(1)

07/4/21

Assignment-1

Operating System

Menaaz Fatima
19D01A0580
CSE-B

1. Define operating systems? Explain its important functions.

Ans: Operating Systems: An operating system is a program that controls the execution of application program and acts as an interface between the user of a computer and the computer hardware.

Operating system is the one program running at all times on the computer with all else being application program.

Functions of Operating System:

* Convinience: An OS makes a computer more convinient to use.

* Efficiency: An OS allows the computer system resources to be used in an efficient manner.

Ability to Evolve: An OS should be constructed in such a way as to permit the effective development, testing and introducing of new system functions.

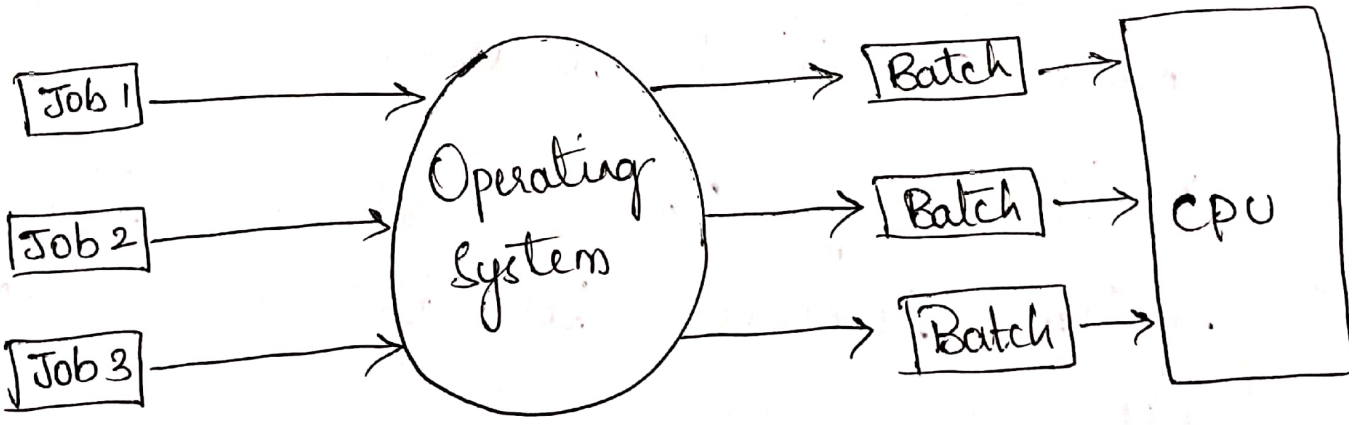
2. What are the different types of operating systems?

Ans: Types of operating systems are as follows:

1. Batch Operating System
2. Time sharing operating System
3. Distributed Operating System
4. Network Operating System.

3. Explain the following with diagram and examples. a). Simple Batch b). multiprogrammed OS

Ans: Simple Batch:



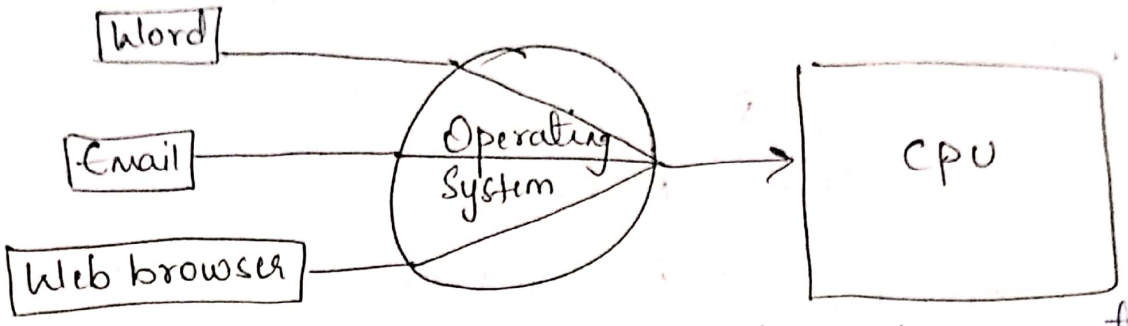
This type of operating system does not interact with the computer directly. There is an operator which takes similar jobs having the same requirement and group them into batches. It is the responsibility of the operator to sort jobs with similar needs.

Example: Payroll System, Bank Statements.

b) Multiprogrammed OS: In MPOS, the operating system reads jobs from disk device where a list of jobs are already being stored through card readers. The OS then pull and store as much as job it can in the memory. Then from the memory, OS start working on a job. Now, whenever a job reaches a situation where it has to be waiting for one or more tasks to be completed like use of any I/O devices, the OS pulls another job from the memory & starts working on it. Whenever this job also starts waiting, the OS pull another job.

4. Explain the following with diagram & examples. a). time shared os b). parallel os

Ans: Time Shared OS:



Each task is given some time to execute so that all the tasks work smoothly. Each user gets the time of cpu as they use a single system. These systems are also known as multitasking systems. The task can be from single user or different user. Time that each task gets to execute is called quantum. After this time interval is over os switches over to new task.

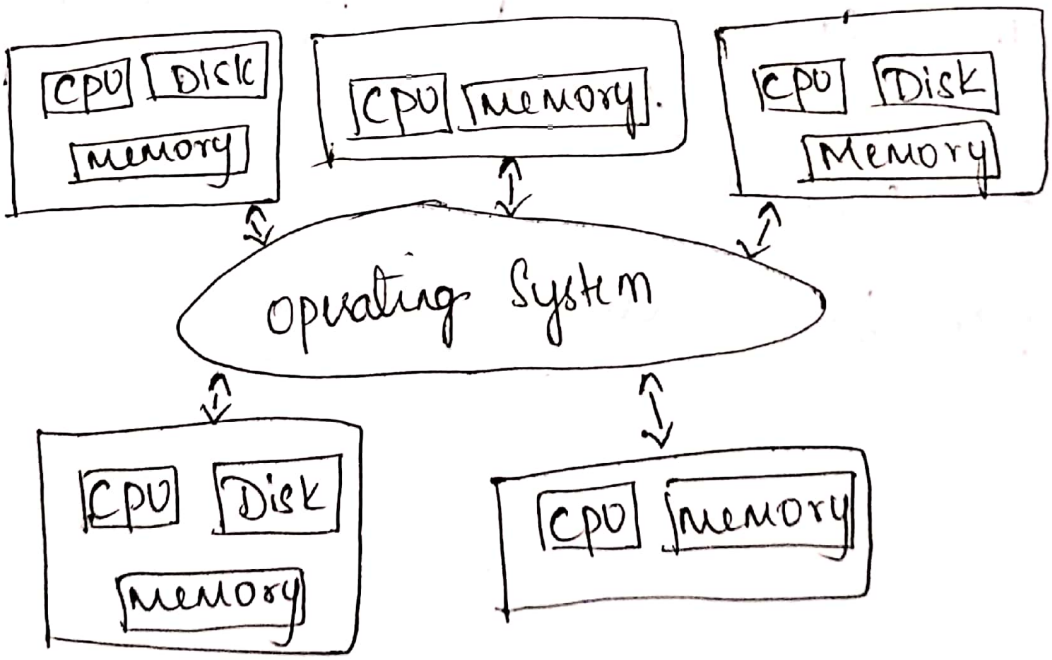
Example: Multics, Unix.

b). Parallel OS: Parallel os are supposed to be used in machines more than one processor connected closely where these multiple processors will be sharing memory, clock

buses and other peripherals. Parallel systems are designed to distribute jobs to multiple processors to have faster speed. But the speed-ratio is for a processor is not in the times for a single processor as, when multiple processors co-operate on a task, a certain amount of overhead is incurred in keeping all the parts working correctly. Different parallel systems exists where each processor runs a copy of operating system.

5. Explain the following with diagram and examples: a). distributed os b) real time os.

Ans: Distributed OS:



It is the recent advancement in the world of computer technology and are being widely accepted all over the world and, with a great pace. Various autonomous and interconnected computers communicate with each other using a shared communication network. Independent systems possess their own memory unit & CPU. These are referred as loosely coupled systems or distributed systems. These system's processors differ in size and function. The major benefit of working with these types of OS is that it is always possible that one user can access that files or software which are not actually present on his system but some other system connected within this network remote access is enabled within the device connected.

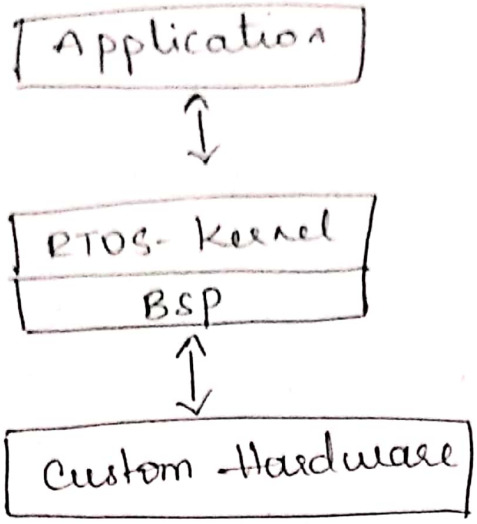
Examples: Locus, etc.

(ii)

b) Real time OS: This type of OS is used in real-time systems. The time interval required to process and respond to inputs is very small. This time interval is called response time. Real time systems are used when there are time requirements that are very strict like missile systems, air traffic control systems, robots, etc. There are two types of real-time operating system.

1. Hard Real Time OS: These OS are used for applications where time constraints are very strictly and even the shortest possible delay is not acceptible. These systems are built for saving life like automatic parachutes or airbags which are required to be readibly available in case of any accident. Virtual memory is rarely found in these systems.

2. Soft Real Time OS: These OS are for applications where for time-constraint is less strict.



6). Explain about various operating system components.

Ans: 1. File Management: A file is a set of related informa^ which is should define by its creator. It commonly represents programs, both source and obje forms and data. Data files can be numeric, alphabetic or alphanumeric.

Functions:

* File & dir crea^ & dele^
* For manipulating files and directories.
* Mapping files onto secondary storage
* Backup files on stable storage media.

2. Process Management: The process management component is a procedure for managing the many processes that are running simultaneously on operating system. Every software applicaⁿ program has one or more processor associated with them when they are running the execution of a process must be sequential so at least one instruction should be executed on behalf of process.

Functions:

* Process creaⁿ & deleⁿ
* Suspension & resumpⁿ
* Synchronisaⁿ process
* Communicaⁿ process.

3. I/O Device Management: one of the important use of an operating system that helps you to hide the variaⁿ of specific hardware devices from the user.

Functions:

* It offers buffer caching system
* Provides general device driver code
* Provides drivers for particular hardware
* Helps you to know idiosyncrasies of specific device.

4. Network Management: is the process of administering & managing computer networks. It includes performance management, fault analytics, provisioning of networks & maintaining quality of service. The computers in the network are connected through a communica^ network, which can be configured in a number of different ways. With the help of network management, the network can be fully or partially connected.

Functions:

* Distributed systems help you to various computing resources in size & func^n.
* A distributed system also offers the user access to the various resources the network shares.
* It helps to access shared resources that help computers to speed-up availability.

5. Main Memory Management: Main memory has a large area of storage or bytes, which has an address. The main memory management process is conducted by using a sequence of reads or writes of specific memory address.

In order for execute a program, it should be mapped to absolute addresses & loaded inside the memory. The selection of a memory management method depends on several factors

Functions:

* It helps you to keep track of primary memory.
* Determine what part of it are in use by when, what part is not in use.
* Allocates the memory when a process requests.
* It also de-allocates the memory when a process no longer requires or has been terminated.

6. Secondary Storage Management: This memory of the computer is very small to store all data & programs permanently. The computer system offers secondary storage to back up the main memory. Today modern computers use hard drives/SSD as the primary storage of both programs and data. However the secondary storage management also works with storage devices, like a USB flash drive & CD/DVD drive.

Functions:

* Storage alloca^
* Free space management
* Disk Scheduling

7. Security Management: the various process in an operating system need to be secured from each other's activities. For that purpose, various mechanisms can be used to ensure that those processes which want to operate files, memory CPU, & other hardware resources should have proper authorization from operating system

7. Explain about various operating system servi

Ans: 1. Program Execu^: Operating system handl many kinds of activities from user program to system programs like printer spooler, name server, file server etc. A process inclu the complete execution context. The major activities are.

* Loads a program into memory.
* Executes the program.
* Handles program's execu^.
* Provides a mechanism for process communica

I/O Opera^: An I/O subsystem comprises of I/O devices & their corresponding driver software. Drivers hide the peculiarities of specific hardware devices from users. An OS manages the communica' btw user & device drivers. I/O operation means read or write operation with any file or any specific I/O device.

File system manipul^: A file represents a collec^ of related information. Computers can store files on the disk for long term storag purpose. A file system is normally organise into directories for easy navig^ & usage. These directories may contain files & other directories.

Communication: In case of distributed syste which are collec^ of processors that do not share memory peripheral devices or a clock, the operating system manages communic^s btw all the process. Two process ofen require data to be transfered blw them. Communic^ ma be implemented by two methods either by shared memory or by message passing.

Error Handling: Error can occur anytime & anywhere. An error may occur in cpu, in I/O devices in the memory hardware. The OS constantly checks for possible errors. The OS takes an appropriate action to ensure correct.

Resource Management: In case of multi-user resources such as main memory, cpu cycles & files storage are to be allocated to each user or job. The OS manages all kinds of resources using schedulers. cpu scheduling algorithms are used for better utilizn.

Protection: Protection refers to a mechanism or a way to control the access of progra processes or users to the resources defin by a computer system. The OS ensures tha all access to a system resources is contro The OS ensures that external I/O devices are protected from invalid access attempts.

8. Explain about various types of system call in operating system.

(8)

Ans: Process Control: These system calls deal with processes such as process creat^, process termin^ etc.

File Management: These system calls are responsible for file manipul^ such as creating file, reading file, writing into a file etc.

Device Management: These system calls are responsible for device manipul^ such as reading from device buffers, writing into device buffers etc.

Information Maintainance: These system calls handle inform^ & its transfer blw the operating system & the user prog

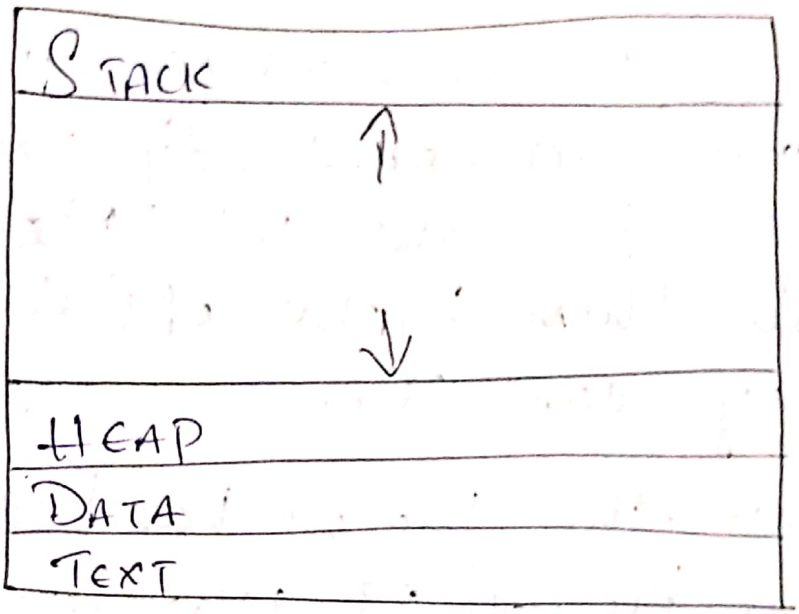
Communication: These system calls are useful for interprocess communic^. The also deal with creating & deleting a communic^ connec^.

ASSIGNMENT-2
OPERATING SYSTEM

MONAA% FATIMA
19B01A0580
CSC-B

1. Define process? Explain about process life cycle with a neat diagram.

Ans: A process is defined as an entity which represents the basic unit of work to be implemented in the system.



Stack:- The process stack contains the temporary data such as method, func? return address & local variables.

Heap:- This is dynamically allocated memo to a process during its run time.

Data:- This section contains the global & static variables.

Text: this includes the current activity represented by the value of program control & the contents of processor registers.

~~2. Explain about process control block (PCB).~~

Ans:

Process Life Cycle:

Start: this is the initial state when a process is first started/created.

Ready: this process is waiting to be assigned to a processor. Ready processors are waiting to have the processor allocated to them by the operating system. so that they can run.

Running: Once the process has been assigned to a processor by the OS scheduler, the process state is set to running & the processor executes its instructions.

Waiting: Process moves into the waiting state if it needs to wait for a resource, such as waiting for user input.

2. Explain about process control block PCB.

Ans: A PCB is a data structure maintained by OS for every process. The PCB is identified by an integer process ID (PID). A PCB keeps all the information needed to keep track of a process.

Process State: The current state of the process i.e. whether it is ready, running waiting or whatever.

Process Privileges: This is required to allow / disallow access to system resources

Process ID: Unique identific^n for each of the process in the OS.

Pointer: A pointer to parent process.

Program Counter: PC is a pointer to the address of the next instruc^n to be executed for the process.

CPU Registers: Various CPU register where process need to be stored for execu^n for running state.

CPU Scheduling Inform^n: Process priority & other scheduling inform^n which is req. to schedule the process.

Memory Management Inf^m: It includes the inform^n of page table, memory limits, Segment table depending on memory used.

Accounting Inform^n: the amount of CPU used for process execu^n, time limits, execu^n ID etc.

IO status Inform^n: Includes a list of IO devices allocated to the process.

3. Explain about process scheduling queues?

Ans: The OS maintains all PCBs in process scheduling queue. The OS maintains a separate queue for each of the process states & PCBs of all processes in the same execu^n state are placed in same queue. When the state or process is changed.

Job Queue: This queue keeps all the processes in the system.

Ready Queue: This queue keeps a set of all process residing in main memory, ready & waiting to execute.

Device Queue: The processes which are blocked due to unavailability of an I/O device constitute this queue.

4. What is scheduler? Explain about types of schedulers.

Ans: Schedulers are special software which handle process scheduling in various ways. Their main task is to select the jobs to be submitted into the system & to decide which process to run.

Long term scheduler: It is also called a job scheduler. It determines which programs are admitted to the system for processing. It selects process from the queue & loads them into memory.

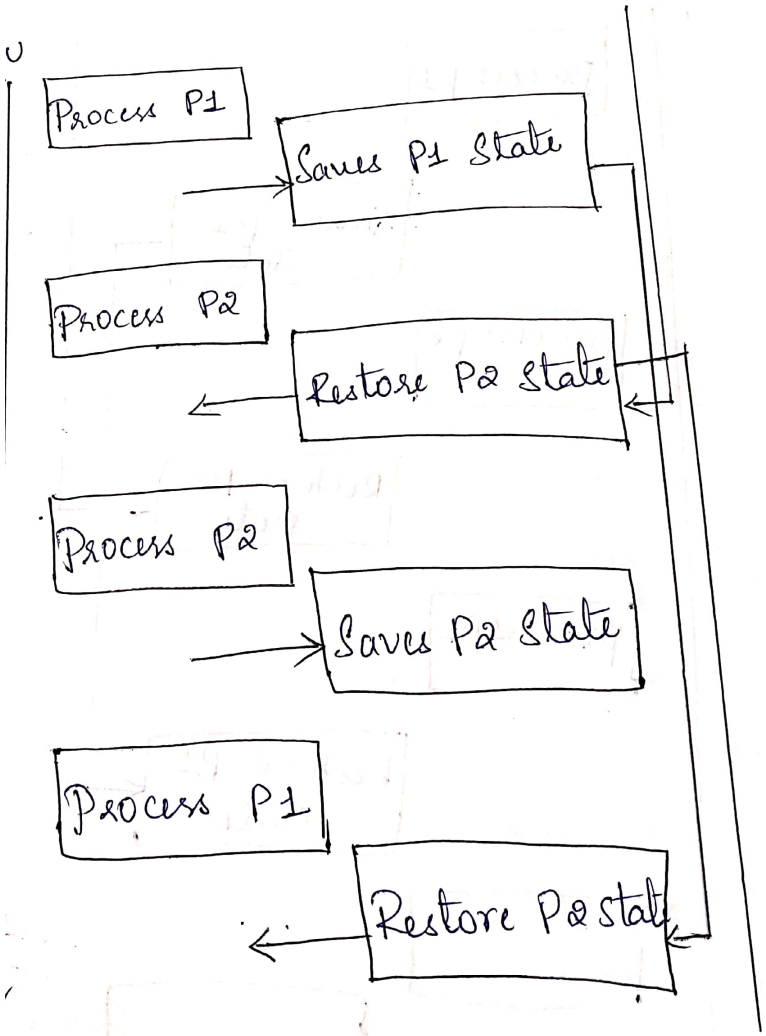
Short term scheduler: It is also called as cpu scheduler. Its main objective is to increase system performance in accordance with the choosen set of online criteria. It is the change of ready state to running state of process.

Medium term scheduler: is a part of swapping. It removes the process from the memory. It reduces the degree of multiprogramming. The medium-term scheduler is in-charge of handling the swapped out-process.

5. Explain about context switch?

Ans: A context switch is the mechanism to store and restore the state or context of a cpu in process control block so that a process execu can be resumed from the same point at a later time. When the scheduler switches the cpu from executing one process to execute another, the state

the current running process
ved into the process control
fter this, the state for the
to run next is loaded
its own PCB & used to set
c, register, etc.



6. Explain about the following scheduling algorithms.

a). FCFS b). SJN c) RR d) Priority e) Multilevel queue.

Ans: a). First Come First Serve: FCFS:-

Jobs are executed on first come, first serve basis. It is a non-preemptive, pre-emptive scheduling algorithm. Its implementation is based on FIFO queue. Poor in performance as average wait time is high.

Process	Arrival Time	Execute Time	Service Time
P0	0	5	0
P1	1	3	5
P2	2	8	8
P3	3	6	16

Wait time:

Process	Wait time: Service time - Arrival
P0	0-0=0
P1	5-1=4
P2	8-2=6
P3	16-3=13

Average Wait time : (0+11+6+13)/4 = 5.75

b). Shortest Job Next (SJN):

It is non-preemptive, pre-emptive, scheduling algorithm. Best approach to minimize waiting time. Easy to implement in Batch system where required cpu time is known in advance.

Process	Arrival Time	Execu^n Time	Service Time
P0	0	5	0
P1	1	3	5
P2	2	8	14
P3	3	6	8

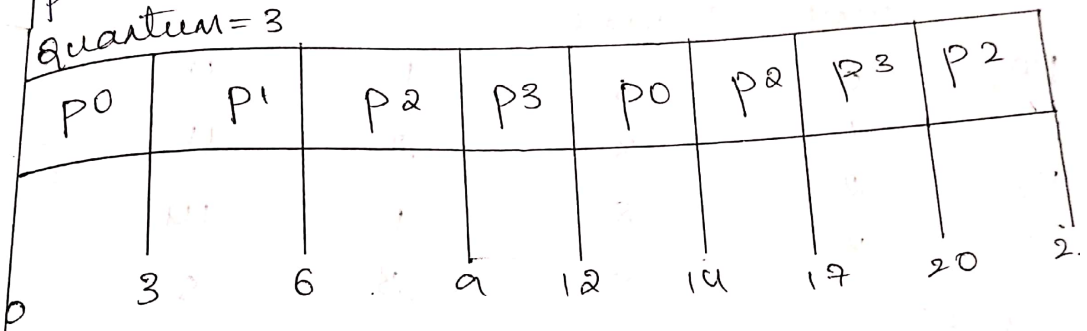
Wait time:

Process	Wait time
P0	0-0=0
P1	5-1=4
P2	14-2=12
P3	8-3=5

Average Wait time : (0+4+12+5)/4 = 5.25.

c) Round Robin Scheduling: Round Robin is the preemptive process scheduling algorithm. Each process is provided a fix time to execute called a quantum. Once a process is executed for a given time period, it is preempted & other process executes for a given time period.

Quantum = 3



Wait time:

Process	Wait time
P0	(0-0) + (12-3) = 9
P1	3-1 = 2
P2	(6-2) + (14-9) + (20-17) = 12
P3	(9-3) + (17-12) = 11

Average wait time : (9 + 2 + 12 + 11) / 4 = 8.5.

d. Priority: Priority scheduling is a non-preemptive algorithm and one of the most common scheduling algorithms in batch systems. Each process is assigned in a priority. Process with highest priority is to be executed first & so on.

Process	Arrival Time	Execution Time	Priority	Service Time
P0	0	5	1	0
P1	1	3	2	11
P2	2	8	1	14
P3	3	6	3	5

Wait time

Process	Wait time
P0	0-0=0
P1	11-1=10
P2	14-2=12
P3	5-3=2

Average wait time: (0+10+12+2)/4 = 6.

c) Multiple level queue are not an independent scheduling algorithm they make use of existing algorithm to group & schedule jobs, with common characteristics. Each queue can have its own scheduling algorithms. priorities are assigned to each queue.

⇒ Explain about various operations on a process.

Ans: a) Process Creation: Processes need to be created in the system for different opera's. This can be done by user request for process creation, system initializ^n etc. A process may be created by another process using fork(). The creating process is called the parent process & the created process is the child process. A child process can have only one parent process but parent process may have many children.

b). Scheduling/Dispatching: The event or activity in which the state of the process is changed from ready to running. It means the operating system puts the process from ready state into running state. Dispatching is done by operating system when the resources are free or the process has higher priority than ongoing process.

c). Process Preemption: An interrupt mechanism is used in preemption that suspects the process executing currently and the next process to execute is determined by the short-term scheduler. Preemption makes sure that all the processes get some cpu time for execution.

d). Process Blocking: The process is blocked if it is waiting for some work/event to occur. This event may be I/O as the IO events are executed in the main memory & dont require processor. After the event is complete, the process again goes to ready state.

d) Process Termination: After the process has completed the execu^ of its last instruc^. It is terminated. The resources held by a process are released after it is terminated. A child process can be terminated by its parent process. If its task is no longer relevant.

8). What is interprocess communication (IPC)? Explain types.

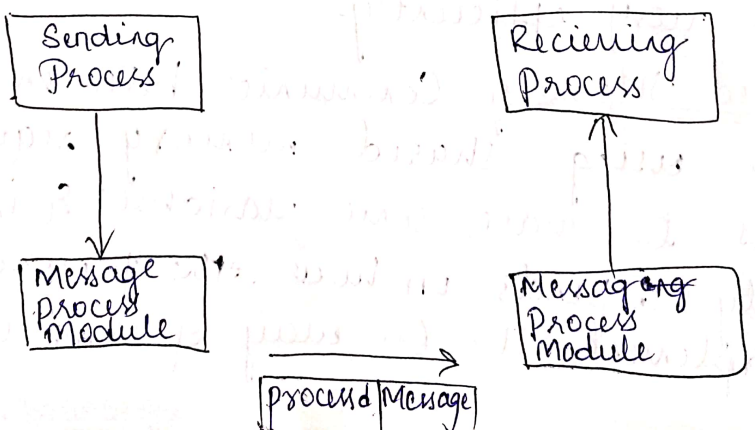
Ans: Interprocess communica^ can be of 2 types, independent and co-operating. An independent process is not affected by the execu^ of other processes, while a co-operating processes can be affected by other executing processes. Though one can think that those processes, which are running independently, will execute very efficiently.

a) Shared Memory: Communic^ between processes using shared memory requires processes to share some variable & it completely depends on how the programmer will implement it. One way of communic^.

between using shared memory can be imag. gined like this. Suppose process1 & process2 are executing simultaneously, & they share some resources or use some inform^ from another process. Process1 generates inform^ about certain computa^s or resources being used & keeps it as a record in shared mem. org. When process2 needs to use the shared inform^, it will check in the record stored in shared memory and take note of the inform^ and act accordingly.

b). Messaging Passing method:

Establish a communic^ link.
Start exchanging messages using basic primitives.
We need at least two primitives
send. recieve.



9). Dfine system call? Explain about different types of system calls with syntax & example.

Ans: The interface between a process and an operating system is provided by system calls. They are available in assembly level language instruc^ns. System calls are usually made when a process in user mode requires access to a resource.

Types:

Process Control: It deals with process such as process crea^, termina^ etc.

File Management: These system calls are responsible for file manipula^ such as creating, reading, writing a file.

Device Management: These sc are respon sible for device manipula^ such as reading from device buffers.

Informa^ Maintainence: these sc handle informa^ & its transfer btw the OS & user program.

OPERATING SYSTEM

ASSIGNMENT- 03

MENAAZ FATIMA
19D01A0530
CSE-B
II nd Year.

1. What is deadlock? What are necessary conditions leads to deadlock?

Ans: Deadlock is a situation where a set of process are blocked because each process is holding a resource and waiting for another resource acquired by some other process.

Conditions which leads to deadlock are

a). Mutual Exclusion
b). Hold & Wait
c). No Preemp^
d). Circular Wait

2. What are methods for handling deadlock

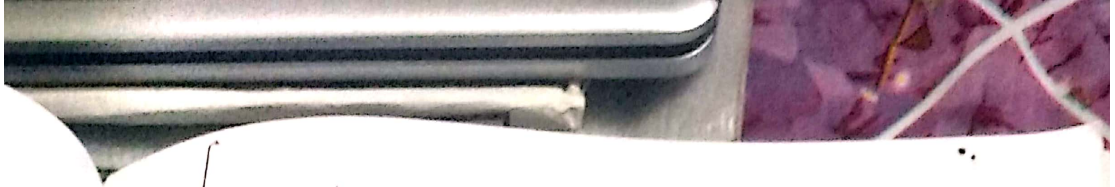
a). Deadlock Prevention: the idea is to not let the system into a deadlock state. Prevention is done by negating one of above mentioned necessary conditions.

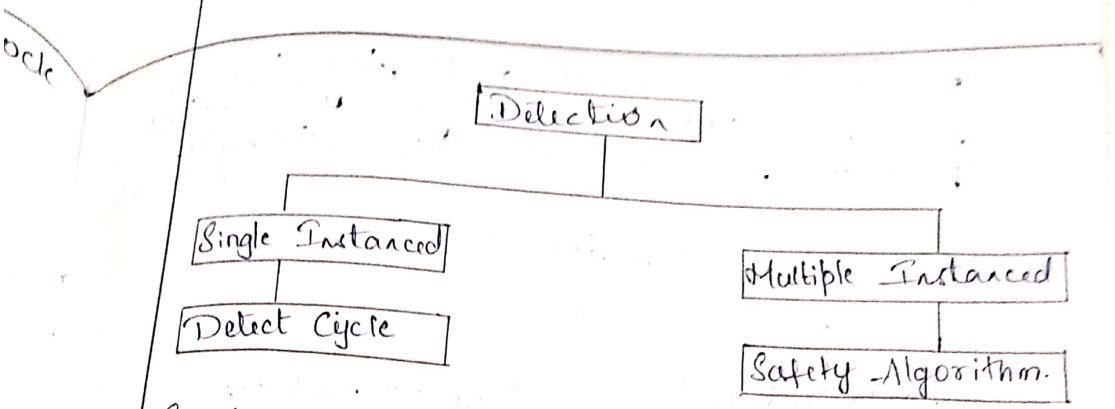
2. Deadlock detec^ & recovery: Let deadl occur, then do preemp^ to handle once occured.

3. Ignore the problem altogether: If deadlock is very rare, then let happen & reboot the system. This the approach that both windows & UNIX take.

3. Explain about deadlock detec^ & re ery from deadlock.

Ans: Deadlock detec^ & recovery: In this approach, the OS doesn't apply an mechanism to avoid or prevent the deadlock. Therefore the system considers that the deadlock will definitely occur in order to get s of deadlocks. The OS periodically checks the system for any deadlo the main task of the OS is detecti the deadlock. The OS can detect th deadlocks with help of resource alloc^ graph.



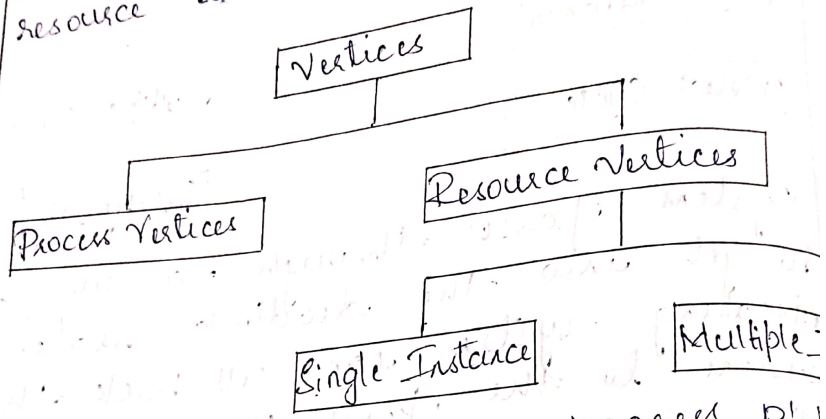


System passes through various states to get into the deadlock state. The operating system can roll back the system to the previous safe state. For this purpose, OS needs to implement check pointing at every state.

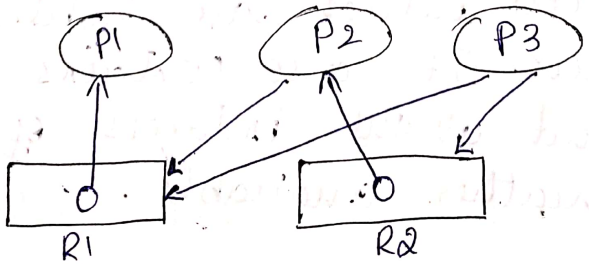
4. Explain about resource allocation graph (RAG) with an example.

Ans: The resource allocation graph is the pictorial representation of state of a system. As its name suggests, the resource allocation graph is the complete information about all the process which are holding some resources. It also contains the information about all the instances of all the resources whether available.

In resource alloc^ graph the proce
is represented by a circle while th
resource is represented by rectangle.



Example: Lets consider 3 proces P1,
& two types of resources R1 & R2. T
resources are having 1 instance e
According to graph R1 is being u
by P1, R2 is holding P2. & waiting
R1; P3 is waiting for R1 as well as
The graph is deadlock free since
cycle is being formed in the graph



5. Explain about process synchronisation?
Explain about race condition.

Ans: Process Synchronisation: It is task phenomenon of co-ordinating the execution of processes in such a way that no two processes can have access to the same shared data.
It is a procedure that is involved in order to preserve the appropriate order of execution of co-operative processes. Process synchronisation is mainly needed in a multi process system where multiple processes are running together, & more than one processes try to gain access to the same shared resource or any data at same time.

Race Condition: At the time when more than one process is either executing the same code or accessing the same memory or any shared variable, in that condition, there is a possibility that the output or the value of shared variable is wrong so for that purpose

all the purposes are doing the job to say that my output is correct. This condi^n is commonly known as race cond^n.

6. Explain about critical sec^ problem? Explain about solu^n for critical sec^ problem?

Ans: A critical sec^ is a code segment that accesses shared variables & has to be executed as an atomic action. It means that in a group of co-operating process, at a given point of time, only one process must be executing its critical section. If any other process also wants to execute its critical section, it must wait until the first one finishes. The entry to the critical section is mainly handled by wait() func^ while the exit from the critical is controlled by the signal() func^.

Solution:

Mutual Exclusion: Out of a group of cooperating process, only one process can be its critical section at a given point of time.

Progress: If no process is in the critical sec^n & if one or more threads want to execute their critical sec then any one of these threads must be allowed to get into its critical section.

Bounded Waiting: After a process makes a request for getting into its critical section, there is a lim for how many other processes can get into their critical section, bef the process request is granted.

Explain the following classical problems of synchronisn.

a). bounded-buffer

b). dining philosopher

c). the readers-writers

Ans: a). Bounded Buffer Problem:

A producer tries to insert data into an empty slot of the buffer. A consumer tries to remove data from a filled slot in the buffer. As you might have guessed by now, those two processes must produce the expected output if they are being executed concurrently.

Solution: One solution to this process is to use semaphore. The semaphore which will be used here are

* m, a binary semaphore which is used to acquire & release the lock.
* empty, a counting semaphore whose initial value is the no of slots in the buffer.
* full, a counting semaphore whose initial value is 0.

b). Dining Philosopher's Problem:

At any instant, a philosopher is either eating or thinking when a philosopher wants to eat, he uses two chopsticks one from their left & one from their right. When a philosopher wants to think

he keeps down both chopsticks at their original place.

Solution: From the problem it's clear that a philosopher can think for an indefinite amount of time. But when a philosopher starts eating, he has to stop at some point of time. An array of five semaphores, stick[5] for each of the five chopsticks.

c) Reader Writers Problem:

There is a shared resource which should be accessed by multiple process. There are two types of process, they are read & write. Any no. of reader can read from the shared resources simultaneously but only one writer can write to the shared resource. When a writer is writing data to the resource, no other process can access the resource.

Solution: We use one mutex m & a semaphore w. An integer variable read_count is used to maintain the no of readers currently accessing the resource. The variables read_count is initialized to 0. A value of 1 is given to m & w.

8) What is semaphore? What are the types of semaphores? Explain.

Ans: In 1965, Dijkstra proposed a new & very significant technique for managing concurrent processes by using the value of a simple integer variable to synchronise the progress of interacting process. This integer variable is called semaphore. It is basically a synchronising tool & is accessed only through two low standard atomic operations wait & signal designed by P(S) & V(S).

Binary Semaphore:

It is a special type of semaphore used for implementing mutual exclusion hence it is called a mutex. A binary semaphore is initialised to 1 & only takes the values 0 and 1 during the execution of program. In BS the wait operation only if the value of semaphore = 1, & the signal operation succeeds when the semaphore = 0.

Counting Semaphore:

-these are used to implement bounded concurrency. -the counting semaphores can range over an unrestricted domain.
-these can be used to control access to a given resource that consists of a finite no. of instances. Here the semaphore count is used to indicate the no of available resources.

9. Explain about monitors in process synchronisn in detail.

Ans: -the monitors is one of the ways to achieve process syncheonisn. The monitor is supported by programming languages to achieve mutual exclusion between process.

* It is the collen of condition variables and procedures combined together in a special kind of module or a package.

* the processes running outside the monitor can't access the internal variable of the monitor but can call procedures of the monitor.

* only one process at a time can execute code inside monitors.

Syntax:

```
Monitor Demo // Name of monitor
{
variables;
condition variables;
procedure P1 {... }
procedure P2 {... }
}
```

10). What is IPC? Explain the following IP methods briefly?

a). pipes b). message passing

c). shared memory d). FIFO.

Ans: Inter process communic^ (ipc) is use for exchanging data between multi threads in one or more processes o programs. The process may be runni on single or multiple computers con by a network.

a). Pipes: Pipe is widely used for communica^ between two related processes. This is a half-duplex method so the first process communicates with the second process.

b). Message Passing: It is a mechanism for a process to communicate & synchronise. Using message passing, the process communicates with each other without resorting to shared variables.

c). Shared Memory:

Shared memory is a memory shared between two or more processes that are established using shared memory between all the processes.

d). FIFO:

Communica^ between two un-related processes. It is full-duplex method, which means the first process can communicate with the second process & the opposite can also happen.

Operating System

Assignment - 04

Menaaz Fatima
19D01A0520
CSE-B

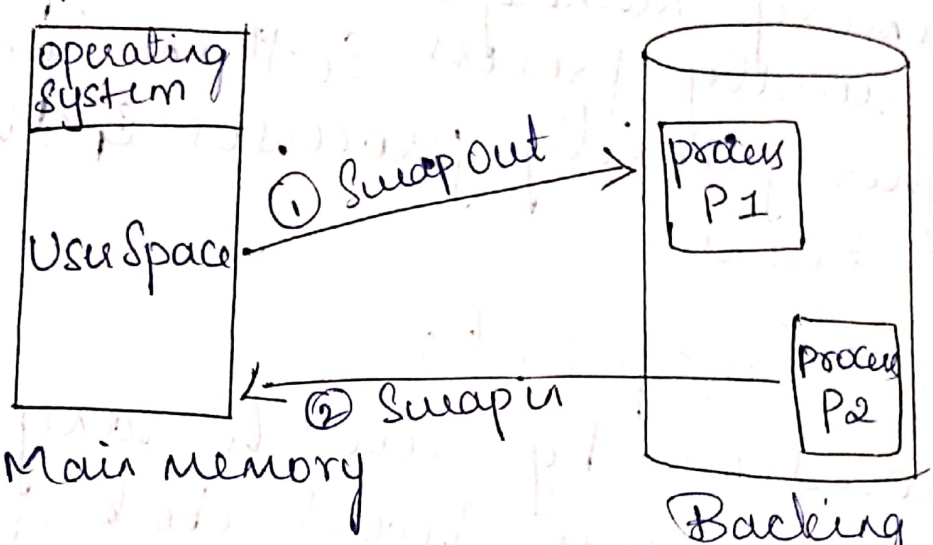
1. Explain about physical versus logical address space.

Ans: The address that is generated by the cpu is commonly referred to as the logical address. It is basically a virtual address. The logical address is basically the address of an instruc^ or data used by any program. The set of all logical addresses that are generated by any program is referred to as logical address space.

The address that is loaded into the memory address register of the memory is commonly referred to as physical address. A physical address cannot be accessed by the user directly but the user can calculate the physical address with the help of logical address.

2. What is swapping? Explain.

Ans Swapping is a memory management technique and is used to temporarily remove the inactive programs from the main memory of the computer system. The swapping process is also known as a technique for memory compact.



In the above diagram, suppose a multi-programming environment with a round-robin scheduling algorithm, whenever the time quantum expire then the memory manager starts to swap out those processes that are just finished and swap another process into the memory That has been freed.

The swapping of process by the memory manager is fast enough that some processes will be in memory, ready to execute, when the cpu scheduler wants to reschedule cpu.

Swapping algorithm is based on priority. If any higher priority process arrives and wants service then memory management swaps out lower priority process & then load the higher priority processes & then execute them.

Swap In & Swap Out:

The procedure by which any process gets removed from hard disk & placed in the main memory or RAM.

Swap Out is the method of removing a process from main memory or RAM & then adding it to Hard disk.

3. What is Contiguous Memory Alloca^? Explain its methods.

Ans: In contiguous memory alloc^, each process is contained in a single contiguous block of memory. Memory is divided into several fixed-size partitions. Each partition contains exactly one process. When a parti.^ is free, a process is selected from the input queue & loaded into it. The free blocks of memory are known as holes.

First Fit Alloc^: - the first hole that is big enough is allocated to program.

Best Fit Alloc^: - the smallest hole that is big enough is allocated to program.

Worst Fit Alloc^: - the largest hole that is big enough is allocated to program.

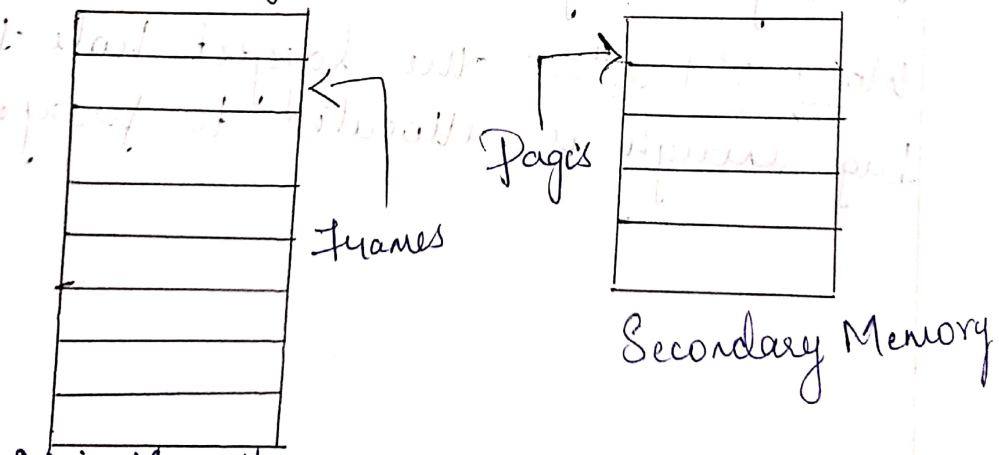
4. Explain about paging technique in detail.

Ans: Paging permits the physical address space of a process to be non-contiguous. It is fixed size partitioning scheme. In paging technique, the secondary memory, and main memory are divided into equal fixed-size partitions.

Method of Paging:

The paging technique divides the physical memory into fixed size blocks known as frames & logical memory into blocks of same size known as pages.

The frame has the same size as that of a page. A frame is basically a place where a page can be placed.



Each process is mainly divided into parts where the size of each part is same as page size.

Pages of a process are brought into the main memory only when there is requirement otherwise they reside in the secondary storage.

One page of a process is mainly stored in one of the frames of memory. Also the pages can be stored at diff locations of memory but always the main priority is to find contiguous frames.

Translation of Logical Address into Physical Addr

CPU always generates a logical address. In order to access the main memory always a physical address is needed.
It consists of two parts
1. Page Number (p)
2. Page Offset (d)

5. Explain about Segmentation technique in detail.

Ans: Segmentation: is another memory management scheme that supports the user-view of memory. Segmentation allows the breaking of the virtual address space of a single process into segments that may be placed in non-contiguous areas of physical memory.

Segmentation with Paging: Both paging and segmentation have their advantages and disadvantages, it is better to combine these two schemes: to improve each other. The combined scheme is known as 'Page the Elements'. Each segment in this scheme is divided into two pages & each segment is maintained in a page table. So the logical address is divided into.

Segment Number (S)

Page Number (P)

-The displacement or Offset Number (D).

6. Explain about Segmentation with Paging.

Ans: Both paging and segmentation have their advantages and disadvantages, it is better to combine these two schemes to improve each other. The combined scheme is known as 'Page Elements'. Each segment in this scheme is divided into two pages & each segment is maintained in a page table. So the logical address is divided into.

Segment Number (s)

Page Number (p)

The displacement or Offset Number (D).

Explain about demand paging.

According to the concept of virtual memory, in order to execute some process only a part of the process needs to be present in the main memory which means that only a few pages will only be present in the same memory at any time. Deciding, which pages need to be kept in the main memory & which need to be kept in the secondary memory

is going to be difficult because we cannot say in advance that a process will require a particular page at particular time.

Therefore, to overcome this problem, there is a concept called demand paging. It suggests keeping all pages of the frames in the secondary memory until they are required. Whenever any page is referred for the first time in the main memory, then the page will be found in the secondary memory.

8. What is Page Replacement? Explain its algorithm.

Ans: Page Replacement Algorithm: The PRA decid which memory page is to be replaced. The process of replacement is sometime called swap out or write to disk. Page replacement is done when the requested page is not found in the main memory.

Algorithms:

1. Optimal Page Replacement Algorithm:
-this algorithm replaces the page which will not be referred for so long in future. Although it can not be practically implementable but it can be used as bench mark.

2. Least Recent Used LRU Algorithm:
-this algorithm replaces the page which has not been referred for a long time.
-this algorithm is just opp to the optional page RA.

3. FIFO: In this algo^{n}, a queue is maintained. The page which is assigned to the frame first will be replaced first. The page which resides at the rare end of queue will be replaced on every page fault.

14/7/2021

OPERATING SYSTEMS
ASSIGNMENT-05

MENAAZ
FATIMA
19D01A0580
CSE

1) Define file. Explain about attributes of file. Explain about operations on file.

Ans: A file can be defined as a data structure which stores the sequence of records.

Attributes of file:

1. Name: Every file carries a name by which the file is recognised in the file system. One directly cannot have two files with the same name.
2. Identifier: Along with the name each file has its own extension which identifies the type of file.
3. Type: In a file system, the files are classified in different types such as video files, audio files, text files etc.

4. Location: In the file system, there are several locations on which, the files can be stored. Each file carries its location as its attribute.

5. Size: The size of file is one of the most important attribute. By size of file, we mean the no. of bytes acquired by the file in the memory.

6. Protection: The admin of the computer may want the different protects for different files. Each file carries its own set of permissions to the different group of users.

7. Time & Date: Every file carries a time stamp which contains the time & date on which the file is last modified.

Operations on file:

Create: Creation of file is most important operation on the file. Different types of files are created by different methods, text editors are used to create a text file etc.

2. **Write:** Writing the file is different from creating the file. The OS maintains a write pointer for every file which points to the posⁿ in the file from which the data needs to be written.

3. **Read:** Every file is opened in three different modes. Read, Write & append. A read pointer is maintained by the OS, pointing to the position up to which the data has been read.

4. **Re-position:** is simply moving the point forward or backward depending upon the user's requirements. Also called seek

5. **Delete:** Deleting the file not only delete all the data stored inside the file, it also deletes all the attributes of the fil The space which is allocated to the file will now become available.

6. **Truncate:** Truncating is simply deleting th file except deleting attributes. File is not completely deleted although, the informaⁿ stored inside file get replaced.

2). Explain about access methods on a file.

Ans: 1. Sequential Access: In sequential access the OS read the file word by word. A pointer is maintained which initially points to the base address of file. If user wants to read first word of file then the pointer provides that word to the user & increases its value by 1 word

2. Direct Access: is mostly required in the case of database system. Suppose every block of the storage stores 4 records & we know that the record we needed is stored in 10th block. In that case, direct access will give the required result despite of the fact that the operating system has to perform some complex tasks such as determining the desired block other number.

3. Indirect Access: If a file can be stored on any of the filed then an index can be assigned to a group of certain records.

However, a particular record can be accessed by its index. The index is nothing but the address of record of a file. In index, accessing, searching in a large database became very quick & easy but we need to have some extra space in the memory to store the index value.

3). Explain about different type of allocation methods.

Ans: 1. Contiguous Allocation: If the blocks are allocated to the file in such a way that all logical blocks of the file gets the contiguous physical block in the hard disk then such alloc^n scheme is known as contiguous alloc^n.

2. Linked list Allocation: solves all problem of contiguous alloc^n. In LLA, each file is considered as the linked list of disk blocks However, the disks blocks allocated to a particular file need not be contigous on the disk. Each disk block allocated to file contains a pointer which points to the next disk block allocated to same file

3. File Allocation Table: FAT main disadvantage of linked list alloc^ is that random access to a particular block is not provided. In order to access a block, we need to access all its previous blocks. To overcome this drawback, file alloca^ table is maintained, which gathers all the disk block links. FAT needs to be cached in order to reduce the no of head seeks. Now head doesn't need to traverse all disk blocks.

4. Indexed Allocation Scheme: Instead of maintaining a file alloc^ table of all the disk pointers, indexed alloca^ scheme stores all the disk pointers in one of the blocks called as indexed blocks. It doesn't hold the file data, but it holds the pointers to all the disk blocks allocated to that particular file.

5. Linked Index Allocation: In index alloca^, the file size depends on the size of a disk block. To allow large files, we have to link several index blocks together.

* Small header giving the name of file.
* Set of first 100 blocks addresses.
* Pointer to another index block.
* For the larger files, the last entry of the index block is a pointer which points to another index block.
* Also called as schema.

6). Inode: In UNIX based operating syst each file is indexed by an inode. Ino are the special disk block which is created with the creation of file syst No' of files or directories in a file syste depends on no of inodes in the system. I includes following inforna?

* Attributes of file
* A no of direct blocks which contain pointers to first 12 block of file.
* Single indirect pointer, that points to index block.
* Double indirect pointer.
* Triple indirect pointer.

11. Explain about free space management.

A file system is responsible to allocate the free blocks to the file, it has to keep track of all the free blocks present in the disk. There are two approaches that free block and in the disk is managed.

1. Bit vector: In this approach the free space list is implemented as a bit map vector. It contains the number of bits where each bit represents each block. If the block is empty then the bit is 1 otherwise it is 0. Initially all the blocks are empty therefore each bit in the bit map vector contains 1.

2. Linked list: It is another approach to free space management. This approach suggests linking together all the free blocks & keeping a pointer in the cache which points to the first free block. All the free blocks on the disks will be linked together with a pointer. When a block gets allocated, its previous free bl will be linked to its next free block.

5). Explain the following system calls with syntax. iseek(), stat(), ioctl().

Ans: a) iseek: used to change the loc' of read/write pointer of file descrip^n. Location can be set either in absolute or relative terms.

Func^ Defin^:

off_t iseek(int fildes, off_t offset, int whence);

b) Stat(): used to access the files inform such as type of file owner of the file, file access permissions etc.

```
Struct stat{}{
mode_t st_mode;
ino_t st_ino;
dev_t st_dev;
time_t st_atime;
long st_blocks;
};
```

c) ioctl(): control device. It manipulates the underlying device parameters of special fil

```
#include <sys/ioctl.h>
int ioctl(int d, int request, ...);
```

Argument d must be file desciptor. Secon argument is device dependent request code. Third argument is an untyped pointer memory.