When a process is to be executed, its pages are loaded into any available memory frames from the backing store. The backing store is divided into fixed sized blocks that are of the same size as the memory frames.

Use of logical pages in address calculation : In this each blocks are located by an index, with the first block being LBA = 0, and the second LBA = 1 and so on. This scheme replaces earlier schemes which exposed the physical details of the storage device to the software of the OS.

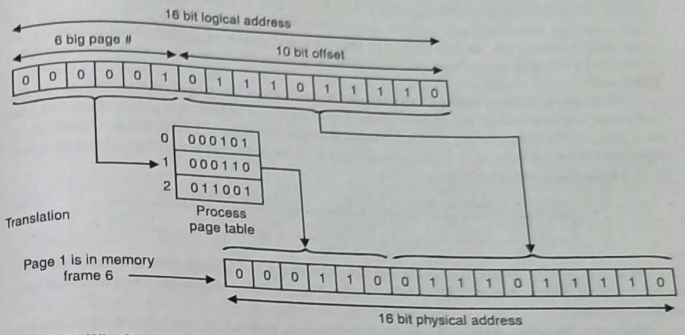
Conversion from logical address to physical address : Logical address in paging :

- *Page size always chosen as a power of 2.*
- *Logical address (p, d) is translated to physical address (f, d) by indexing the page table by p and appending the page displacement d to the frame number f.*



Address translation in Paging

eg :

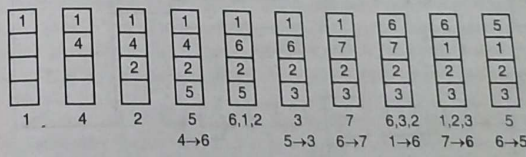


Q 45. Why it is necessary to reallocate a program in memory? **(PTU, May 2004**

Ans. In system with virtual memory programs in memory must be able to reside in different parts of the memory at different times. This is because when the program is swapped back into memory after being swapped out for a while it cannot always be placed in the same location. The virtual memory management unit must also deal with concurrency. Memory management in the operating system should, therefore, be able to reach or reallocate programs in memory and handle memory reference and addresses in the code of the program so that they always point to the right location in memory.

Q 46. Consider the following page reference string :
1, 4, 2, 1, 5, 6, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 5
Find the number of page faults for the reference string for LRU replacement summing 4 frames. All frames are initially empty. **(PTU, Dec. 2007)**

Ans. Page reference string is :
1, 4, 2, 1, 5, 6, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 5



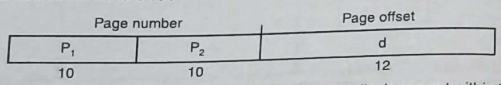
Total number of page faults = 6.

Q 47. Explain different types of page table.

Ans. Most modern computer systems support a large logical address space (2^{32} to 2^{64})

such an environment, the page table itself becomes excessively large. For example, consider a system with a 32-bit logical-address space. If the page size in such a system is 4KB (2^{12}), then a page table may consist of upto 1 million entries ($2^{32}/2^{12}$). Assuming that each entry consists of 4 bytes each process may need to 4MB of physical-address space for page table alone. Clearly, we would not want to allocate the page table into smaller pieces. There are several ways to accomplish this division.

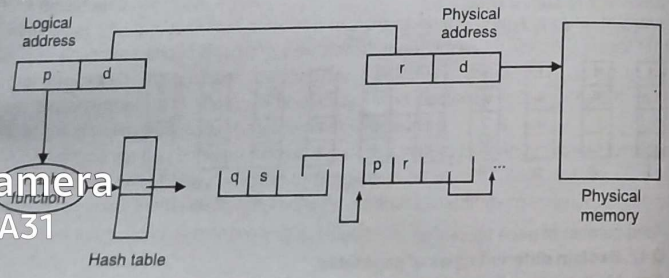
One way is to use a two-level paging algorithm. In which the page table itself is also paged. Remember one example to one 32 bit machine with a page size of 4KB. A logical address is divided into a page number consisting of 20 bits, and a page offset consisting of 12 bits. Because we page the page table, the page number is further divided into a 10-bit page number and a 10-bit page offset. Thus, a logical address is as follows :



Where P_1 is an index into the outer page table and P_2 is the displacement within the page of the outer page table. Because address translation works from the outer page table. The pentium-II uses this architecture.

Hashed page table : A common approach for handling address spaces larger than 32 bits is to use a hashed page table. With the hash value being the virtual page number. Each entry in the hash table contains a linked list of elements that hash to the same location (to handle collisions). Each element consists of three fields : (a) The virtual page number (b) the value of the mapped page frame, and (c) a pointer to the next element in the linked list.

Inverted page table : Usually, each process has a page table associated with it. The page table has one entry for each page that the process is using (or one slot for each virtual address, regardless the latter's ralidity). This table representation is a natural one, since processes reference pages through the page's virtual addresses. The operating system is able to calculate wherein the table associated physical-address entry is, and to use that value directly. One of the drawbacks of this method is that each page table may consist of million of entries. These tables may consume large amounts of physical memory, which is required just to keep track of how the other physical memory is being used.



Hash Page Table

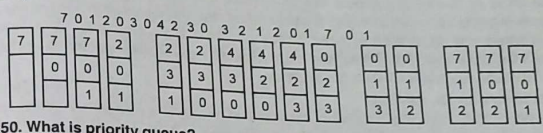
Q 48. Explain page replacement.

Ans. Once the main memory fills up, a page must be swapped out to make room for any pages to be swapped in. This is known as page replacement.

Q 49. Explain FIFO page replacement algorithm.

Ans. The simplest page replacement algorithm is a first in, first out algorithm. A **fifo** replacement algorithm associates with each page the time when that page was brought into memory. When a page must be replaced, the oldest page is chosen. When a page is brought into memory, we insert it at the tail of the queue.

Example :



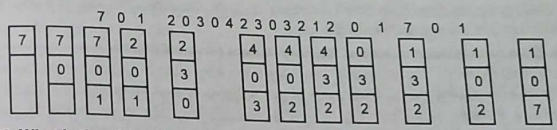
Q 50. What is priority queue? **(PTU, Dec. 2004)**

Ans. A priority is associated with each process and the CPU is allocated to the process with the highest priority. Priority queue is used in priority scheduling algorithm. Equal priority processes are scheduled in FCFS order.

Q 51. Explain LRU page replacement algorithm.

Ans. Optimal algorithm is not feasible, so we need or we can replace the page that has not been used for the longest period of time called least-recently used algorithm.

Example



Q 52. What is thrashing? How does the system detect thrashing? What can the system do to eliminate the problem? **(PTU, May 2015 ; Dec. 2005)**

OR

What is thrashing? How is it eliminated? **(PTU, Dec. 2015, 2007)**

Ans. Thrashing : If a process does not have enough frames, it will very quickly have page fault. At this point it must replace some page. Since all of its pages are in use. It must replace page, which will be needed again. Consequently it very quickly faults again and again and so on. This very high paging activity is called thrashing. A process of thrashing if it spends more time for paging than for executing.

Detection of Thrashing : The process schedules find the decreasing CPU utilization and increases the degree of multiprogramming. The new process tries to take page from running processes, causing more page faults and longer queue for the paging device. The CPU utilization drops further, so that the scheduler tries to increase the degree of multi- programming even more and thrashes occurs and system 'break down'.

Elimination of thrashing : The thrashing effect can be limited by a local replacement scheme. Within the local replacement, if a process starts thrashing, it cannot take frames from another process and so cause it to thrash also. The thrashing effect can be prevented if a process is allocated as many frames as it needs. One possible technique is the working set strategy which is looking at what a process is actually using and then deciding how many frames should be allocated. The main goal of this approach is greatly to reduce the page fault rate.

Q 53. Explain Belady's Anomaly. **(PTU, Dec. 2017 ; May 2014, 2006)**

Ans. For some page replacement algorithms the page fault rate may increase as the number of allocated frames increases. We expect that giving more memory to a process would improve its performance. In some early research, investigators noticed that this assumption was not always true. Belady's anomaly was discovered as a result. Neither optimal replacement nor LRU replacement suffers from Belady's anomaly. There is a class of page replacement algorithms, called stack algorithms, that can never exhibit Belady's anomaly.

Q 54. What is cycle stealing? **(PTU, May 2006)**

Ans. When DMA controller seizes the memory bus, the CPU is momentarily prevented from accessing main memory, through it can still access data items in its primary and secondary cache. Although this cycle stealing can slow down the CPU computation, off loading the data transfer work to a DMA controller generally improves the total system performance.

Q 55. Explain causes of thrashing. **(PTU, May 2015)**

Ans. Thrashing results in severe performance problems. The operating system monitors CPU utilization. If CPU utilization is low, we increase the degree of multiprogramming by introducing a new process to the system. A global page-replacement algorithm is used ; it replaces pages without regard to the process to which they belong. If process enters a new phase and needs more frames. It starts faulting and taking frames away from other processes. So to increase the CPU utilization and decreases the degree of multiprogramming thrashing is used.

Q 56. Give comparison of virtual memory versus physical memory.

Ans. Virtual memory is a technique that allows the execution of processes that may not be completely in memory. One major advantage of this scheme is that programs can be larger than physical memory. Further, virtual memory abstracts main memory into an extremely large, uniform array of storage, separating logical memory as viewed by the user from physical memory. This technique frees programmes from the concerns of memory storage limitations. Virtual memory also allows processes to easily share files and address spaces and it provides an efficient mechanism for process creation.

Q 57. Define compaction. **(PTU, Dec. 2019, 2008)**

Ans. In multiprogramming where many processes can reside in memory at the same time, swapping creates multiple holes in memory, it is possible to combine them all into. One big hole by moving all the processes downward as far as possible. This technique is known as memory compaction. It is usually not done because it requires a lot of CPU time.

Q 58. Explain optimal page replacement algorithm.

Ans. The best possible page replacement algorithm is easy to describe but impossible to implement. The optimal page algorithm simply says that the page with highest label should be removed. If one page will not be used for 8 million instructions and another page will not be used for 6 million instructions, the former page should be removed, as it will not be used in the near future.

Q 59. What is demand and pre-paging? **(PTU, Dec. 2006 ; May 2006)**

Ans. A demand paging system is similar to paging system with swapping. Process resides on secondary memory. When we want to execute a process, we swap it into memory rather than swapping the entire process into memory, however, we use a lazy swapper. A lazy swapper never swaps a page into memory unless that page will be needed.

A swapper manipulates entire process whereas a page is concerned with the individual page of a process.

Q 60. Explain different paging techniques. **(PTU, Dec. 2009)**

Ans. Paging : Paging is a memory management scheme that permits the physical address space of a process to be non-contiguous. Paging avoids the considerable problem of fitting the varying sized memory chunks on to the backing store, from which most of memory management schemes suffer.

In this physical memory is broken into fixed sized blocks called frames. Logical memory is also broken into blocks of the same size called pages. When a process is to be executed, its pages are loaded into any available memory frames from the backing store.

Q 61. What is buffering?

Ans. 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. One reason is to cope with a speed is match between the producer and consumer of a data stream. Another use of buffering is to adopt between devices that have different data transfer sizes. **(PTU, May 2008)**

Q 62. Explain virtual memory and associative memory. **(PTU, May 2008)**

Ans. Virtual memory is a technique that allows the execution of processes that may not be completely in memory. Virtual memory refers to the concept whereby a process with a large size than available memory can be loaded and executed by loading to process in parts. The main advantage of this concept is that program can be larger than physical memory.

As well as it abstracts main memory into main memory into an extremely large uniform array of storage, separating logical memory from physical memory. Virtual memory is the separation of user logical memory from physical memory. This separation allows an extremely large virtual memory to be provided for programmers when only a smaller physical memory is available. It makes the task of programming much easier.

Associative memory is a special type of computer memory used in certain very high speed searching applications. It is also known as content addressable memory, associative storage or associative array, although the last term more often used for programming data structure. Several custom computers were built to implement associative memory.

Unlike standard computer memory, in which user supplies a memory address and the RAM returns the data word stored at that address, a CAM designed such that the user supplies a data word and the CAM searches its entire memory to see if that data word is stored anywhere in it. If the data word is found, the CAM returns a list of one or more storage address where the word was found. Thus, a CAM is the hardware embodiment of what in software terms would be called an associative array.

Q 63. Compare various memory management techniques. **(PTU, Dec. 2008)**

OR

What are the various memory management techniques? Discuss with example. **(PTU, Dec. 2019 ; May 2010)**

Ans. 1. Swapping : A process need to be in memory to be executed. A process however can

be swapped temporarily out of memory to a backing store and then brought back into memory for continued execution. For example assume a multiprogramming environment with a round robin CPU scheduling algorithm. When a quantum expires the memory manager will start to swap out the process that just finished and to swap in another process to the memory space that has been freed.

2. Contiguous memory allocation : We usually want several user processes to reside in memory at the same time. We therefore need to consider how to allocate available memory to the proceses that are in the input queue waiting to be brought into memory. In this contiguous memory allocation each process is contained in a single contiguous section of memory. One of the simplest methods for memory allocation is to devide memory into several fixed sized partitions. Each partition may contain exactly one process. Thus the degree of multi programming is bound by the number of partitions.

3. Paging : Paging is a memory management scheme that permits the physical address space of a process to be non contiguous. Paging avoids the considerable problem of fitting the varying sized memory chunks onto the backing store, from which most of the previous memory management schemes suffered. Physical memory is broken into fixed sized block called frames. Logical memory is also broken into blocks of the same size called pages. When a process is to be executed its pages are loaded into any available memory frames from the backing store. The backing store is divided into fixed sixed block that are of the same size as the memory frames.

4. Segmentation : Segmentation is memory management scheme that supports this uses view of memory. A logical address space is a collection of segments. Each segment has a name and a length. The address specify both the segment name and the offset within the segment. The uses therefore specifies each address by two quantites a segment name and an offset. For simplicity of implementation segment are numbered and are referred to by a segment number, rather than by a segment name.

Q 64. Explain Segmentation with help of an example.

(PTU, Dec. 2008 ; May 2006)

OR

What is the need of segmentation? **(PTU, Dec. 2010)**

OR

Explain the concept of segmentation taking suitable examples.

(PTU, Dec. 2009)

Ans. *Segmentation is a memory-management scheme that supports user view of memory. A logical address space is a collection of segments. Each segment has a name and a length. The addresses specify both the segment name and the offset within the segment. The user therefore specifies each address by two quantities, a segment name and an offset. For simplicity, segments are numbered and are referred to by a segment number rather than by segment name. Thus, a logical address consists of a two tuple :*

<segment – number, offset>

Normally, the user program is compiled and the compiler automatically constructs segments reflecting the input program.

Although the user can refer to objects in the program by a two-dimensional address, the actual physical memory is still, of course, a one-dimensional sequence of bytes. Thus, we must define an implementation to map two-dimensional user-defined addresses into one-dimensional physical addresses. This mapping is affected by a segment table.

Each entry of the segment table has a segment base and a segment limit. The segment base contains the starting physical address where the segment resides in memory, where as the segment limit specifies the length of the segment.

A logical address consists of two parts a segment number, s and an offset into the segment, d. The segment number is used as an index into the segment table. The offset d of the logical address must be between 0 and the segment limit. If it is not, we trap to the operating system. If this offset is legal, it is added to the segment base to produce the address in physical memory of the desired type. The segment table is thus essentially an array of base-limit register pairs.

As an example, we have five segment numbered 0 through 4. The segments are stored in physical memory. The segment table has a separate entry for each segment, giving the begining address of the segment in physical memory and the length of that segment. A particular advantage of segmentation is the association of protection with the segments.

Q 65. Fragmentation of memory. **(PTU, May 2012)**

OR

What is meant by fragmentation ? How is it achieved in LINUX operating system?

(PTU, Dec. 2014)

Ans. In computer storage, fragmentation is phenomenon in which storage space is used inefficiently, reducing capacity and often performance. Fragmentation loads to storage space being "wasted" and the term also refers to the wasted space itself.

Types of fragmentation :

- External fragmentation
- Internal fragmentation

External fragmentation : External fragmentation arises when free memory is separated into small blocks and is interspersed by allocated memory.

Internal fragmentation : More computing memory is allocated than is needed.

When the file system becomes full, it can become fragmented. This is similar to the DOS concept of fragmentation but is not nearly as pronounced and is typically rare on modern GNU/Linux filesystems ; by design GNU/LINUX filesystems are resistant to fragmentation. Keep your filesystems from running near full capacity, and you may never need to worry about fragmentation at all. If there is no space on a filesystem, you cannot write to it at all.

To check on fragmentation, you can unmount the file system and run fsck on it. As part of fsck execution, fragmentation is computer and displayed. You can defragment a file system by backing it up; using mkfs to make a clean, empty image, and then restoring the filesystem. The utility that you use to do your backup and restore is irrelevant and completely up to you. You can use dump/re-store, cpio, or a third party backup program.

Q 66. What do you mean by memory reference string? **(PTU, Dec. 2012)**

Ans. In page replacement algorithm goal is to want lowest page fault rate and evaluate algorithm by running it on a particular string of memory references (reference string) and computing the number of page faults on that string.

Algorithm :

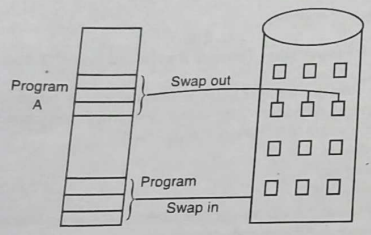
1. FIFO
2. Optimal page replacement
3. Least recently used (LRU) etc.

Q 67. What is the difference between seek time and rotational latency time?
(PTU, Dec. 2017, 2012)

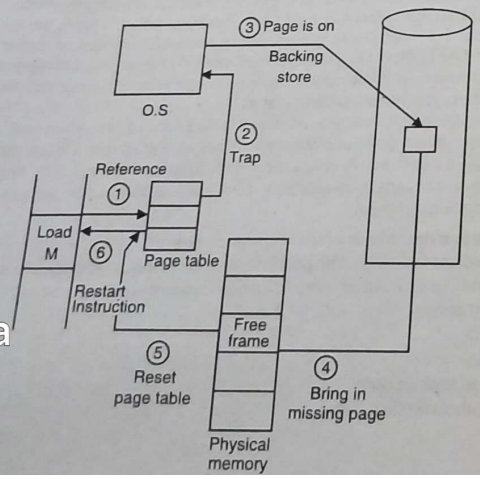
Ans. Seek time : It is a time a program or a device takes to locate a particular piece of data.
Rotational latency time : It is a time delay between the moment something is initiated.

Q 68. Explain demand paging with proper diagram and example.
(PTU, May 2012)

Ans. A demand-paging system is similar to a paging system with swapping. Process reside on secondary memory. When we want to execute a process, we swap it into memory. Rather than swapping the entire process into memory. However, we use a lazy swapper. A lazy swapper never swaps a page into memory unless that page will be needed. Since we now viewing a process as a sequence of pages, rather than as one large contiguous address space, use of swap is technically incorrect.



The procedure



Q 69. List the advantages and disadvantages of contiguous memory allocation.
(PTU, Dec. 2012)

Ans. Contiguous memory allocation is efficient way of allocating memory to the processes.
Advantages :
1. It supports fast sequential and direct access.
2. It provides a good performance.
3. The number of seek required is minimal.
4. Simplicity.
5. No special hardwork.
6. Easily move a process during execution. Operating system can allow process to run efficiently.
7. Provided as much contiguous space as required in memory.

Disadvantages :
1. CPU wasted.
2. Main memory not fully used.
3. Limited job size.
4. Fragmentation problem is here.
5. A partition of memory is permanently allocated to the operating system.
6. Resources are not managed in an efficient manner.

Q 70. What is thrashing? Describe the cause of thrashing. How thrashing can be prevented?
(PTU, Dec. 2012)

Ans. Thrashing : If a process does not have enough frames, it will very quickly have page fault. At this point it must replace some page. Since all of its pages are in use. It must replace page, which will be needed again. Consequently it very quickly faults again and again so on. This very high paging activity is called thrashing. A process of thrashing if it spends more time for paging than for executing.

Detection of Thrashing : The process schedules find the decreasing CPU utilization and increases the degree of multi programming. The new process tries to take page from running processes, causing more page faults and longer queue for the paging device. The CPU utilization drops further, so that the scheduler tries to increase the degree of multi-programming even more and thrashes occurs and system 'break down'.

Elimination of thrashing : The thrashing effect can be limited by a local replacement scheme. Within the local replacement, if a process starts thrashing, it cannot take frames from another process and so cause it to thrash also.

The thrashing effect can be prevented memory, the free memory space is broken into little pieces. External fragmentation exists when enough total memory space exist to satisfy a request, but it is not contiguous, storage is fragmented into a larger number of small holes.

1. Ensure that the files memory segments, CPU and other resources can be operated on by only those processes that have gained proper authentication from the operating system.
2. If a computer has multiple users and allows the concurrent execution of multiple processes.
3. Mechanism for controlling the access of program, processor, or users to resources defined by a computer system.

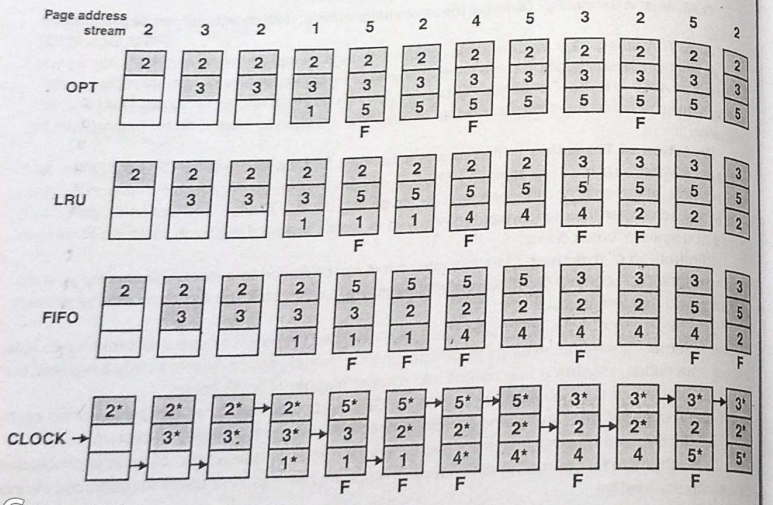
Q 71. On a system with paging, a process cannot access memory that it does not own. Why? How could the operating system allow access to other memory? Why should it or should it not?
(PTU, May 2011)

Ans. An address on paging system is a logical page number and an offset. The physical page

is found by searching a table based on the logical page number to produce a physical page number. Because the operating system controls the contents of this table, it can limit a process to accessing only these physical pages allocated to the process. There is no way for a process to refer to a page it does not own because the page will not be in page table. To allow such access, an operating system simply needs to allow entries for non-process memory to be added to the process's page table. This is useful when two or more processes need to exchange data, they just read and write to the same physical addresses (which may be at varying logical addresses). This makes for very efficient interprocess communication.

Q 72. Explain the LRU page replacement algorithms. Why is it difficult to implement it in pure form? **(PTU, Dec. 2011)**

Ans. The least recently used (LRU) policy replaces the page in memory that has not been referenced for the longest time. By the principle of locality, this should be the page least likely to be referenced in the near future. And, infact, the LRU policy does nearly as well as the optimal policy. The problem with this approach is the difficulty in implementation one approach would be to tag each page with the time of its last refernce, this would have to be done at each memory reference, both instruction and data. Even if the hardware would support such a scheme, the overhead would be tremenders. Alternatively, one could maintain a stack of page-references, again an expensive prospect.



LRU page replacement algorithm

Q 73. What are the main issues in managing the main memory? **(PTU, Dec. 2011)**

Ans. *Following are the main issues in managing the main memory :*

1. Allocation
2. Swapping, Fragmentation and Compaction
3. Garbage Collection
4. Protection
5. Virtual memory
6. I/O Support

Q 74. What do you mean by virtual memory? Why is it needed? Discuss the hardware support required by the operating system to implement the virtual memory concept. **(PTU, Dec. 2011)**

Ans. Virtual memory is a technique that allows the execution of processes that may not be completely in memory. Virtual memory refers to the concept whereby a process with a large size than available memory can be loaded and executed by loading to process in parts. The main advantage of this concept is that program can be larger than physical memory.

As well as it abstracts main memory into main memory into an extremely large uniform array of storage, separating logical memory from physical memory. Virtual memory is the separation of user logical memory from physical memory. This separation allows an extremely large virtual memory to be provided for programmers when only a smaller physical memory is available. It makes the task of programming much easier.

Advantages :

1. The size of user's program would be longer be constrained by the available size of physical memory. Users would be able to write program for very large virtual address space, simplifying the programming task.
2. Since each uses utilizes less physical memory, more users can keep their programs in memory simultaneously which will cause increase in CPU utilization and throughput.
3. Since a process may be loaded into a space of arbitrary size, which is turn reduces external fragmentation without the need to change the scheduled order of process execution. Moreover the amount of space in use by a given process may be changing during its memory residence. As a result the operating system may speed up the execution of important processes by allocating more physical memory.

Hardware requirement for virtual memory : Any process running under 32 bit windows versions gets a set of virtual memory addresses going from 0 to 4, 294, 967, 295 ($2^{*32} - 1 = 4GB$), no matter how much RAM is actually installed on the computer. Actually, this is essentially the same for all operating systems running on 32-bit hardware implement virtual memory.

In the normal, default 32 bit windows OS configuration, 2GB of this virtual address space are allocated to the process private use and the other 2GB are allocated to shared and operating system use.

Only that portion of the address space that is actually referenced by a process or the system is associated with a page frame in real memory or in the page file.

Q 75. What is locality of references and explain its use? What is working set? What is it used for? Also discuss the working set modal in detail. **(PTU, Dec. 2011)**

Ans. Locality of References : Locality of references also called the principle of locality, is the term applied to situations where the same value or related storage locations are frequently accessed. There are three basic types of locality of reference, temporal, spatial and sequential.

Temporal Locality : Here a resource that is referenced at one point in time is referenced again soon afterwards.

Spatial Locality : Here the likelihood of referencing a storage location is greater if a storage location near it has been recently referenced.

Sequential Locality : Here storage is accessed sequentially in descending or ascending order.

Working Set : The group of physical memory pages is currently dedicated to a specific process is known as the working set for that process. The number of pages in the working set can grow and shrink, depending on the overall availability of pages on a system wide basis.

Working set storage management policy is used to maintain the working sets of active program in main memory. The decision to add a new process to the active set of processes is decided on the basis of whether enough space is available in the main memory to accommodate the working set of pages for the new process.

Working set change during process execution. Pages are added or deleted and critical changes may occurs as the process requires a completely different working set. This initial and sub sequent working set of a process may differ completely in size and contents. It complicates the storage management under a working set strategy.

Q 76. What are different memory allocation strategies? **(PTU, Dec. 2013)**

Ans. (i) Contiguous allocation
(ii) Linked allocation
(iii) Indexed allocation

Q 77. Compare the following main memory organization schemes : contiguous memory allocation, pure segmentation and pure paging with respect to the following issues :
(i) External fragmentation (ii) Internal fragmentation (iii) Ability to share code across processes. **(PTU, Dec. 2013)**

Ans. (A) Contiguous memory allocation (variable size method) :

(i) External Fragmentation : There is external fragmentation, such as address space are allocated contiguously and holes develop as finished processes release its space and new processes are allocated and the size of the new process is almost smaller than the old one.

(ii) Internal Fragmentation : There is no internal fragmentation.

(iii) Ability to share code across processes. It does not allow process to share code.

(B) Pure segmentation :

(i) External Fragmentation : There is external fragmentation, such as fragmentation would occur as segments of finished processes are replaced by segments of new processes and the size of the new process is almost smaller than the old one.

(ii) Internal Fragmentation : There is no internal fragmentation.

(iii) Ability to share code across processes :
Able to share code between processes.

(C) Pure Paging :

(i) External Fragmentation : There is no external fragmentation.

(ii) Internal Fragmentation : There is internal fragmentation.
(It appear in the last frame because the process size almost not a multiple of page size).

(iii) Ability to share code across processes : Able to share code between processes.

Q 78. How address calculation is done in segmentation? Give memory partition of 100 K, 500 K, 200 K and 600 K (in order). How would each of the first fit, best fit and worst fit algorithm place process of 212 K, 417 K and 112 K and 426 K (in order)? Which algorithm makes most efficient use of memory? **(PTU, Dec. 2013)**

Ans. Segmentation is a memory management scheme that supports user view of memory.

First fit :
In first fit 212 K is put in 500 K partition.
417 K is put in 600 K partition.
112 K is put in 288 K partition (This is new partition 288 K = 500 K – 212 K)
426 K must wait.

Best fit :
In best fit 212 K is put in 300 K partition
417 K is put in 500 K partition
112 K is put in 200 K partition
426 K is put in 600 k partition

Worst fit :
In worst fit 211 K is put in 600K partition
417 K is put in 500 K partition
112 K is put in 388 K partition
426 K is must wait.

In this best-fit turns out to be the best because there is no wait processes.

Q 79. What are the major tasks of memory management ? **(PTU, May 2014)**

Ans. Memory management is the functionality of an operating system which handles or manages primary memory. Memory management keeps track of each and every memory location either it is allocated to some process or it is free. It checks how much memory is to be allocated to processes. It decides which process will get memory at what time. It tracks whatever some memory gets freed or unallocated and correspondingly it updates the status.

We can also say that the core task of any memory management system is to bring programs in to main memory for execution by processor.

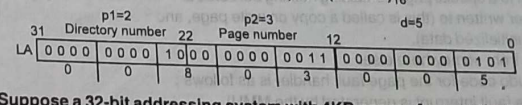
Q 80. What is the conceptual difference between fixed-size partitioning and paging? **(PTU, May 2014)**

Ans. In a fixed size partitioning of the main memory all partitions are of the same size. The memory resident processes can be assigned to any of these processes or we can also say that main memory is divided into a number of static partitions at system generation. A process may be loaded into a partition of equal or greater size. Fixed sized partitions are relatively simple to implement. Inefficient use of memory due to internal fragmentation, maximum no of active processes is fixed.

In paging, main memory is divided into a number of equal- size frames. Each process is divided into a number of equal-size pages of the name length as the frames. A process is loaded by loading all of its pages into available, not necessarily contiguous, frames. No external fragmentation,a small amount of internal fragmentation.

Q 81. Suppose a logical address LA = (8400901)10 and 4KB pages. Derive the directory number and the page number. **(PTU, May 2014)**

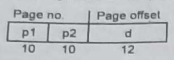
Ans. Suppose a logical address LA = $(8400901)10 = (803005)_{16}$



Q 82. Suppose a 32-bit addressing system with 4KB pages. Answer the following.
(a) Show the structure of the logical address.
(b) How many page directories have the system?
(c) What is the size of page directory entries?

(d) How many page tables have the system?
(e) What is the size of page tables?
(f) How many pages are required for each page directory or page table?
(g) Describe page table and page directory entries (size,fields). **(PTU, May 2014)**

Ans. (a)



(b) There are 10 bits which select an entry in the page directory so there are 2^10= 1024 entries in page directory.

(c) Size of page directory entries are 1024 × 4 bytes = 4KB

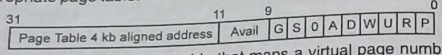
(d) 32 bit address, 4KB pages require 2^{20} entries in the page table.

• Break the page table into one directory page of 1024 entries on each of the sub page (still one million entries total).

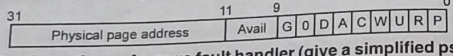
(e) 1024 × 1024 × 32 bits = 4MB

(f) 2^20 Pages 4KB Pages in the system.

(g) Page directory entry : Entry in a page directory that make to the base address of a page table, which stores page table entries or we can also say that a page directory value that specifies the address of the appropriate page table.



Page Table entry : Entry in a page table that maps a virtual page number to a page frame number. Page table entries store other information about a page, such as how the page may be accessed and whether the page is resident or we can also say that a page table value that contains the actual physical address of the page that holds the referenced code or data.



Q 83. Describe the actions of a page fault handler (give a simplified pseudo-code). **(PTU, May 2014)**

Ans. In the case of our hypothetical application, the CPU first presents the desired address (12374) to the MMU. However, the MMU has no translation for this address. So, it interrupts the CPU and cause software, known as a page fault handler, to be executed. The page fault handler then determines what must be done to resolve this page fault it can :

- Find where the desired page resides on disk and read it in (this is normally the case if the page fault is for a page of code)
- Determine that the desired page is already in RAM (but not allocated to the current process) and reconfigure the MMU to point to it.
- Point to a special page containing nothing but zeros and later allocate a page only if the page is ever written to (this is called a copy on write page, and if often used for pages containing zero-initialized data).
- Get it from somewhere else.

The pseudo code for the page fault handler is as follows :

0. Page fault interrupt is generated by the MMU)
1. Acknowledge Interrupt

(CPU will receive vector from the MMU and call the page fault interrupt handler) :

.....

1. Save the context from the CPU into the head of the Ready Que.
2. Perform a disk read operation using the location on disk information in the page table
3. Remove the head of the Ready Queue and insert into the DMAQ
   The operating system must also introduce an additional variable in the PCB that can be used to indicate the reason for the DMA disk request.
   – because an ordinary disk read (to obtain disk data is handled differently than a disk read to fetch a missing page.)
4. Update current page table register of MMU to point to the page table of the program that is the new head of the Ready Queue.
   (This will switch the MMU to the page table from the program at the head of the ready queue.)
5. Restore the context using the head of the Ready Queue.

Q 84. What is page fault rate ? **(PTU, May 2014)**

Ans. The page fault rate is defined as the number of page faults per memory reference.

Q 85. Describe dynamic partitioning. What is the main problem in dynamic partitioning ? How is it solved ? **(PTU, May 2014)**

Ans. Partitions are of variable length and number. Each process is allocated exactly as much memory as it requires. Eventually holes are formed in main memory. This is called external fragmentation. Must use compaction to shift processes so they are contiguous and all free memory is in one block. used in IBM's OS/MVT (Multiprogramming with a variable number of Tasks)

Dynamic partitioning of memory is not without disadvantages. It requires more complex bookkeeping and memory management algorithms, thus consuming more of the operating system's space and time. Though internal fragmentation may be negligible, external fragmentation may become a serious problem, imposing a time penalty for compaction. Sharing may also be complicated when shared object are subjected to compaction of memory. One solution to this problem is compaction.

Q 86. Consider the following snapshot of a system :

	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	0 0 1 4	0 6 5 6	

Answer the following questions using the banker's algorithm :
(a) What is the content of the Matrix Need ?
(b) Why are segmentation and paging sometimes combined into one scheme ?

Ans. (a) Need = Max – Allocation **(PTU, May 2014)**

The values of Need for processes Po through P4 respectively are (0,0,0,0), (0,7,5,0), (1,0,0,2), (0,0,2,0) and (0,6,4,2).

	Need			
	A	B	C	D
P0	0	0	0	0
P1	0	7	5	0
P2	1	0	0	2
P3	0	0	2	0
P4	0	6	4	2

(b) Segmentation and paging are often combined in order to improve upon each other. Segmented paging is helpful when the page table becomes very large. A large contiguous section of the page table that is unused can be collapsed into a single segment table entry with a page table address of zero. Paged segmentation handles the case of having very long segments that require a lot of time for allocation. By paging the segments, we reduce wasted memory due to external fragmentation as well as simplify the allocation.

Q 87. An operating system supports a paged virtual memory, using a central processor with a cycle time of 1 microsecond. It costs an additional 1 microsecond to access a page other than the current one. Pages have 1000 words, and the paging device is a drum that rotates at 3000 revolutions per minute and transfers 1 million words per second. The following statistical measurements were obtained from the system :

- **1 percent of all instructions executed accessed a page other than the current page.**
- **Of the instructions that accessed another page, 80 percent accessed a page already in memory.**
- **When a new page was required the replaced page was modified 50 percent of the time.**

Calculate the effective instruction time on this system, assuming that the system is running one process only, and that the processor is idle during drum transfers. (PTU, May 2014)

Ans. Effective access time, EAT

$= 0.99 \times (1\mu sec + 0.008\times (2\mu sec) + 0.002 \times (10{,}000\mu sec + 1{,}000\mu sec)$

$+ 0.001 \times (10{,}000\mu sec + 1{,}000\mu\ sec)$

$= (0.99 + 0.016 + 22.0 + 11.0)\mu sec$

$= 34.0\mu sec$

Q 88. Write policy. **(PTU, Dec. 2014)**

Ans. When a system writes a datum to cache, it must at some point write that datum to backing store as well. The timing of this write is controlled by what is known as the write policy. There are two basic writing approaches :

1. Write - through
2. Write - back (or write behind)

Q 89. Overlays. **(PTU, Dec. 2014)**

Ans. The entire program and data of a process must be in the physical memory for the process to execute. The size of a process is limited to the size of physical memory. If a process is larger than the amount of memory, a technique called overlays can be used. Overlays is to keep in memory only those instruction and data that are needed at any given time. When other instructions are needed, they are loaded into space that was occupied previously by instructions that are no longer needed.

Overlays are implemented by user, no special support needed from operating system, programming design of overlay structure is complex.

Q 90. LIFO v/s FIFO **(PTU, Dec. 2014)**

Ans. LIFO stands for Last in, First out. LIFO is a term that is used to refer to whatever was added last is used first.

FIFO stands for first In, First out. FIFO is a term that is used to refers to use whatever was added first.

Q 91. Dynamic loading. **(PTU, Dec. 2014)**

Ans. In dynamic loading, the process or subroutine is not loaded into the memory from the disk until it is actually referenced, thus giving us a better memory-space utilization. If the process references other processes or subroutines, the dynamic loader first checks whether it has already been loaded in the memory or not. If the process in the memory, then it is executed, else relocatable loader is used to bring the process/subroutine into the memory and control is passed on the newly loaded process.

Dynamic loading requires the user to design the program in such a way so as to take advantage of this scheme since it is not dependent on the OS. The main advantage of dynamic loading can be felt in systems that handle infrequently occuring memory loading events, e.g. for an error or exception handling.

Q 92. Explain the need of virtual memory ? **(PTU, Dec. 2014)**

Ans. Storage allocation has always been an important consideration in computer programming due to the high cost of main memory and the relative abundance and lower cost of secondary storage. Program code and data required for execution of a process must reside in main memory to be executed, but main memory may not be larger enough to accomodate the needs of an entire process. Early computer programmers divided programs into sections that were transferred into main memory for a period of processing time. As the program proceeded, new section moved into main memory and replaced sections that were not needed at that time. In the early era of computing the programmer was responsible for devising this overlay system. As higher level language became popular for writing more complex programs and the programmer become less familiar with the machine, the efficiency of complex programs suffered from poor overlay systems. The problem of storage allocation became more complex.

Two theories for solving the problem of inefficient memory management emerged-static and dynamic allocation.

Q 93. Differentiate between Local and Global Page Replacement. **(PTU, Dec. 2019 ; May 2015)**

Ans. In a local page replacement algorithm, only pages belonging to the process that has page faulted are considered for replacement.

Local replacement requires that each process select from only its own set of allocated frames. Here the number of frames allocated to a process does not change.

In a global page replacement algorithm, pages considered by any process are considered. Global page replacement 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.

Other words; if process P generates a page fault, page can be selected in two ways :

Select for replacement one of its frames.

Select for replacement a frame from a process with lower priority number.

So we can also say that global replacement allows a process to select a replacement frame from the set of all frames, even if that frame belongs to some other process; one process can take a frame from another. Local replacement requires that each process select from only its allocated frames.

Q 94. Differentiate between paging and segmentation scheme of memory management in detail. **(PTU, May 2015)**

Ans.

Segmentation	Paging
1. Program is divided into variable size segments.	1. Program is divided into fixed size pages.
2. User (or compiler) is responsible for dividing the program into segments.	2. Division into pages is performed by the operating system.
3. Segmentation is slower than paging.	3. Paging is faster than segmentation.
4. Segmentation is visible to the user.	4. Paging is invisible to the user.
5. Segmentation eliminates internal fragmentation.	5. Paging suffers from internal fragmentation.
6. Segmentation suffers from external fragmentation.	6. There is no external fragmentation.
7. Processor uses page number, offset to calculate absolute address.	7. Processor uses segment number, offset to calculate absolute address.
8. OS maintain a list of free holes in main memory.	8. OS must maintain a full frame list.

Q 95. What are the major advantages of partitioned allocation ? **(PTU, Dec. 2015)**

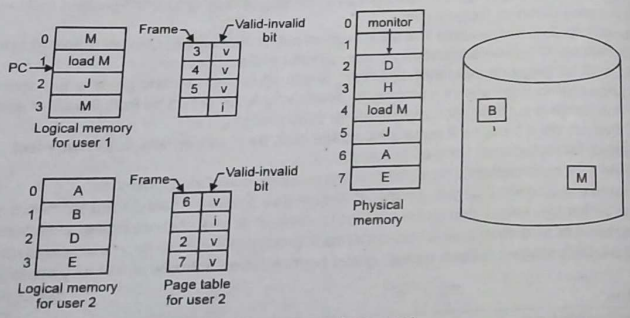
Ans. Partitioned allocation offers three major advantages

1. If facilitates multiprogramming, hence, more efficient utilization of the processor and I/O devices.
2. It requires no special costly hardware.
3. The algorithm used are simple and easy to implement.

Q 96. What is the need of Page replacement ? Consider the following reference string 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1

Find the number of Page Faults with FIFO, Optimal Page replacement and LRU with three free frames which are empty initially. Which algorithm gives the minimum number of page faults ? **(PTU, May 2015)**

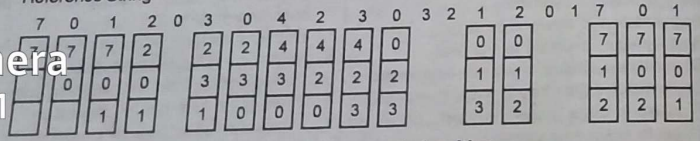
Ans. Page replacement algorithms are the techniques using which operating system decides which memory pages to swap out, write to disk when a page of memory needs to be allocated. Paging happens whenever a page fault occurs and a free page cannot be used for allocation purpose accounting to reason that pages are not available or the number of free pages is lower than required pages. When the page that was selected for replacement and was paged out, is referenced again then it has to read in from disk, and this requires for I/O completion. This process determines the quality of the page replacement algorithm; the lesser the time waiting for page-ins, the better is the algorithm. A page replacement algorithm looks at the limited information about accessing the pages provided by hardware, and tries to select which pages should be replaced to minimize the total number of page misses, while balancing it with the costs of primary storage and processor time of the algorithm itself.



Need for page replacement

1. FIFO

Reference String

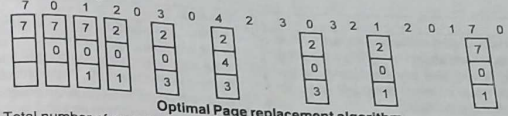


FIFO Page replacement algorithm

Total number of page faults in FIFO = 15

2. Optimal Page replacement

Reference string

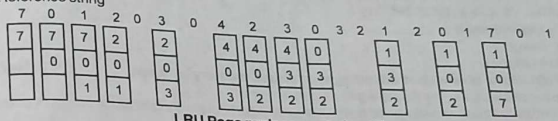


Optimal Page replacement algorithm

Total number of page faults in optimal = 9

3. LRU Page replacement

Reference string



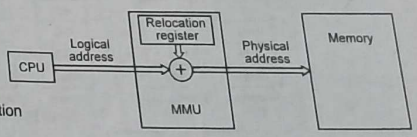
LRU Page replacement algorithm

Total number of page faults in LRU = 12

Q 97. Discuss about logical versus physical address space with an example. **(PTU, Dec. 2015)**

Ans. The address generated by the CPU is called the logical address. The address seen by the memory is called the physical address. In compile time and load time, address binding schemes, the logical and physical addresses are equal. However, in run-time binding scheme, the logical and physical addresses differ. In this scheme the logical addresses is called the virtual address. The set of all logical addresses is called the logical address space and the set of all physical addresses is called the physical address space.

The translation from logical address to physical address is done by a hardware section called the memory management section.



Logical to physical address translation by memory management section.

The address generated by the CPU is combined with the contents of the relocation register to generate the physical address. Thus if the logical address is 36H and the relocation register contents are 1000H, then the physical address is 1036H. For this scheme to work, the program should be relocatable, i.e. devoid of any absolute addresses.

Q 98. What are consequences of choosing large page sizes, and what of small page sizes? **(PTU, May 2016)**

Ans. Large pages :

Pro : smaller page table, less page faults, less overhead in reading/writing of pages.

Con : more internal fragmentation, worse locality of reference.

Smaller pages :

Pro : reduces internal fragmentation, better with locality of reference

Con : bigger page table, more page faults, overhead in reading/writing of pages.

Q 99. What is the difference between dynamic and static linking ? **(PTU, Dec. 2019 ; May 2016)**

Ans.

Static	Dynamic
1. Static linking is performed by programs called linkers as the last step in compiling a program.	1. Dynamic linking is performed at run time by the operating system.
2. Statically linked files are significantly larger in size because external programs are built into the executable files.	2. In dynamic linking only one copy of shared library is kept in memory. This significantly reduces the size of executable programs, thereby saving memory and disk space.

Q 100. Write various goals of Security. **(PTU, May 2017)**

Ans. Various goals of Security :

1. Confidentiality
2. Integrity
3. Availability

Q 101. Write at least two advantages of virtual memory concept. **(PTU, May 2018)**

Ans. Following are the advantages of virtual memory :

(i) Virtual memory allows speed gain when only a particular segment of the program is required for the execution of the program.

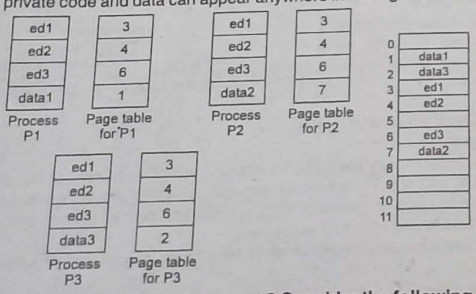
(ii) More processes may be maintained in main memory.

(iii) Makes programming easier, there no longer need to worry about the memory size limitations.

Q 102. Explain with an example the concept of shared pages in detail. **(PTU, Dec. 2017)**

Ans. Shared code : In this one copy of read only code shared among processes and similar to Multiple threads sharing the same process space. It is also useful for interprocess communication if sharing of read-write pages is allowed.

Private code and data : In this each process keeps a separate copy of the code and data and the pages for the private code and data can appear anywhere in the logical address space.



Q 103. What is the need of Page replacement ? Consider the following reference string 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1 Find the number of Page Faults with FIFO, Optimal Page replacement and LRU with four free frames which are empty initially. Which algorithm gives the minimum number of page faults ? **(PTU, May 2018)**

Ans. 1. FIFO

Reference string

7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1

F4				2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1
F3			1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
F2		0	0	0	0	0	0	4	4	4	4	4	4	4	4	4	7	7	7
F1	7	7	7	7	7	3	3	3	3	3	3	3	3	2	2	2	2	2	2
	•	•	•	•	Hit	•	Hit	•	Hit	Hit	•	Hit	Hit	•	•	Hit	Hit	•	Hit	Hit

• : Total No. of page faults = 10

2. Optimal Page Replacement

Reference string

7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1

F4				2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
F3			1	1	1	1	4	4	4	4	4	4	4	4	4	4	4	4	4
F2		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F1	7	7	7	7	7	3	3	3	3	3	3	3	3	3	3	3	7	7	7
	•	•	•	•	Hit	•	Hit	•	Hit	Hit	Hit	Hit	Hit	•	Hit	Hit	Hit	•	Hit	Hit

• : Total No. of page fault in optimal page replacement = 8

3. LRU Page Replacement :

Reference String

7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1

F4				2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
F3			1	1	1	1	4	4	4	4	4	4	4	4	4	4	4	4	4
F2		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F1	7	7	7	7	7	3	3	3	3	3	3	3	3	3	3	3	7	7	7
	•	•	•	•	Hit	•	Hit	•	Hit	Hit	Hit	Hit	Hit	•	Hit	Hit	Hit	•	Hit	Hit

• : Total No. of page fault in LRU = 8

So the Optimal page replacement and LRU both give same number of Page faults which is minimum than FIFO Page replacement.

Q 104. Consider the following segment table :

Segment	Base	Length
0	219	600
1	2300	014
2	90	100
3	1327	580
4	1952	96

What are the physical addresses for the following logical addresses ? Explain :
(a) 0,430 (b) 1,10 (c) 2,500 (d) 3,400 (e) 4,112 **(PTU, May 2016)**

Ans. (a) 0,430 :

Segment 0 has a length of 600, which is greater than 430

So, 219 + 430 = 649

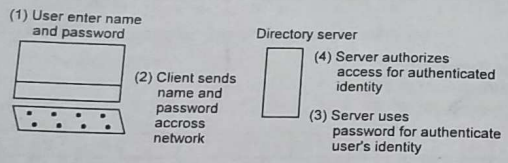
(b) 1, 10 :
Similarly segment 1 has a length of 14, which is greater than 10.
So, 2300 + 10 = 2310
(c) 2500 :
illegal reference; traps to operating system.
(d) 3,400
Segment 3 has a length of 580, which is greater than 400
So, 1327 + 400 = 1727
(e) illegal reference; traps to operating system.

Q 105. What is the primary goal of authentication and how can you achieve that through the password mechanism ? **(PTU, Dec. 2019)**

Ans. Authentication is important because it enables organizations to keep their network secure by permitting only authenticated user or processes to access its protected resources which may include computer systems, networks, databases, websites and other network based applications or services.

Simple password authentication offers an easy way of authenticating users. In password authentication the user must supply a password for each server, and the administratior must keep track of the name and password for each user, typically on separate servers.

Steps in password based authentication



Password based authentication

In the above figure, the following are the steps :

1. The user enters a name and password.
2. The client sends the DN and password across the network.
3. The server determines whether the password sent from the client matches the password stored for the entry with the DN sent from the client. If so, the server accepts the credentials as evidence authenticating the user identity.
4. The server determines whether the identified user is permitted to access the requested resource. If so, the server allows the client to access the resource.

❑❑❑

FOR NOTES

Chapter 6

Device Management

Contents

I/O devices, Device Controllers, Direct memory access, Principles of I/O software : Goals of Interrupt handlers, Device drivers, Device independent I/O software, Secondary Storage Sctructure : Disk structure, Disk scheduling algorithms.
File Management : Concept of File, Access methods, File types, File opeation, Directory structure, File system structure, Allocation methods (contiguous, linked, indexed), Free Space Management (bit vector, linked list, grouping), directory implementation (linear list, hash table), efficiency and performance.
Disk Management : Disk structure, Disk scheduling - FCFS, SSTF, SCAN, C-SCAN, Disk reliability, Disk formatting, Boot-block, Bad blocks.

POINTS TO REMEMBER

- Disk scheduling algorithm can improve the effective bandwidth, the average response time and the variance in response time. Algorithm such as SSTF, SCAN, E-SCAN, Look and C-Look are designed to make such improvements by strategies for disk queue ordering.
- The operating system manages the disk blocks. A disk must be low level formatted to create the sectors on the raw hardware new disks usually can preformatted.
- Free space allocation methods also influence the efficiency of use of disk space. The performance of the file system depend on the reliability of secondary storage.
- I/O system calls are costly in terms of CPU consumption because of the many layers of software between a physical device and the application.
- The work of moving data between devices and main memory is performed by the CPU as programmed I/O or is off loaded to a DMA controller.
- Performance can be formed by external fragmentation. Some system have utilities that scan the file system to identify fragmented files, they can then more blocks around to decrease the fragmentation.
- Input scheduler performs the same job as scheduler. It allocates devices control units and channels.
- I/O devices are mapped into the system memory map along with RAM and ROM.
- In port mapped I/O devices are mapped into a separate address space.
- I/O system calls are costly in terms of CPU consumption because of the many layers of software between physical device and the application.
- The system call interface provided to applications is designed to handle several basic categories

of hardware including block devices, character devices, memory mapped files network sockets and programmed interval timers.
- An efficient swap spaces is a key to good performance system usually bypass the file system and use raw disk access for paging I/O.
- Time taken to move arms to position disk head on the desired track called seek time.
- Disk scheduling is a term used to describe the method computer operating systems decide the order that block I/O operations will be submitted to storage volumes.
- RAID, an acronym for redundant array of independent disks or redundant array of inexpensive disks a technology that provides increased storage reliability through redundancy.
- Disk scheduling algorithms are first in first out, shortest seek time first, scan, circular scan, loop, circular loop.
- When the operating system fails to read the stored data and instructions from a disk, the disk is said to have crashed.
- A file is an abstract data defined and implemented by the operating system. It is sequence of logical records. A logical record may be a byte a line or a more complex data item.
- A single level directory in a multi user system causes naming problem. Since each file must have a unique name.
- A two level directory solve this problem by creating a separate directory for each uses.
- The directory lists the files by name and include such information as the file allocation on the disk length, type, owner, time of creation time of last use and so on.
- A tree structural directory allows a uses to create sub directories to organize this files. A cyclic group directory structures allow sub directories and files to be shared but complicate searching and deletion.
- Disks are segmented into one or more partitions each containing a file system or left 'row'. File system may be mounted into the system naming structures to make them available.
- File sharing depends on the semantics provided by the system. Files may have multiple readers, multiple writers, or limits on the sharing.
- The file system resides permanently on secondary storage which is designed to hold a large amount of data permanently.
- Every file system type can have different structure and algorithm.
- The various files can be allocated space on the disk in three worst through contiguous linked or indexed allocation.
- Direct occurs is very inefficient with linked allocation.
- Indexed allocation can be done in clusters of multiple block to increase throughput and to reduce the number of index entries needed.
- Protection is an internal problem. Security must consider both the computer system and the environment people, buildings, business, valuable objects and threats within which system is used.
- The data stored in the computer system must be protected from unauthorized access, malicious destruction or alternation and accidental introduction of inconsistency.
- One time password changes the sent data each time to avoid replay attacks.
- Two factors authentication requires two forms of authentication such as hardware calculator with an activation PIN.
- There are several types of attacks that can be launched against individual computer or the masses.

QUESTION-ANSWERS

Q 1. Explain device management functions. **(PTU, Dec. 2018)**

Ans. The two main jobs of a computer are I/O and processing, it becomes essential to know the role of operating system in managing and controlling the input/output operations and input/output devices.

1. Track the status of each device (such as tape drives, disk drives, printers, plotters and terminals).
2. Allocate the devices.
3. Deallocate the devices at two levels :

(a) At command level, when input/output command has been executed and the device has been temporarily released.

(b) At process level, when process has terminated and the device has been permanently released.

Q 2. How does operating system manage devices?

Ans. The devices can be managed and allocated by an operating system in three possible ways :

1. Dedicated : An operating system can use a device in a dedicated manner by assigning it to only one process at a time, such device will remain assigned to only one process at a time and will serve that process till it terminates.

2. Shared devices : The operating system can assign a device in such a way that it can be shared among several processes. Example of the devices that can be shared include printers, hard disk, etc.

3. Virtual devices : Some devices that would normally have to be dedicated (card readers, printers) may be converted into shared devices through technique such as SPOOLING.

Q 3. What is memory mapped I/O?

Ans. Memory Mapped I/O (MMIO) and port (I/O) are two complements by methods of performing input/output between CPU and peripheral devices in a computer. Memory mapped I/O uses the same address bus to address both memory and I/O devices and the CPU instructions used to access the memory and are also used for accessing devices. In order to accomodate the input/output devices, areas of the CPUs addressable space must be reserved for input/output. The reservation might be temporary - the commodare 64 could bank switch between its input/output devices and regular memory or permanent. Each input/output devices monitors the CPUs address bus and responds to any of the CPUs access of device-assigned address space, connecting to the data bus to a desirable device's hardware register.

Q 4. What is input/output subsystem?

Ans. Input/output devices are so very widely in their functions and speed, a variety of methods are needed to control them. These methods form the input/output subsystem of the kernel, which separates the rest of the kernel from the complexity of managing input/output devices.

The two basic components of this subsystem are :

(i) Input/output channels.

Q 5. What is post mapped I/O?

Ans. An computing I/O, refers to the communication between an information processing system and the outside world possibly a human, or another information process of system. Inputs are the signals or data received by the system, and outputs are the signals or data sent from it. The term can also be used as part of action ; to "perform I/O" is to peform an input or output operation. I/O devices are used by a person (or other system) to communicate with a computer. For instance, a keyboard or a mouse may be an input device for a computer, while monitors and printers are considered output devices for a computer. Devices for communication between computers. Such as modems and network cards, typically serve for both input and output.

Q 6. Explain I/O channels.

Ans. Input channels are used to provide a path or a channel for the data to flow between input/output devices and the main memory. Input/output channels use channel programs that specify action to be performed by devices and control the transmission of data between main memory and control units. They are required to resolve the disparity in speed of the input/output devices and the CPU. They are specialized in processing units. If all input and output are performed via the channel, the CPU is free to perform its high-speed computation without wasting time or slow operations such as reading cards.

Q 7. Types of I/O channels.

Ans. I/O channels use channel programs that specify action to be performed by devices and control the transmission of data between main memory and control units.

Types of I/O channels :

1. Selector channel.
2. Multiplexer channel.

Q 8. Explain I/O control units. **(PTU, May 2017)**

Ans. Due to the high cost of channels, i.e., input-output channels, there are fewer channels than devices. The channels, must, therefore, be switched from one device to another. This switching of devices can be done by input control unit which interprets signal, sent by channel for switching from one device to another. It can be shared by many similar types of input devices (such as tapes, disks, etc.) and can interpret the commands associated with these devices.

Q 9. What are various kinds of interrupts? How we handle them? **(PTU, May 2012)**

Ans. When a process is executed by CPU and when a user request for another process, then this will create disturbance for the running process. This is called interrupt.

Types of Interrupts :

1. External interrupts
2. Internal interrupts

The **external interrupts** occurs when an input and output devices request for any operation and the CPU will execute the instructions first.

Internal interrupts : Internal interrupts are those which are occurred due to some problem in the execution.

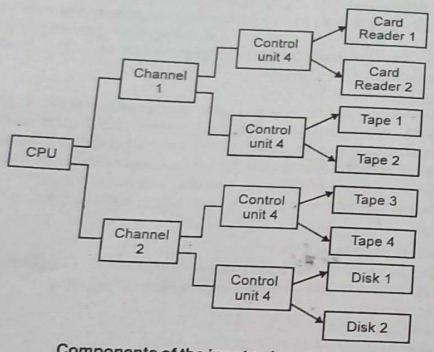
Q 10. Explain multiplexer channel.

Ans. It is capable of servicing many devices, simultaneously. It is able to accomplish this only for slow input devices such as card readers, card punches and printers.

A third type of input channel which is a compromise solution that allows multiple channel programe for highspeed devices to be active on the same input channel is block multiplex or channel.

Q 11. Explain diagrammatic approach of input subsystem.

Ans.



Components of the input subsystem

Q 12. Explain input scheduler.

Ans. Input scheduler performs the same job as process scheduler. It allocates devices, control units and channels. If these are more input requests pending than available paths, it is necessary to choose with input requests to satisfy that

Input requests are not preempted, i.e., once channel program has started it is allowed to continue towards completion, even though input requests with higher priorities may have entered the queue. Input scheduler must synchronize its work with traffic controller to make sure that a path is available to satisfy input requests.

Q 13. Give the comparison of port mapped I/O and memory mapped I/O.

Ans. Memory Mapped I/O : I/O devices are mapped into the system memory map along with RAM and ROM. To access a hardware device, simply read or write to those 'special' addresses using normal memory access instructions. The advantage to this method is that every instruction which can access memory can be used to manipulate I/O device. The disadvantage to this system is that the entire address bus must be fully decoded for every device.

Port Mapped I/O : I/O devices are mapped into a separate address space. This is usually accomplished by having a different set of signal lines to indicate a memory access versus a part access. The address lines are usually shared between the two address spaces, but less of them are used for addressing ports. An example of this is the standard PC which uses 16 bits of port decode a discrete address and, therefore, less cost to add hardware device to a machine. On the older PC compatible machines, only 10 bits of address and, therefore, less cost to add hardware devices to a

machine, on the older PC compatiable machines. Only 10 bits of address space were decoded for I/O ports and so there were only 1024 unique port locations ; modern PC decode disk 16 address lines. To read or write from a hardware device, special port I/O instructions are used the same task. For instance, if you want to test one bit on memory mapped port, there is a single instruction to test a bit memory, but for ports you must read the data into register, then test a bit.

Q 14. What would be the effect of the system running too many input jobs?

Ans. To start an input operation, the CPU loads the appropriate registers within the device the contents. The device controller, in turn, examines the contents of these registers to determine what action to take. Once the input started, two courses of action are possible. In the simplest case, the input is started, than at input completion, control is returned to the user process. This case is known as synchronous input. The other called asynchronous input, returns control to user program without waiting for the input to complete. **(PTU, Dec. 2003)**

An input device interrupts when it needs service. When an interrupt occurs, the operating system first determines which input device caused the interrupt. It then indexes into the input device table to determine the status of that device and modifies the table entry to affect the occurrence of the interrupt.

Q 15. What is interrupt? What happens when an interrupt occurs?

Ans. It may be more efficient to arrange for the hardware controller to notify the CPU when the device becomes ready for service, rather than to require the CPU to poll repeatedly for an I/O completion. The hardware mechanism that enables a device to notify the CPU is called an interrupt. **(PTU, Dec. 2007, 2004)**

When the CPU detects that a controller has asserted a signal on the interrupt request line, the CPU saves a small amount of state, such as the current value of the instruction pointer and jumps to the interrupt handler routine at a fixed address in memory. The interrupt handler determines the cause of the interrupt, performs the necessary processing and executes a return from interrupt instruction to return the CPU to execution state prior to the interrupt.

Q 16. Explain I/O device handler.

Ans. In addition to setting up the channel command words, handling error conditions and processing input interrupts, the input device handlers provide detailed scheduling algorithms that are dependent upon the type of the device. There is usually a different device handler algorithm for each type of input device. Scheduling algorithm for disks have been the motive of minimizing three factors arm movement, mean response time and variance in response time.

Q 17. Explain seek time, rotational delay and transfer time. **(PTU, May 2019, 2018, 2015)**

Ans. Seek time : Time taken to move arms to position disk head on the desired track.

Rotational delay : Time spent in waiting for desired block to rotate under the head.

Transfer time : Time spent in actually moving data to/from the disk surface.

Q 18. Explain secondary storage (Disk) structure. **(PTU, May 2018)**

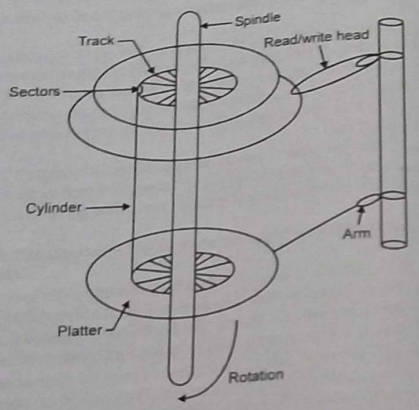
Ans. Disks are secondary storage devices of choice. Their main advantage over tapes is that they allow random access to the data stored on them. Data is stored and retrieved in units called disk blocks or pages. Unlike RAM, time to retrieve a disk block varies as it depends upon the location of the block on the disk.

It consists of platters that spin around the spindle. The two surfaces of a platter are converted with a magnetic material similar to that on a magnetic tape. Information is recorded magnetically on the platters.

The surface of the platter is logically divided into circular tracks, which are subdivided sectors. The set of tracks that are at one arm position form a cylinder. There may be thousand concentric cylinders in the disk derive.

Q 19. Explain diagrammatic approach of disk structure.

Ans.



Q 20. What is RAID?

Ans. RAID stand for Redundant Array of Independent Disks. A category of disk drives employ two or more drives in combination for fault tolerance and performance. RAID disk drives used frequently on servers but are not generally necessary for personal computers. RAID allows to store the same data redundantly (in multiple faces) in a balanced to improve overall performa There are number of different RAID levels :

Level 0 : Striped disk array without tolerance :

Provides data striping (spreading out blocks of each file across multiple disk drives) bu redundancy. This improves performance but does not deliver fault tolerance. If one drive fails th data in the array lost.

Level 1 : Mirroring and Duplexing : Provides disk mirroring level 1 provides twice the r transaction rate of single disks and the same write transaction rate as single disks.

Level 2 : Error correcting coding : Not a typical : Not a typical implementation and rarely us

Level 2 stripes data at the bit level rather than the block level.

it - Interleaved Party : Provides byte-level striping with a dedicated party disk. Le

3. which cannot service simultaneous multiple requests, also is rarely used.

Level 4 : Dedicated party Derive : A commonly used implementation of RAID, level 4 provi block - level striping (Like Level 0) with a parity disk. If a data disk fails, the party data is use

create a replacement disk. A disadvantage to level 4 is that the parity disk can create write bottlenecks.

Level 5 : Block interleaved distributed parity : Provides data striping at the byte level and also stripe error correction information. This results in excellent performance and good fault tolerance. Level 5 is one of the most popular implementations of RAID.

Level 6 : Independent data disks with double parity : Provides block-level striping with parity data distributed across all disks.

Level 0 + 1 : A Mirror of stripes : Note one of the original RAID levels, two RAID 0 strikes are created, and a RAID 1 mirror is created over them. Used for both replicating and sharing data among disks.

Level 10 : A stripe mirrors : Not one of the original RAID levels, multiple RAID 1 mirrors are created and the RAID 0 stripe is created over these.

Level 7 : A trade mark of storage computer corporation that adds caching to levels 3 or 4.

Q 21. Explain the structure of Disc in brief.

Ans. A compact disk is an optical disc used to store digital data. It was originally developed to store sound recordings exclusively, but later it also allowed the preservation of other types of data. Audio CDs have been commerically available since October 1982. In 2010, they remain the standard physical storage medium for audio.

Standard CDs have a diameter of 120 mm and can be hold up to 80 minutes of uncompressed audio (700 MB of Data). The mini CD has various diameters ranging from 60 to 80 mm; they are sometimes used for CD signals or device drivers, storing up to 24 mns of audio.

The technology was eventually adopted and expanded to encompass data storage CD-ROM, Write-once audio data storage CD-R, rewritable media CR-RW, Video Compact Discs (VCDs), super video compact discs (SVCD), Photo CD, Picture CD CD-i and Enchanced CD.

CD-ROM and CD-RS remain widely used technologies in the computer industry. The CD and its extension are successful in 2004, worldwise sales of CD audio, CD-ROM and CD-R reached about 30 billion discs. By 2007 billion CDs have been sold worldwide.

- Sectors
- Tracks
- Cylinders

A disk is accessed as an array : sector 0 is the first sector of the top track of the outermost cylinder. The next sectors of the same track are then ordered. Then, sectors from the next track are ordered. After all the sectors of all the tracks are orders, we move to the next cylinder. The innermost is the last cylinder disk arm.

Q 22. What is disk scheduling?

Ans. Disk scheduling is a term used to describe the method computer operating systems decide the order that block I/O operations will be submitted to storage volumes. I/O scheduling is sometimes called 'disk scheduling'.

Purpose : I/O schedulers can have purpose depending on the goal of the I/O scheduler. Some common goals are :

1. To minimize time wasted by hard disk seeks.
2. To prioritize a certain process I/O requests.
3. To give a share of the disk bandwidth to each running process.

Q 23. What is boot sector? Explain its types.

Ans. A boot sector of a hard disk, floppy disk or similar data storage device that contains code for booting programs stored in other parts of the disk.

On a IBM PC compatible machine the BIOS selects a boot device, then it copies the first sector from the device to address location 0×7.

Types of Boot Sector :

1. A master boot record is the first sector of a data storage device that has been partitioned. The MBR sector may contain code to locate the active partition and invoke its volume boot record.
2. A volume boot record is the first sector of a data storage device that has not been partitioned or the first sector of an individual partition on a data storage device that has been partitioned.

Q 24. Explain the objectives of disk scheduling algorithm.

Ans. 1. Minimizing the response time, i.e., the average time that a request must wait before it is satisfied.

2. Maximizing the throughput, i.e., the average number of requests satisfied per unit of time.

Q 25. What is RAID? Explain types.

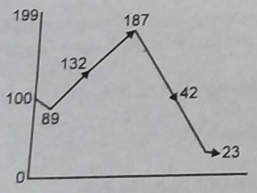
Ans. RAID, an acronym for redundant array of independent disks or redundant array of inexpensive disks a technology that provides increased storage reliability through redundancy, combining multiple low cost, less-reliable disk drive components into a logical unit where all drives in the array are interdependent. This concept was first defined by DAVIDA Patterson, Carth A Gibson and Randy Katz at the University of California. Berkeley in 1987 as redundant array of inexpensive disks. Marketers representing industry RAID manufactures later reinvented the term to describe a redundant array of independent disks as a means of dissociating a low cost expectation from RAID technology.

RAID is now used as an umbrella term for computer data storage schemes that can divide and replicate data among multiple hard drives. The different schemes or architectures are named by the word RAID followed by a number (e.g., RAID0, RAID1) RAID's various designs involves two key design goals ; increase data reliability and to increase input/output performance. When multiple physical disks are set-up to use RAID technology, they are said to be in a RAID array. This array distributes data across multiple disks, but the array is seen by the computer user and operating system as one single disk, RAID can be set-up to serve several different purposes.

Q 26. Explain shortest seek time first (SSTF).

Ans. When each request is received, the disk controller selects the request that needs to access a track/sector, which is closes to the current location of the head. The throughput in this case is much better than in FIFO. However, some processes may have to wait for a long time until it request(s) are satisfied.

Example :



Total time is estimated by total arm motion :

$|100 - 89| + |89 - 132| + |132 - 187| + |187 - 42| + |42 - 23| = 273$

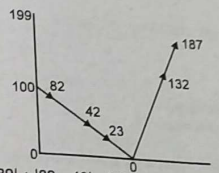
Q 27. List out various disk scheduling algorithms.

Ans. 1. First in first out (FIFO)
2. Shortest seek time first (SSTR)
3. Scan
4. Circular scan
5. Loop
6. Circular loop (C-loop).

Q 28. Explain Scan.

Ans. In this algorithm, the head always constanly moves from the mostinner cylinder to the outer cylinder, then it changes its direction back towards the centre. As the head moves, if there is a request for the current disk position, it is satisfied. The throughput is better than FIFO. The scan algorithm is more fair than SSTF as far as starvation of requests is considered

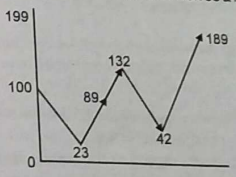
23, 89, 132, 42, 187



$|100 - 89| + |89 - 42| + |42 - 23| + 23 - 0| + |0 - 132| + |132 - 187|$

Q 29. What is first in first out algorithm?

Ans. This is the simplest algorithm. It processes the I/O requests in the same order as they arrived. This technique improves the response time as a request gets response in fair amount of time. However, the throughput is not efficient. It involves a lot of random head movements and disk rotations.



Q 30. Explain the advantages of first come first serve and also disadvantages.

Ans. Advantages :

1. It is the simple scheduling technique.
2. It will follow the order of request. It means that first in first out will be followed up.
3. As the request will be placed in queue that will be processed without observing the higher order or lowest order.

Disadvantages :

1. It does not provide fastest service to us.
2. Reading and writing time to the disk will take more time.

Q 31. Explain loop scheduling.

Ans. In this scheduling head will move in any direction according request done by the user. In this scheduling head will move in any direction it can be forward or backward.

e.g.

Process	Block
P1	92
P2	152
P3	82
P4	30
P5	20
P6	50

According to this example user makes a request for a particular block, then head will be automatically shifted to that block.

Q 32. Explain advantages and disadvantages of SCAN and C-SCAN.

Ans. (i) Scan

Advantages :

1. It is simple disk scheduling technique.
2. It is faster technique than FCFS, SSTF technique.
3. It is better technique to avoid starvation of process.

Disadvantages :

1. If any process arrives before the head position, then process will have to wait until the head position reaches to that location.
2. It is not best when you want to execute a process immediately.

(ii) C-SCAN

Advantages :

1. This scheduling is to ensure that every request will be processed.
2. This is the simple technique to process different types of request.

Disadvantages :

1. The process requests before the head positions will have to wait for their turn.
2. Waiting for execution can create starvation condition for different types of processors.

Q 33. What are interrupts? Explain.

Ans. In computing, an interrupt is a asynchronous signal including the need for attention or a synchronous event in S/W indicating the need for a change in execution.

A hardware interrupt causes the processor to save its state of execution and begin execution of interrupt handler.

Q 34. What is disk managements?

Ans. These involve initialization of the disk, booting from disk, and bad-block recovery etc.

Tasks performed :

1. Disk initialization
2. Low level formatting
3. Partitioning
4. Logical formatting

Q 35. Explain disk reliability.

Ans. When the operating system fails to read the stored data and instructions from a disk, the disk is said to have crashed. In case of disk crash, the operating system begins to flash messages like :

Invalid media player error : Recovering from a disk crash may take hours. It may involve restoring of backup copies.

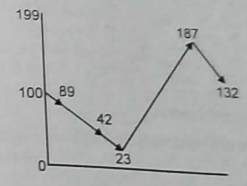
Q 36. What is tertiary storage devices? Explain types.

Ans. Low cost and reliability is defining characteristics of teritary storage. The removable and cheaper media are :

1. Floppy disk
2. Removable magnetic disk
3. Magneto-optic disk
4. Optical disks
5. WORM
6. Pre-recorded read only disks
7. Magnetic tapes.

Q 37. Explain circular loop C-loop.

Ans. This is a variation of loop where requests are satisfied only when the head moves outwards as in C-SCAN. Thus, no request is satisfied when the head moves inwards after determining that no requests are there beyond the current point.



$|100 - 89| + |89 - 42| + |42 - 23| + |23 - 187| + |187 - 132|$

Q 38. Explain I/O traffic controller. **(PTU, Dec. 2017)**

Ans. I/O scheduler is mainly concerned with policies, the I/O traffic controller is primarily concerned with mechanics (Is it possible to assign the device). The traffic controller maintains all the status informations of a device.

1. It must determine if there is at least one path available.
2. If there is more than one path available it must determine which one to select.

Q 39. Write a note on device management. How various devices work in synchronization to each other? **(PTU, May 2012)**

Ans. The two main jobs of a computer are I/O and processing, it becomes essential to know the role of operating system in managing and controlling the input/output operations and input/output devices. The devices can be managed and allocated by an operating system in three possible ways.

1. Dedicated : An operating system can use a device in a dedicated manner by assigning it only one process at a time.

2. Shared devices : The operating system can assign a device in such a way that it can be shared among several processes.

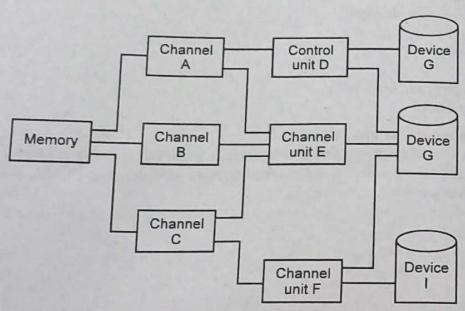
3. Virtual devices : Some devices that would normally have to be dedicated may be converted into shared devices through technique such as SPOOLING.

Synchronization of device approachs are :

- Sample clock timed devices
- Oversample clock timed devices.

Q 40. Explain diagrammatic approach of asymmetric I/O configuration.

Ans.



Asymmetric I/O Configuration

Q 41. What is a scheduler? How many types of schedulers coexist in a complex operation system? Explain. **(PTU, May 2009)**

OR

What is scheduler? Explain different types of schedulers. **(PTU, May 2007)**

Ans. Scheduler : *Process scheduler is a part of the operating system that is responsible for changing the state of process. In other words it is that component which is responsible for scheduling of the processes to be executed by the CPU. Schedulers are defined to be of three types :*

1. Long term scheduler

2. Short term scheduler

3. Medium term scheduler

1. Long term scheduler : *In select process from secondary storage device e.g. disk and leads them into memory for execution. Long term schedules is know as 'long term' because time for which scheduling is valid is long. In other words it executes less frequenty as it takes from minutes to hours for the creation of new processes in the system. Thus its jobs is to ensure that there are always fixed number of ready processor in the memory of the system.*

2. Short term scheduler : *It select among the process that are ready to execute an allocate the CPU to one of them. Short term scheduler on the other hand must have to work very often. A process must be executing for only a very short duration before it gets blocked. This requires that CPU should immediately take up another process for execution and thus the schedule should immediately select a process for the CPU.*

3. Medium term scheduler : It required at the times when a process is to be removed from the memory and thus to reduce the degree of multiprogramms. Later on this process may be put into the memory again and its execution can be again started from where it was left off. The process of moving a process in and out of the memory is called swapping. All version of windows operating system use swapping.

Q 42. What are the various techniques available for secondary storage management? Describe any two techniques. **(PTU, May 2011)**

Ans. A system will have several level of storage, including primary storage, secondary storage and cache storage. Instructions and data must be placed in primary storage and cached to be referenced by a running program. Main memory will be too small to accommodate all data and programs and its data will be lost when power is lost. Accordingly, the computer system must provide secondary storage to back up main memory.

Secondary storage devices consists of tapes, disks and other media designed to hold information that will eventually be accessed in primary storage. It will be divided into bytes or words consisting of a fixed number of bytes. Each location in storage has an address ; the set of all address available to a program is called an address mode :

The three major activities of an operating system with respect of secondary storage management are :

1. Managing the free space available on the secondary storage device.
2. Allocation of storage space when new file have to written.
3. Scheduling the requests for memory access.

Q 43. What is file system? **(PTU, May 2019 ; Dec. 2016)**

Ans. A system which provides the mechanism for online storage and access to file contents like data and programs is called file system. The file system permanently resides on secondary storage holding permanent data.

Q 44. What is file access methods?

Ans. The information in the file can be accessed in several ways :

1. Sequential access
2. Direct access
3. Other access methods are :

(i) Contiguous allocation
(ii) Linked allocation
(iii) Indexed allocation.

Q 45. Explain file naming.

Ans. The exact rules for file naming vary somewhat from system to system, but all operating systems allow strings of one to eight letters legal file names.

Thus, 'ekta', 'walia', 'lecturer' are possible file names like 'OSI', 'Sys-Prog' are also permitted.

Extension	Meaning
file.bak	Backup file
file.bas	Basic source program
file.data	datafile
file.obj	object file

1. Dedicated : An operating system can use a device in a dedicated manner by assigning it only one process at a time.

2. Shared devices : The operating system can assign a device in such a way that it can be shared among several processes.

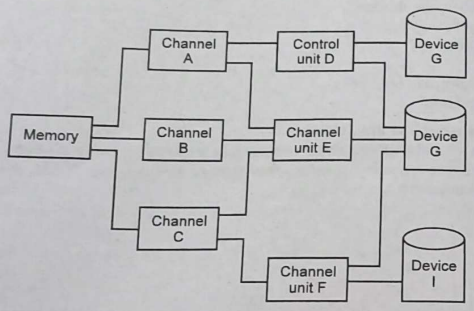
3. Virtual devices : Some devices that would normally have to be dedicated may be converted into shared devices through technique such as SPOOLING.

Synchronization of device approachs are :

- ❑ Sample clock timed devices
- ❑ Oversample clock timed devices.

Q 40. Explain diagrammatic approach of asymmetric I/O configuration.

Ans.



Asymmetric I/O Configuration

Q 41. What is a scheduler? How many types of schedulers coexist in a complex operation system? Explain. **(PTU, May 2009)**

OR

What is scheduler? Explain different types of schedulers. **(PTU, May 2007)**

Ans. Scheduler : Process scheduler is a part of the operating system that is responsible for changing the state of process. In other words it is that component which is responsible for scheduling of the processes to be executed by the CPU. Schedulers are defined to be of three types :

1. Long term scheduler
2. Short term scheduler
3. Medium term scheduler

1. Long term scheduler : In select process from secondary storage device e.g. disk and leads them into memory for execution. Long term schedules is know as 'long term' because time for which scheduling is valid is long. In other words it executes less frequenty as it takes from minutes to hours for the creation of new processes in the system. Thus its jobs is to ensure that there are always fixed number of ready processor in the memory of the system.

2. Short term scheduler : It select among the process that are ready to execute an allocate the CPU to one of them. Short term scheduler on the other hand must have to work very often. A

process must be executing for only a very short duration before it gets blocked. This requires that CPU should immediately take up another process for execution and thus the schedule should immediately select a process for the CPU.

3. Medium term scheduler : It required at the times when a process is to be removed from the memory and thus to reduce the degree of multiprogramms. Later on this process may be put into the memory again and its execution can be again started from where it was left off. The process of moving a process in and out of the memory is called swapping. All version of windows operating system use swapping.

Q 42. What are the various techniques available for secondary storage management? Describe any two techniques.

Ans. A system will have several level of storage, including primary storage, secondary storage and cache storage. Instructions and data must be placed in primary storage and cached to be referenced by a running program. Main memory will be too small to accommodate all data and programs and its data will be lost when power is lost. Accordingly, the computer system must provide secondary storage to back up main memory. **(PTU, May 2011)**

Secondary storage devices consists of tapes, disks and other media designed to hold information that will eventually be accessed in primary storage. It will be divided into bytes or words consisting of a fixed number of bytes. Each location in storage has an address ; the set of all address available to a program is called an address mode :

The three major activities of an operating system with respect of secondary storage management are :

1. Managing the free space available on the secondary storage device.
2. Allocation of storage space when new file have to written.
3. Scheduling the requests for memory access.

Q 43. What is file system? **(PTU, May 2019 ; Dec. 2016)**

Ans. A system which provides the mechanism for online storage and access to file contents like data and programs is called file system. The file system permanently resides on secondary storage holding permanent data.

Q 44. What is file access methods?

Ans. The information in the file can be accessed in several ways :

1. Sequential access
2. Direct access
3. Other access methods are :

(i) Contiguous allocation
(ii) Linked allocation
(iii) Indexed allocation.

Q 45. Explain file naming.

Ans. The exact rules for file naming vary somewhat from system to system, but all operating systems allow strings of one to eight letters legal file names.

Thus, 'ekta', 'walia', 'lecturer' are possible file names like 'OSI', 'Sys-Prog' are also permitted.

Extension	Meaning
file.bak	Backup file
file.bas	Basic source program
file.data	datafile
file.obj	object file

Q 46. Write the advantages of file access methods.

Ans. A file system is method of storing and organizing computer files and their data. Essentially, it organizes these files into a database for the storage, organization, manipulation and retrieval by the computer's operating system.

File systems and used on data storage devices such as hard disks or CD-ROMs to maintain the physical location of the files. Beyond this, they might provide access to data or a file server by acting as clients for a network protocol or they may be virtual and exist only as an access method for virtual data.

Q 47. What is proxy server?

Ans. Proxy server is a server that acts as an intermediary for requests from clients seeking resources from other servers. A client connects to the proxy server, requesting some service, such as a file, connection web page, or other resources, available from a different server. The proxy server evaluates the request according to its filtering rules, e.g., it may filter, the proxy provides the resource by connecting to the relevant server and requesting the service on behalf of the client.

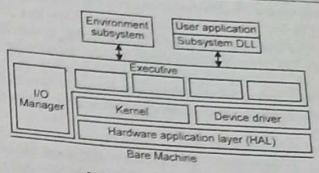
Q 48. What do you understand by the term File system and its role in Operating system ? Discuss the architecture of Windows OS. **(PTU, May 2017 ; Dec. 2015)**

Ans. File system : A computer file is a block of arbitrary information, which is available to a computer program and is usually based on some kind of durable storage. A file is durable in the sense that it remains available for programs to use after the current program has finished. Computer files can be considered as the modern counterpart of paper documents which traditionally were kept in office and library 'files', which are the source of the term.

Architecture of the windows OS : The hardware abstraction layer (HAL) interfaces with the bare machine and provides abstractions of the I/O interfaces, interrupt controllers, and inter processor communication mechanisms in a multiprocessor system. The kernel uses the abstractions provided by the HAL to provide basic services such as interrupt processing and multiprocessor synchronization. This way, the processing is shielded from peculiarities of a specific architecture, which enhances its portability. The HAL and the kernel are together equivalent to a conventional kernel. A device driver also uses the abstractions provided by the HAL to manage I/O operations on a class of devices. The kernel performs the process synchronization and scheduling functions. The executive comprises nonkernel routines of the OS; its code uses facilities in the kernel to provide services such as process creation and termination, virtual memory management, an interprocess message passing facility for client-server communication called the local procedure call (LPU), I/O management and a file cache to provide efficient file I/O and a security reference monitor that performs file access validation. The I/O manager uses device drivers, which are loaded dynamically when needed. Many functions of the executive operate in the kernel mode, thus avoiding frequent context swiches when the executive interacts with the kernel; it has obvious performance benefits.

The environment subsystems provide support for execution of programs developed for other operating systems like MS-DOS, Win 32, and OS/2. Effectively, an environment subsystem is analogous to a guest operating system with in a virtual machine OS. It operates as a process that keeps track of the

state of user applications that use its services. To implement the interface of a guest OS, each nt subsystem provides a dynamic link library (DLL) and expects a user application to invoke the DLL when it needs a specific system service. The DLL either implements the required ice itself, passes the request for service to the executive, or sends a message to the environment subsystem process to provide the service.



Architecture of windows

Q 49. List the various operations that can be performed on a file.

Ans. A file system is a method of storing and organizing computer files and their data. Essentially, it organizes these files into a database for the storage, organization, manipulation and retrieval by the computers operating system.

File systems are used on data storage devices such as hard disks or CD-ROMs to maintain the physical location on the files. Beyond this, they might provide access to data on a file server by acting as clients for a network protocol.

Q 50. Explain disadvantages of contiguous and linked list allocation.

Ans. Disadvantages :

1. Difficulty in finding space for a new file. If the file is to create is n blocks long, then the operating system must search for n free contiguous blocks.
2. This suffers from external fragmentation.
3. When the file is created, the total amount of space it will need must be known and allocated.

Linked list allocation :

Advantages :

1. There is no external fragmentation.
2. No need to declare the size of a file is created.

Disadvantages :

1. It is inefficient to support direct-access, it is effective only for sufficient sequential access files.
2. Another problem is reliability.

Q 51. What is authentication?

Ans. Authentication is the act establishing or conforming something is authentic, that is, that claims by or about the subjects are true. This might be involve confirming the identity of a person, tracing the origins of an artifact, ensuring that a product is what is packaging and labelling claims to be, or assuming that a computer program is trusted one.

Q 52. What do you mean by file management? Explain the various access and allocation methods of files in detail. **(PTU, Dec. 2019, 2016 ; May 2008)**

OR

Explain different access methods in a file system. **(PTU, May 2007)**

Ans. A file is a named collection of related information that is recorded on secondary storage. The information in a file is defined by its creator. Many different types of information may be stored in a file-source programs, object programs, executable programs, numeric data, text, payroll records

and so on. Computers can store information on several different storage media, such as magnetic disks. Files are mapped, by the operating system, on to physical devices. These storage devices are usually non-volatile, so the contents are persistent through power failures and system reboots.

Access Methods : The information in the file can be accessed in several ways. Following are the access methods :

1. Sequential Access : Information in the file is processed in order, one record after the other. This mode of access is by for the most common e.g., editors and compilers usually access files in this fashion.

2. Direct Access : A file is made-up of fixed length logical records that allow programs to read and write records rapidly in no particular order. For direct access, the file is viewed as a numbered sequence of blocks or records. A direct-access file allows arbitrary blocks to be read or written. Thus, we may read block 14, then read block 53 and then write block 7. There are no restrictions on the order of reading or writing for a direct access file.

3. Other access methods : Other access methods can be built on top of a direct access method. These methods generally involve the construction of an index for the file. To find a record in the file, we first search the index, and then use pointer to access the file directly and to find the desired record.

Allocation methods : Many files will be stored on the same disk. The main problem is how to allocate space to these files so that disk space is utilized effectively and files can be accessed quickly. Following are the allocation methods :

1. Contiguous allocation : It requires each file to occupy a set of contiguous blocks on the disk. Disk addresses define a linear ordering on the disk, when head movement is needed, it is only one track. Contiguous allocation of a file is defined by the disk address and length of the first block.

2. Linked allocation : With linked allocation, each file is a linked list of disk blocks, the disk blocks may be scattered anywhere on the disk. The directory contains a pointer to the first and last block of the file.

3. Indexed allocation : Linked allocation solves the external-fragmentation and size declaration problems of contiguous allocation. Indexed allocation brings all the pointers together into one location : the index block.

Each file has its owns index block, which is an array of disk-block addresses. The ith entry in the index block points to the ith block of the file. The directory contains the address of index block. To read the i^{th} block, use the pointer in the ith index-block entry to find and read the desired block.

Q 53. Explain file protection.

Ans. A system which provides the mechanism for online storage and access to file or authentication contents like data and programs is called file system. The file system permanently resides on secondary storage holding permanently data.

Q 54. List various permissions available in AFS?

Ans. In AFS, access control lists grant permissions on a peruser and group basis. Each directory has an ACL that controls the directory and the files at contains.

There are seven permissions that may be granted, to either groups of users or individuals. Some system defined group exists, but you can before your own groups. ACLs always are applied to directories rather than to individual files. Some of the seven permission bits control access to the directory and some control access to the files within the directory.

Q 55. Discuss various methods of file allocation with advantages and disadvantages. (PTU, Dec. 2008)

OR

List and discuss various methods of file allocation. (PTU, Dec. 2010)

Ans. Following are the methods of file allocation.

1. Contiguous allocation : It requires each file to occupy a set of contiguous address on the disk. Disk addresses define a linear ordering on the disk. With this ordering, accessing block b+1 after block b normally requires no head movement. When head movement is needed, it is only one track.

Advantages :

1. Simplest method of allocation.
2. Performance is good because entire file can be read from the disk in a single operation.

Disadvantages :

1. Difficulty in finding space for a new file. If the file is to created is n blocks long, then the OS must search for n free contiguous blocks.
2. This suffer from external fragmentation.
3. When the file is created, the total amount of space it will need must be known and allocated.

2. Linked list allocation : In this, each file is a linked list of disk block. The directory contains a pointer to the first and block of the file. With this, each directory entry has a pointer to the first disk block of the file.

Advantages :

1. There is no external fragmentation.
2. No need to declare the size of a file when that file is created.

Disadvantages :

1. It is inefficient to support direct-access, it is effective only for sequential access files.
2. Another problem is reliability.

3. Indexed allocation : In this, each file has its own index block, which is an array of disk sector of addresses. The ith entry in the index block points to the ith sector of the file. The directory contains the address of the index block of a file.

Advantages :

1. It supports direct access, without suffering from external fragmentation.

Disadvantages :

1. The index block will occupy some space and thus could be considered as an overhead.

Q 56. Explain types of access.

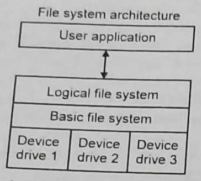
Ans. The need to protect files is a direct result of the ability to access files. Systems that do not permit access to the files of other users do not need protection. Thus, we could provide complete protection by prohibiting access.

Types of access :

1. Read : Read from the file.
2. Write : Write or rewrite the file.
3. Execute : Load the file into memory and execute it.
4. Append : Write new information at the end of file.
5. Delete : Delete the file and free its space for possible reuse.
6. List : List the name and attributes of the file.

Q 57. Explain file system architecture.

Ans. The file system provides a simple and unified way to access resources. The basic unit is file. A file consists of essential data about data, non-essential metadata, and some information. Unless the file is a directory, the information is given as is and not analyzed by the file.



Q 58. Differentiate between physical and logical file system. **(PTU, Dec. 2017)**

Ans. Physical file : A collection of byte stored on a disk or tape. When a program wants to use a particular file, "data", the operating system must find the physical file called "data".

Logical file : A channel like a telephone line that hides the details of file's location and physical format to the program. Operating system makes the hoop up by assigning a logical file to data. The logical file has a logical name which is what is used inside the program.

Q 59. Explain logical file system.

Ans. The logical file system is the level of the file system at which users can request file operations by system call. This level is the abstraction of physical view. It deals with all metadata associated with a file (UID, GID, mode, dates, etc.), *i.e.*, everything.

Q 60. What is access control list?

Ans. An access control list (ACL) with respect to a computer file system, a list of permissions attached to an object. An ACL specifies which users or system processes are granted access to objects, as well as what operations are allowed on given objects. Each entry in a typical ACL specifies a subject and an operation. The entry (ALICE, delete) on the ACI for file W×4.

Q 61. What are device drivers?

Ans. In computing, a device driver or software driver is a computer program allowing higher level computer programs to interact with a hardware device. A drive typically communicates with the device through the computer bus or communications subsystems to which the hardware connects. When a calling program invokes a routine in the driver, the driver issues commands to the device.

Q 62. Explain the different methods for allocating the space to files on disk.

Ans. A system and method allocates disk memory space for storage of compared records and enables the compressed records to be stored in sequential physical positions on the disk memory space.

Q 63. Why is protection of file required? **(PTU, Dec. 2008)**

Ans. When information is kept in computer system we want to keep it safe from physical and improper access (protection).

Reliability is generally provided (physical damage) by duplicate copies of files. Many computers have systems programs that automatically copy desk files to tape at regular intervals to maintain a copy should a file system be accidentally destroyed. File systems can be damaged by hardware

problems ; power surges as failure, head crashes, dirt, temperature extremes and vandalism. Protection can be provided by many ways. For small system protection is done by physically removing the floppy disk and locking them in a desk drawn or file cabinet.

Q 64. Compare the access lists and capabilities.

Ans.

Access List	Capabilities List
1. Each object (resource) has a list of pairs of the form <subject, access rights>	1. Each subject (user, process or procedure) has a list of pairs of the form <object, access rights>.
2. It would be tedious to have separate listing for each subject (user), therefore, they are grouped into classes. For example, in UNIX, there are three classes : self, group, anybody else.	2. Here capabilities are the names of the objects. The objects not referred to in a capability list cannot be even named.
3. The default is : Everyone should be able to access a file.	3. The default is : No one should be able to access a file unless they have been given a capability.
4. Access lists are simple and are used in almost all file systems.	4. Capabilities are used in systems that need to be very secure as it prohibits sharing of information unless access is given to a subject.

Q 65. What is file system? Explain file protection and allocation methods.

(PTU, May 2018, 2009)

Ans. File system : A system which provides the mechanism for on line storage and access to file contents like data and programs is called file system. The file system permanently resides on secondary storage holding permanent data.

File Protection : When information is kept in computer system we want to keep it safe from physical damage and improper access. Reliability is generally provided by duplicate copies of files. Many computer have system programs that automatically copy disk files to take at regular intervals to maintain a copy should a file system be accidently destroyed. File system can be damaged by hardware problem ; power surges as failure, head crashes, dirt, temperature extremes and vandalism. Files my be deleted accidently.

Protection can be provided in many ways. For a small single uses system we might provide protection by physical removing the floppy disk and locking them in a desk drawer or file cabinet. Multi user system provide controlled access by limiting the tapes of file access that can be made. Access is permitted or devied depending on several factors one of which is the type of access requested.

Allocation methods : Many files will be stored on the same disk. The main problem is how to allocate space to these files so that disk space is utilized effectively and files can be accessed quickly. Following are the allocation methods :

1. Contiguous allocation : It requires each file to occupy a set of contiguous blocks on the disk. Disk addresses define a linear ordering on the disk, when head movement is needed, it is only one track. Contiguous allocation of a file is defined by the disk address and length of the first block.

2. Linked allocation : With linked allocation, each file is a linked list of disk blocks, the disk

blocks may be scattered anywhere on the disk. The directory contains a pointer to the first and last block of the file.

3. Indexed allocation : Linked allocation solves the external-fragmentation and size declaration problems of contiguous allocation. Indexed allocation brings all the pointers together into one location : the index block.

Each file has its owns index block, which is an array of disk-block addresses. The ith entry in the index block points to the ith block of the file. The directory contains the address of index block. To read the i^{th} block, use the pointer in the ith index-block entry to find and read the desired block.

Q 66. Consider a system that supports the strategies of contiguous, linked and indexed allocation. What criteria should be used in deciding which strategy is best utilized for a particular file? **(PTU, Dec. 2011)**

Ans. 1. Contiguous : If file is usually accessed sequentially, if file is relatively small.

2. Linked : If file is large and usually accessed sequentially.

3. Indexed : If file is large and usually accessed randomly.

To determine best strategy for a particular file, the two criteria's are used which are :

1. Storage efficiency
2. Data-block access time.

Storage efficiency should be more and access time should be less.

Q 67. What is device driver?

Ans. A device driver is a particular form of software application that is designed to enable interaction with hardware devices. Without the required device driver, the corresponding hardware device fails to work.

A device driver usually communicates with the hardware by means of the communications subsystem or computer bus to which the hardware is connected. Device drivers are operating system specific and hardware dependent. A device driver acts as a translator between the hardware device and the programs or operating systems that use it. A device driver may also be called a software driver.

Q 68. What is the function of device independent I/O software ?

Ans. The basic function of the device independent software is to perform the I/O functions that are common to all devices and to provide a uniform interface to the user-level software. Though it is difficult to write completely device independent software but we can write some modulus which are common among all the devices. Following is the list of functions of device independent I/O software :

- Uniform interfacing for device drivers.
- Device naming-Mnemonic names mapped to major and minor device numbers.
- Device protection
- Providing a device-independent block size.
- Buffering because data coming off a device cannot be stored in final destination.
- Storage allocation on block devices.
- Allocation and releasing dedicated devices.
- Error reporting.

Q 69. What are various types of files used in OS ?

Ans. File types refers to the ability of the operating system to distinguish different types of file s text files source files and binary files etc. Many operating systems support many types of operating system like MS–DOS and UNIX has the following types of files :

1. Ordinary files : These are the files that contain user information. These may have text, database or executable program. The user can apply various operations on such files like add, modify, delete or even remove, the entire file.

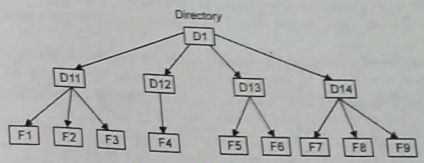
2. Directory files : These files contain list of filenames and other information related to these files.

3. Special files : These files are also known as device files. These files represent physical device like disks, terminals, printers, networks, tape drive etc. These files are of two types :

- **Character special files :** data is handled character by character as in case of terminals or printers.
- **Block special files :** data is handled in blocks as in the case of disks and tapes.

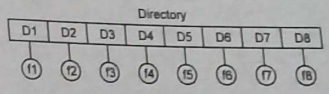
Q 70. Explain structures of directory in operating system.

Ans. A directory is a container that is used to contain folders and file. It organizes files and folders into a hierarchical manner.



There are several logical structures of adirectory which are as follows :

1. Single level directory Single level directory is simplest directory structure. In it all files are contained in same directory which make it easy to support and understand.



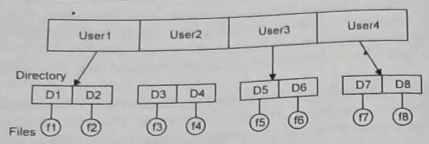
Advantages :

- Since it is a single directory, so its implementation is very easy.
- If the files are smaller in size, searching will become faster.
- The operations like file creation, searching, deletion, updating are very easy in such a directory structure.

Disadvantages :

- There may chance of name collision because two files can not have the same name.
- Searching will become time taking it the directory is large.
- This can not group the same types of files together.

2. Two Level directory : As we have seen, a single level directory often leads to confusion of file names among different users. The solution to this problem is to create a separate directory for each user. In the two level directory structure, each user has there own user files directory (UFD). The UFDs has similar structure, but each lists only the files of a single user. System's master file directory (MFD) is searched whenever a new user id = S logged in. The MFD is indexed by username or account number, and each entry points to the UFD for that user.



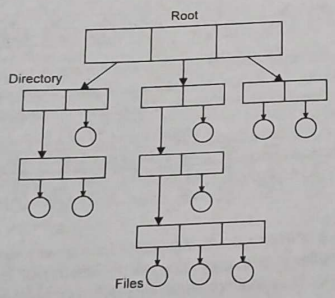
Advantages :

- We can give full path like/user-name/directory name
- Different users can have same directory as wall as file name.
- Searching of files become more easy due to path name and user grouping.

Disadvantages :

- A user is not allowed to share files with other users.
- Still it not very scalable, two files of the same types cannot grouped together in the same user.

3. Tree structured directory : Once we have seen a two level directory as a tree of height 2, the natural generalization is to extend the directory structure ta a tree of arbitrary height. This generalization allows the user to create their own subdirectories and to organize on their files accordingly.



A tree structure is the most common directory structure. The tree has a root directory and every file in the system have a unique path.

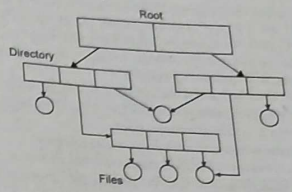
Advantages :

- *Very generalize, since free path name can be given.*
- *Very scalable, the probability of name collision is less.*
- *Searching becomes very easy. We can use both absolute path as well as relative.*

Disadvantages :

- *Every file does not fit into the hierarchical model, files may be saved into multiple directories.*
- *We can not share files.*
- *It is inefficient, because accessing a file may go under multiple directions.*

4. Acyclic graph directory : An acyclic graph is a graph with no cycle and allows to share sub directories and files. The same file or sub directories may be in two different directories. It is a natural generalization of the tree-structured directory.



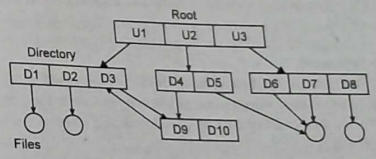
Advantages :

- We can share files.
- Searching is easy due to different–different paths.

Disadvantages :

- We share the file via linking, in case of delecting it may create the problems.
- If the link is softlink then after deleting the file we left with a dangling pointer.
- In case of hardlink, to delect a file we have to delete all the reference, associated with it.

5. General graph directory structure : In general graph directory structure, cycles are allowed within a directory structure where multiple directories can be derived from more than one parent directory.



Advantages :

- It allows cycles.
- It is more flexible than other directories structure.

Disadvantages :

- It is more costly than others.
- It needs garbage collection.

Q 71. What do you mean by disk formatting ?

Ans. Disk formating is the configuring process of a data storage media such as a hard disk drive, floopy disk or flash drive for initial use. Any existing files on the drive would be erased with disk formatting. Disk formatting is usually done before initial installation or before installation of a new operating system. Disk formatting is also done. If there is a requirement for additional storage in the computer.

Q 72. What is boot block in operating system.

Ans. Basically for a computer to start running to get an instance when it is powered up or rebooted it need to have an initial program to run. And thus initial program which is known as bootstrap need to be simple. It must initialize all aspects of the system, from CPU registers to device controllers and the contents of the main memory and then starts the operating system. To do this job the bootstrap program basically finds the operating system kernel on disk and then loads the kernel into memory and after thus it jumps to the initial address to begin the operating system execution.

Q 73. What is bad block in operating system ?

Ans. A bad block is an area of storage media that is no longer reliable for storing and retrieving data because it has been physically damaged or corrupted. A soft or logical bad block occurs when the operating system is unable to read data from a sector.

Q 74. Explain free space management in operating system.

Ans. The system keeps tracks of the free disk blocks for allocating space to files when they are created. Also, to reuse the space released from deteting the files, free space management becomes crucial. The system maintains a free space list which keeps track of the disk blocks that are not alloacted to some file or directory. The free space list can be implemented mainly as :

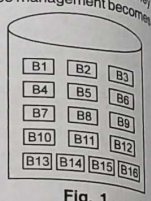


Fig. 1

1. Bitmap or Bit Vector : A Bitmap or Bit vector is series or collection of bits where each bit corresponponds to a disk block. The bit can take values 0 and 1. 0 indicates that the block is allocated and 1 indicates a free block. The given instance of disk blocks the disk in the fig. 1 can be represented by a bitmap of 16 bits as 0000111000000110.

Advantages :

1. Simple to understand.
2. Finding the first free blocks is efficient. It requires scanning, the words in a bitmap for a non zero word. The first free block is then found by scanning for the first 1bit in the non-zero word. The block number can be calculated as (number of bits per word) * (number of 0 values words) + offset of bit first bit 1 in the non-zero word.

For the fig. 1 we scan the bitmap sequentially for the first non zero word. The first groups of 8 bits (00001110) constitute a non zero words since all bits are not 0 After the non 0 words is found, we look for the first 1 bits. This is the 5th bit of the non-zero words. So offset = 5. Therefore, the first free block number = $8 \times 0 + 5 = 5$.

2. Linked List : In this approach, the free disk blocks are linked together i.e a free block contains a pointer to the next free block. The block number of the very first disk block is stored at a separate location on disk and is also cached in memory. In fig. 2, the free space list head points to block 5 which points to Block 6, the next free block and so on.

The last free block would contain a null pointer indicating the end of free list. A draw back of this method is the I/O required for free space list traversal.

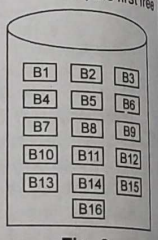


Fig. 2

3. Grouping : This approach stores the address of the free blocks in the first free block. The first free block stores the address of some, say n free blocks. Out of these n blocks, the first n–1 blocks are actually free and the last block contains the address of the next free n blocks.

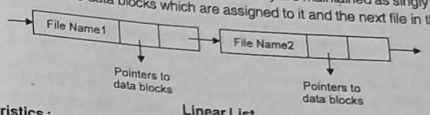
4. Counting : This approach stores the address of the first free disk blocks and a number n of free contiguous disk blocks that follow the first block. Every entry in the list would contain :

1. Address of first free disk block
2. A number n.

Q 75. Explain directory implementation algorithms in detail.

Ans. There is the number of algorithms by using which, the directories can be implemented However, the selection of an appropriate directory implementation algorithm may significantly affect the performance of the system. The directory implementation algorithms are classified according to the data structure they are using. There are mainly two algorithms which are used in these days.

1. Linear List : In this algorithm, all the files in a directory are maintained as singly linked list. Each file contains the pointers to the data blocks which are assigned to it and the next file in the directory.

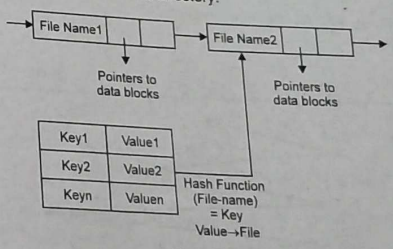


Linear List

Characteristics :

1. When a new file is created, then the entire list is checked. Whether the new file name is matching to a existing file name or not. In case, it doesn't exist, the file can be created at the beginning or at the end. Therefore, searching for the unique name is a big concern because traversing the whole list takes time.
2. The list needs to be traversed in case of every operation (creation, deletion, updating etc.) on the files therefore the systems becomes inefficient.

2. Hash Table : To overcome the drawbacks of single linked list implementation of directories, there is an alternative approach that is hash table. This approach suggests to use hash table along with the linked lists. A key value pair for each file in the directory gets generated and stored in the hash table. The key can be determined by applying the hash function on the file name while the key points to the corresponding file stored in the directory.



Hash Table

Q 76. How we can improve efficiency and performance in operating system ?

Ans. A variety of techniques are used to improve the efficiency and performance of secondary storage :

1. Efficiency : The efficient use of disk space depends heavily on the disk allocation and directory algorithm in use. For instance, UNIX inodes are preallocated on a volume. Even an "empty" disk has a percentage of its space lost to inodes. However by preallocating the inodes and spreading them across the volume, we improve the file system's performance. This improved performance results from the UNIX allocation and free space algorithms which try to keep a file's data blocks near that file's inode black to reduce seek time.

2. Performance : Even after the basic file system algorithms have been selected, we can still improve performance in several ways. Most disk controllers include local memory to form an on board cache that is large enough to store entire tracks at a time. Once a seek is performed, the track is read into the disk cache starting at the sector under the disk head. The disk controller then transfers any sector requests to the OS. Once blocks make it from the disk controller into main memory the operating system may cache the blocks there. Some systems maintain a separate section of main memory for a buffer cache, where blocks are kept under the assumption that they will be used again shortly. Other systems cache file data using a page cache. The page cache uses virtual memory techniques to cache files data as pages rather than as file system oriented blocks. Caching file data using virtual addresses is far more efficient than caching through physical disk blocks accesses interface with virtual memory rather than the file system. This is known as virtual memory.

Q 77. Suppose that the head of moving head-disk with 200 tracks, numbered 0 to 199, has just finished a request at track 125. The queue of the requests is kept in FIFO order :

86, 147, 91, 177, 94, 150, 102, 175, 130.

What is the total number of head movements needed to satisfy requests for the following disk scheduling algorithms :

(a) FCFS

(b) SSTF

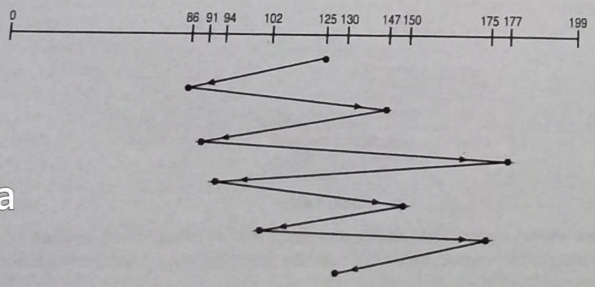
(c) Scan.

(PTU, May 2010)

Ans. (a) FCFS : Head-disk is initially at track 125. It will first move to 86 track from 125, then to 147, 91, 177, 94. 150, 102, 175 and finally to 130. For a total head movements of 547 tracks. This is diagrammed as :

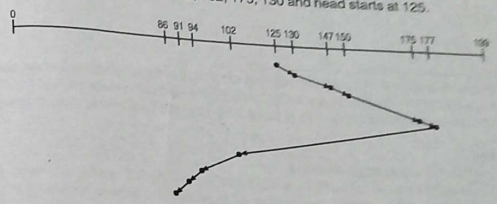
queue = 86, 147, 91, 177, 94, 150, 102, 175, 130

head start at 125



(b) SSTF : Initially head disk at 125th track the closet request to initial track is 130 track, the next request is at track 147. Then to 150, 175, 177, 102, 94, 91, 86. For a total head movements of 143 tracks. This is diagrammed as :

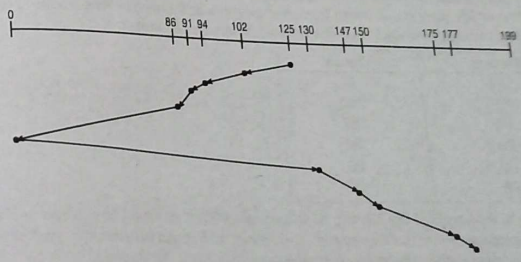
Queue = 86, 147, 91, 177, 94, 150, 102, 175, 130 and head starts at 125.



(c) Scan : In this scheduling we need to know the direction of head movements, in addition to the head current position 125 track. If the disk head is moving toward 0, the head will move toward the other end of the disk, servicing the request at 130, 147, 150, 175 and 177 track. It is diagrammed as :

Queue : 86, 147, 91, 177, 94, 150, 102, 175, 130

Head starts at 125



In this total head movements are 302 tracks.

Q 78. Compare various disk scheduling algorithms by taking suitabale example.

(PTU, Dec. 2011)

Ans. Following are disk scheduling algorithms

1. First Come First Server : In this scheduling we will process the request that arrive first then move to process that arrive later. This is the simplest disk scheduling technique. This technique is fair enough to process the request. This technique does not provide fastest service to process the request that arrive to the CPU. For Example

Process	Block
P1	92
P2	152
P3	82
P4	30
P5	20
P6	50

According to this example the starting position of head is at 20. During the first process head will move to block 92 then it will move to block 182 this is the forward movement of head then it will go back to block 81 this is backward movement of head. All processes processing required 242 blocks to be traveled.

Advantages :

(a) It is the simple scheduling technique
(b) It will follow the order of request. It means that First In First Out will be followed up
(c) As the request will be placed in queue that will be processed without observing the higher order or lowest order.
(d) This technique can be implemented easily.

Disadvantages :

(a) It does not provide fastest service to us
(b) Reading and Writing time to the disk will take more time. Because switching from one block from one block.

2. Shortest Seek Time First: Shortest Seek Time First is a scheduling in which job will be processed first which contains less time from the current head location. This scheduling technique will ensure that process which will take less time will be processed first.

Process	Block
P1	92
P2	152
P3	82
P4	30
P5	20
P6	50

According to the example the starting head location is 80. It will check which one process will take less time to process. From starting position it will move to process P3 after that it will check that which block is very near to the current block location.

Advantages :

(a) Shortest Time taking process will be executed first then other process will be processed.
(b) It will not consider the order of insert.
(c) It is the best technique when we have short time process in the queue.

Disadvantages :

(a) This technique will take us to starvation condition. For example
we are at block no. 20 and next block location is 220.
According to this technique process at block location 220 will remain in wait condition .

3. SCAN Scheduling : In this scheduling technique we will start from one block then we will

move to another block. In other word we can say that head will start moving from one block to another block the request that arrive between starting to ending location of memory that will be processed? In this scheduling the process will executed in a specific order.

For Example

Process	Block
P1	56
P2	78
P3	18
P4	120
P5	220
P6	250

The Processing will perform in the following order.

First of all starting position of the head will be considered . Then first location P1 will be processed then P2,P4,P5,P6 in the last when head will be restarted the process no. P3 will be processed.

Advantages :

(a) It is the simple disk scheduling technique
(b) It is faster technique than FCFS,SSTF technique.
(c) It is Better technique to avoid starvation of process.

Disadvantages :

(a) if any process arrive before the head position then process will have to wait until the head position reach to that location.
(b) It is not best when you want to execute a process immediately.

4. C-SCAN Scheduling : C-SCAN scheduling in which head will move in circular condition. that means when head reach to end of the disk then it will be started automatically to the starting of the disk.

For Example :

Process	Block
P1	56
P2	78
P3	88
P4	120
P5	220
P6	250

According to this example there are six processes from P1 to P6 and head position start from block no. 109 then it will process in the following order

Process number	Process Name
1	P4
2	P5
3	P6
4	P1
5	P2
6	P3

This example indicate the process will be started with P4 and ending with P3.

Advantages :

(a) This scheduling with ensure every request will be processed
(b) This is the simple technique to process different types of request
(c) The request will be processed in a particular sequence

Disadvantages :

(a) The process request before the head position will have to wait for their term
(b) Waiting for execution can create starvation condition for different types of processes.

5. LOOK Scheduling : In This scheduling head will move in any direction according to request done by the user. In this scheduling head will move in any direction it can be forward or backward.

For Example

Process	Block
P1	92
P2	152
P3	82
P4	30
P5	20
P6	50

According to the example user make a request for a particular block then head will be automatically shifted to that block.

Advantages :

(a) request will be processed fast
(b) Priory can be given according to the user request
(c) This is the simple techniques for scheduling

Disadvantages :

(a) Unrequired blocks can be read by the head because moving from one location to another location that will travel lot of blocks from one location to another location.

Q 79. Directory implementation. **(PTU, Dec. 2014)**

Ans. The simplest form of directory implementation is linear list, if has a linear list of file names with pointer to the data blocks. To add a new file first a full checking of the linear list is made so that there would not be any duplicate entries If there is no duplicate entry a new entry is made at the end of the directory. Deletion is simple, we search the directory and release the space associated with it. To reuse a directory entry we can mark the entry as unused or we can attach it to the list of free directories.

The real disadvantage of linear list is that its operations are time consuming. It because we have to traverse the entire list to find an entry. A stored list reduces this problem but sorting it self is a problem and the list are constantly updated, so sorting will be needed all ways.

Hash table - linear list with hash data structure

- Decrease directory search time
- Collisions - situations where two file names hash to the same location.
- Fixed size (disadvantage : hash function is dependent on fixed size).

Q 80. What is meant by load control ? Explain various methods of controlling load? **(PTU, Dec. 2014)**

Ans. Load control is concerned with determining the number of processes that will be resident

in main memory, which has been referred to as the multiprogramming level. The load control, policy is critical in effective memory management. If too few processes are resident at any one time, then there will be many occasions when all processes are blocked, and much time will be spent in swapping. On the other hand, if too many processes are resident, then, on average, the size of the resident set of each process will be inadequate and frequent faulting will occur. The result is thrashing.

The phenomena of thrashing was the necessity for better load control, i.e. the operating system has to monitor the paging rate and reduce the level of multiprogramming if the paging rate is too high. Virtual memory is not magic, and you can not run as many programs as you want simultaneously or else thrashing will occur. The earlier systems had a form of load control but it was based on CPU utilization, which is the wrong measure. Load control must be based on the paging rate. The idea of load control is quite simple. You monitor the page fault rate be incrementing a counter every time there is a page fault. Every second or so, you check the page fault rate to see if it is over some threshold. If it is, you remove on process from memory, i.e. you swap it out and give all its page frames to other processes. A new process can be started or a swapped-out process can be swapped in only if the page fault rate is low enough.

Q 81. Explain various device managements policies ? **(PTU, Dec. 2018, 2014)**

Ans. Device manager divides task into 3 parts, with each handled by specific software component of I/O subsystem

I/O traffic controller watches status of all devices, control units, and channels.

Monitors status of every device, control unit, and channel ; Becomes more complex as number of units in I/O subsystem increases and as number of paths between these units increases.

Three main tasks :

1. It must determine if there's at least 1 path available.
2. If there's more than 1 path available, it must determine which to select; and.
3. If paths are all busy, it must determine when one will become available.

Maintain a database containing status and connections for each unit in I/O subsystem, grouped into channel control blocks, control unit control blocks, and Device control blocks.

Traffic controller maintains Database for Each unit in I/O subsystem.

Channel Control Block	Control unit Control Block	Device Control Block
❑ Channel identification status	❑ Control unit identification status	❑ Device identification status
❑ List of control units connected to it	❑ List of channels connected to it	❑ List of control units connected to it.
❑ List of processes waiting for it	❑ List of devices connected to it ❑ List of processes waiting for it	❑ List of processes waiting for it.

I/O Scheduler : I/O Scheduler implements policies that determine allocation of, and access to, devices, control units, and channels. I/O scheduler performs same job as process scheduler ; it allocates the devices, control units, and channels. Under heavy loads, when # requests > # available paths, I/O Scheduler must decide which request satisfied first. I/O requests are not preempted : Once channel program has started, it's allowed to continue to completion even though I/O requests with higher priorities may have entered queue.

Feasible because programs are relatively short (50 to 100 ms.)

Some systems allow I/O scheduler to given preferential treatment to I/O request from "high-priority" programs. If a process has high priority then its I/O requests also has high priority and is

satisfied before other I/O requests with lower priorities.

I/O scheduler must synchronize its work with traffic controller to make sure that a path is available to satisfy selected I/O requests.

I/O device handler : I/O device handler processes the I/O interrupts, handles error conditions, and provides detailed scheduling algorithms, which are extremely device dependent.

Each type of I/O device has own device handler algorithm.

- first come first served (FCFS)
- shortest seek time first (SSTF)
- SCAN (including Look, N-step SCAN, C-SCAN, and C-Look)

Every scheduling algorithm should :

- Minimize arm movement.
- Minimize mean response time.
- Minimize variance in response time.

Q 82. Discuss the functions of I/O Traffic Controller. **(PTU, May 2018 ; Dec. 2015)**

Ans. The I/O traffic controller is concerned with the mechanics and not policies. The traffic controller maintains the status information of a device.

Functions of I/O traffic controller.

1. It must determine if there is atleast one path available.
2. If more than one path is available then it must determine which one to select.
3. If all paths are busy, it must determine when one will become available.

To accomplish these three functions, I/O traffic controller maintains a database that stores the status and connection between channels CUs etc. All this is achieved by means of various control blocks like Unit Control Blocks (UCB), Control Unit Control Block (CUCB), and Channel Control Block (CCB).

Device unit identification
Status of the device
List of CUs connected to this device
List of processes waiting for this device

Unit Control Block (UCB)

The first function is achieved by starting at the desired UCB and working back through the connected CUs and channels to find a combination that is available. In similar fashion, it is possible to determine all available paths. The second function is important when I/O configuration is asymmetric. This is because in this configuration, selection of one path may block out other I/O requests.

Q 83. What is disk formatting and what basic information is stored with each disk sector? **(PTU, May 2016)**

Ans. Disk formating : It is the process of preparing a hard disk or other storage medium for use, including setting up an empty file system. A disk has eight or more disk sectors per track. Each sector is uniquely assigned a disk address before a disk drive can access a piece of data. The disk address comprised sector number, track number and surface number.

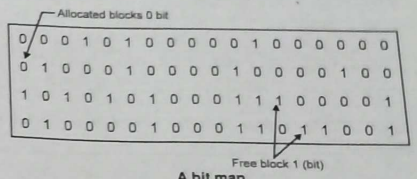
Q 84. What is disk scheduling ? When is it used ? **(PTU, Dec. 2016)**

Ans. Disk scheduling is done by operating system to schedule I/O requests arriving for disk. Disk scheduling is also known as input/output scheduling. Disk scheduling is important because input/output requests may arrive by different processes and only one I/O request can be served at a time by disk controller. Thus other I/O requests need to wait in waiting queue and need to be scheduled.

Disk scheduling is used to reduce disk head movements and the average wait time for I/O operations.

Q 85. Describe bit map approach in free space managememt. Use block diagrams. List advantages and disadvantages. **(PTU, May 2016)**

Ans. Bit map is also known as bit vector. It is widely used to keep track of the free blocks on a disk. To track all the free and used blocks on a disk with total n blocks, a bit map having n bits is required.



A bit map

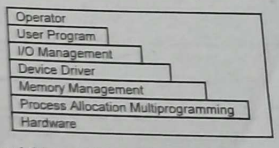
- The main advantage of this method is its relatively simple and n contiguous free blocks can be found together.
- A disadvantage of bit map is that the file system may need to search the entire bitmap to find a free block, resulting in execution overhead.

Q 86. Define the term Disk Bandwidth. **(PTU, May 2018 ; Dec. 2016)**

Ans. Disk Bandwidth : The disk bandwidth is the total number of bytes transferred divided by the total time between the first request for service and the completion of the last transfer. We can improve the bandwidth by scheduling the servicing of disk input/output requests in a good order.

Q 87. Explain the Layered architecture in detail. **(PTU, May 2019)**

Ans. Layered architecture : The layered architecture of operating system was developed in 60's. In this, the O/S is broken up into number of layers or we can also say that a layered operating system architecture separate system functionality into hierarchical layers.



The main disadvantage of this architecture is that it requires an appropriate definition of the various layers and a careful planning of the proper placement of the layer.

❑❑❑