UNIT-1

Introduction to Operating System

1.1 Need of Operating System

The need for operating system arises from that user need services and OS provides that services.

A user application would need to store information. The OS makes memory available to an application when required.

Similarly user programs use input facility to communicate with the application using key board or mouse.

OS provides following resources for many application (like Image processing, Games)

- ✓ Processing information
- ✓ Storage information
- ✓ Mechanism to input information
- ✓ Provision for outputting information.

Concept of Operating System

Operating system:

The Operating System Controls and coordinates the use of the hardware among the various application programs for the various users.

It hides complexity of its hardware from user and poride easy interface.

The operating system acts as manager of these resources and allocates them to specific programs and users as necessary for tasks efficiently.

The primary goal of operating system is convenience for the user. The operating system makes the use of system easier.

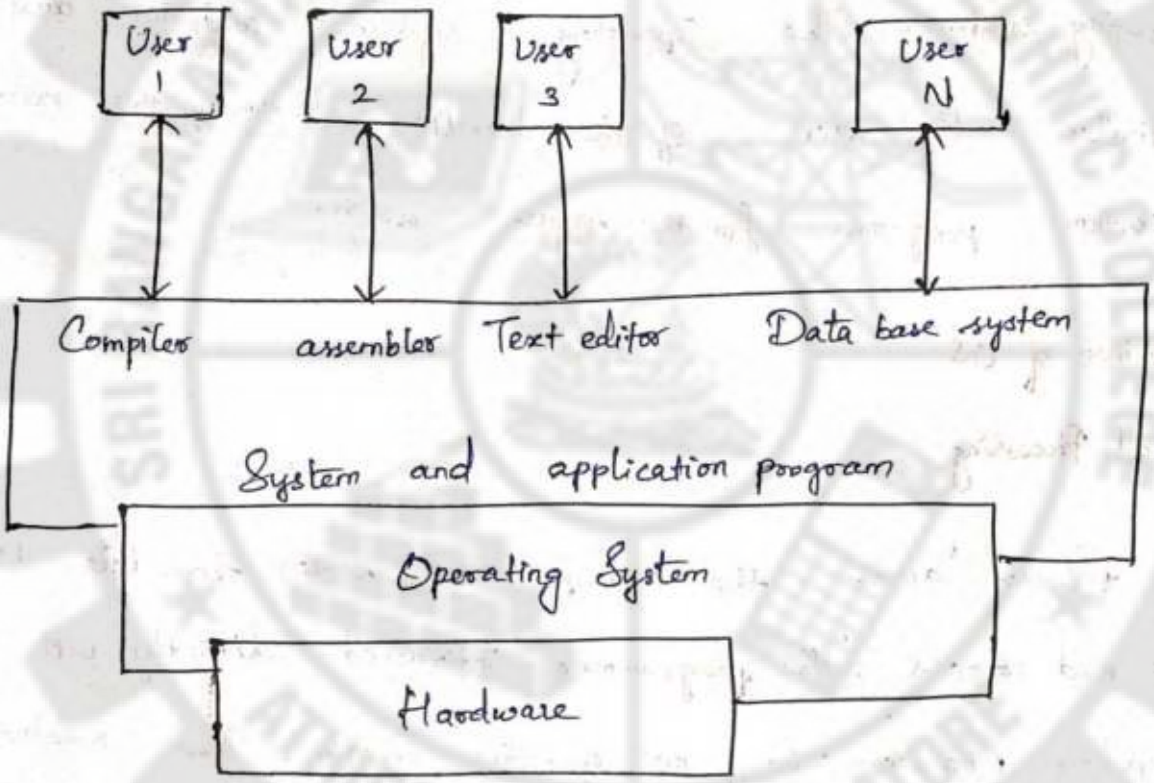
The secondary goal of operating system is Efficient operation of the computer system.

Components of Computer System:

A computer can be divided into four Components.

- Hardware
- Operating System

- Application Programs
- Users



Component of Computer System.

Hardware: The hardware of computer system includes the CPU, the memory, I/O devices. They are the basic resources.

Application program includes compilers, database system, games and busines programs. They define the way in which the hardware resources are to be used to solve the computing problems of the users.

Users: There are may different users trying to solve diffrent problems using different application programs.

Operating system: The Operating system controls and coordinates the user of the hardware among the various application programs for the various users.

Evolution of OS

Serial Processing

Users access the computer in series from late 1940's to mid 1950's, the programmer interacted directly with computer hardware i.e., no operating system. These machines were run with a consde consisting of display lights, toggle switches, some foom of input device and a printer.

Simple Batch Systems.

To speed up processing, jobs with similar needs are batched together and run as a group.

The operator will sort programs into batches with similar requirements.

The problems with batch systems are lack of interaction between the user and job.

CPU is often idle, because the speeds of the mechanical I/O devices are slower than CPU.

Personal- Computer Systems (PCs)

A computer system is dedicated to a single user is called personal computer, appeared in the 1970s.

Micro computers are considerably smaller and less expensive than mainframe computers.

The goals of the operating system have changed with time; instead of maximizing CPU and peripheral utilization, the systems developed for maximizing user convenience and responsiveness.

Parallel Systems.

Most systems to date are single-processor systems; that is they have only one main CPU.

Multiprocessor have more than one processor.

Advantages:

- ✓ Throughput
- ✓ Save money by sharing peripherals, cabinets and power supplies
- ✓ Increase reliability
- ✓ Fault-tolerant.

Types of OS.

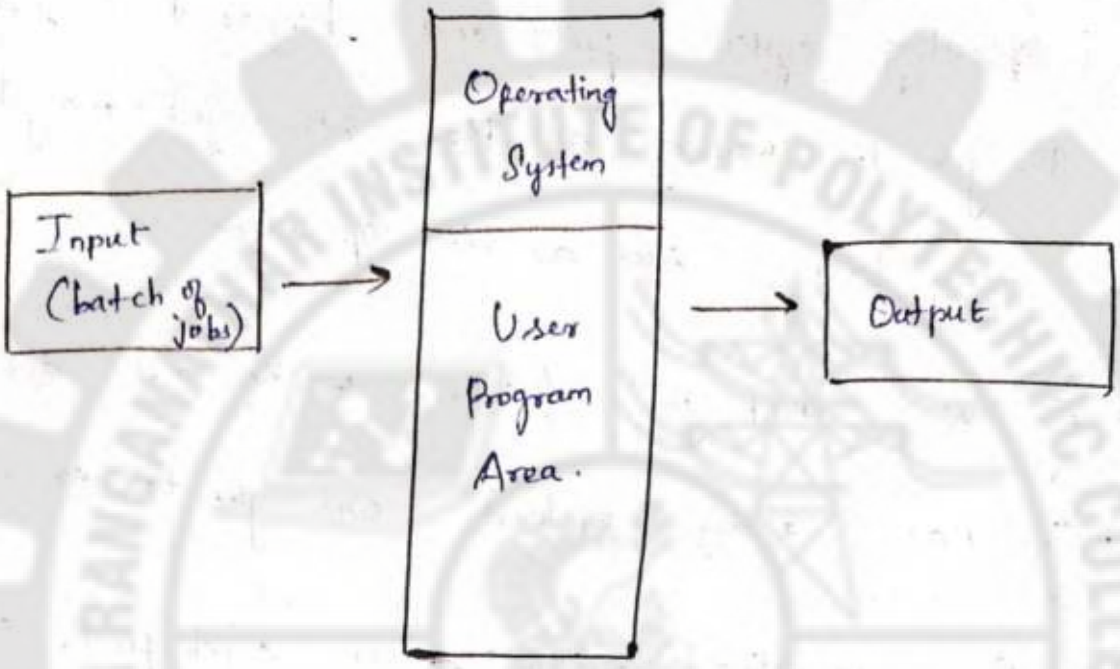
1. Batch OS.

The users of such system did not interact directly with the computer system, the user prepare job consisting of data, program and control information about the job and use to submit it to computer operator.

The batch system has no interaction between the user and the job while the job is executing.

The CPU is often idle. This idleness occurs because the speed of the mechanical I/O devices is slower than electronic devices.

It is fairly simple its major task was to transfer control automatically from one to next.



Advantage:

- Simple
- Difficult to debug program
- A job could enter an infinite loop.

Disadvantage:

- Debugging is possible only offline, after output appears.
- Operation were too much time consuming.

II. Multiprogramming Operating system:

Multiprogramming increases CPU utilization by organizing jobs so that the CPU always has one job to execute.

The OS keeps several jobs in memory simultaneously.

The operating system picks and begins to execute one of the jobs in memory and the other job may have to wait for some task, such as I/O operation.

Eventually the first job finishes waiting and gets CPU back. CPU has always at least one job to execute.

When the operating system selects a job pool, it loads that job into main memory is called job scheduling.

Advantage:

High and efficient CPU utilization.

User feels that many programs are allotted CPU almost simultaneously.

Disadvantage:

- CPU scheduling is required
- To accomodate many jobs in memory, memory management is required.

IV. Real time Operating System.

- A real time operating system is used where there are rigid time requirement on the operation of a Processor or the flow of data, and thus is often

used as a control device in a dedicated application.

- RTOS has very little user interface, capability and no end user utilities.
- A real-time operating system has well-defined, fixed time constraints processing must be done with the defined constraints, or the system will fail.

There are two flavours of real-time systems:

1. Hard real-time
2. Soft real-time

Hard real time:

- A hard real-time system guarantees that critical tasks complete on time.
- Secondary storage of any sort is usually limited or missing, with data being stored in short-term memory or in read-only memory (ROM)
- Most advanced operating system feature such virtual memory is not found in real-time system.
- Ex: oil refinery (plant), ATM.

Soft real time system:

✓ A less restrictive type of real-time Operating system is a soft-real time system, where a critical real-time task gets priority over other tasks, and retains that priority until it completes.

✓ Soft real-time system have more limited utility that hard real-time system.

✓ They are risky if used for industrial control and robotics because of their lack of deadline support.

Ex: digital audio multimedia.

V. Multitasking Operating System or Time sharing OS.

✓ It is a logical extension of multiprogramming operating System.

✓ The ability to execute more than one task at the same time, a task being a program is called multitasking.

✓ The terms multitasking and multiprocessing are often used interchangeably although multiprocessing implies that more than one CPU is involved.

✓ The time - shared operating system uses CPU scheduling and multiprogramming to provides each user with a small portion of a time - shared computer

A program in execution is called process

Time - Slice is amount of that is given to process to execute on processor (CPU)

There are two basic types of multitasking :

- Pre - emptive multitasking (forcibly)
- Non pre - emptive or cooperative (multitasking voluntarily)

Preemptive - a program is allowed to execute only for some maximum time duration, after this time duration CPU is forcibly taken away from the program

✓ Minimize the response time for user.

✓ Eg: OS/2, Windows 95, Windows NT, the Amiga OS and UNIX use

Non - pre - emptive :

- Here each program can control the CPU for as long as it needs it.

Ex: Microsoft Windows 3.x and the MultiFinder (for macintosh computers).

VI. Multithreading Operating System.

- Thread is a flow of control within a process.
- A thread also called a lightweight process (LWP) is a basic unit of CPU utilization
- Thread contains a thread ID, a program counter, a register set, and a stack.
- Process is a heavy eight process, has a single thread of control.
- If the process has multiple threads of control, it can do more than one task at a time.
- Eg. A web browser might have one thread display images or text while other thread retrieves data from the network.

Advantages.

- Responsiveness
- Resource sharing
- Economy
- Utilization of multiprocessor architecture.

Operating System Services or Function of Operating System.

- The operating System provides an environment for the execution of programs.

- The operating system provides services to programs to the users of those programs. The operating system services are provided for the convenience of the programmer.

For user:

Operating system provides services to the user program and system.

1) User Interface:
   - Command line Interface (CLI)
   - Batch interface
   - Graphical User Interface (GUI)

2) Program execution
3) I/O operations
4) File System manipulation
5) Communications.
6) Error Detection.
7) Resource Allocation

8) Accounting

9) Protection

The user view and Operating view.

Operating system can be explored from two view points: the user view and system view.

User view

- The user view of the computer varies by the interface being used.
  - It is top down view
  - Primary goal: user convenience.

1. Single user system

Pc consisting of monitor, keyboard, mouse, and system unit are designed to be used by single person.

In the case, the operating system is designed mostly for ease of use, with some attention paid on performance and none paid to resource utilization.

Performance is important to the user, but it does not matter if the most of the system is sitting idle, waiting for the slow I/O speed of the user.

2) User connected to mainframe or minicomputer.

- Many user are connected to mainframe or mini computer through terminal to access to share the resources and exchange information.
- From that user's point of view, the operating system is designed to maximize resource utilization.

3) User on klorkstations.

- User at workstation, connection to networks of other workstations and servers have dedicated resources at their disposal, but they also share the resource such as networking and server - file, compute and print servers.
- The operating system is designed to compromise between individual usability and resource utilization.

4) Handheld computers.

- Due to power and interface limitations they perform relatively few remote operations.
- The operating system are designed mostly for individual usability. but performance per amount of battery life is important.

5) Embedded Computers.

Some computers have little or no user view for eg: embedded computers.

System view :

- It is bottom up view
- from computers view the operating system is the program that is most intimate with hardware.
- The operating system is a resource allocator.
- The operating system acts as manager of resources - hardware and software - required solve a problem.
- The fundamental goal of OS to execute user progrms and to make solving user problems easier

Case Study

Linux

- Linux Operating System has primarily three components.
- Kernel - kernel is the core part of Linux. It is responsible for all major activities of this operating system. It consists of various modules and it interacts directly with the underlying hardware.

System Library - System libraries are special functions or programs using which application programs or system utilities accesses kernel's features.

System Utility - System utility programs are responsible to do specialized, individual level tasks.

Window 7 :

- Objectives: Learn about main features in each windows 7 edition and what are minimum hardware requirements.

Prerequisites : no prerequisities

Key terms : windows 7 editions, starter, home basic, home premium, professional, enterprise, ultimate, hardware requirements, processor architecture.

There are 6 different Windows 7 editions :

- Starter
- Home Basic
- Home Premium
- Professional
- Enterprise
- Ultimate.

UNIT-2

Process Management

Syllabus.

2.1 Process and Process Management

- Process model overview
- Programmers view of process
- Process states

2.2 Process and Processor Scheduling

- Scheduling Criteria
- First Come First Serve
- Round Robin
- SJF
- SRTN

CONT.

2.3 Schedulers.

- Inter Process communication & synchronization
- Race condition
- Mutual Exclusion
- Monitors.

2.4 Dead lock

- Prevention
- Avoidance
- Detection and recovery

2.1 Process and Process management

Process :

- In computing, a process is an instance of a computer program that is being executed.
- It contains the program code and its current activity.
- Whereas a program is a set of Instructions.

Process v/s Program

- A process is a program in execution, it also includes the current activity, as represented by the value of programing counter and the contents of the program counter.
- A process includes a process stack which contains the temporary data and a data section which contains the global variables.
- While a program is a sequence of instructions written in a manner to obtain the desired output.

Process Control Block.

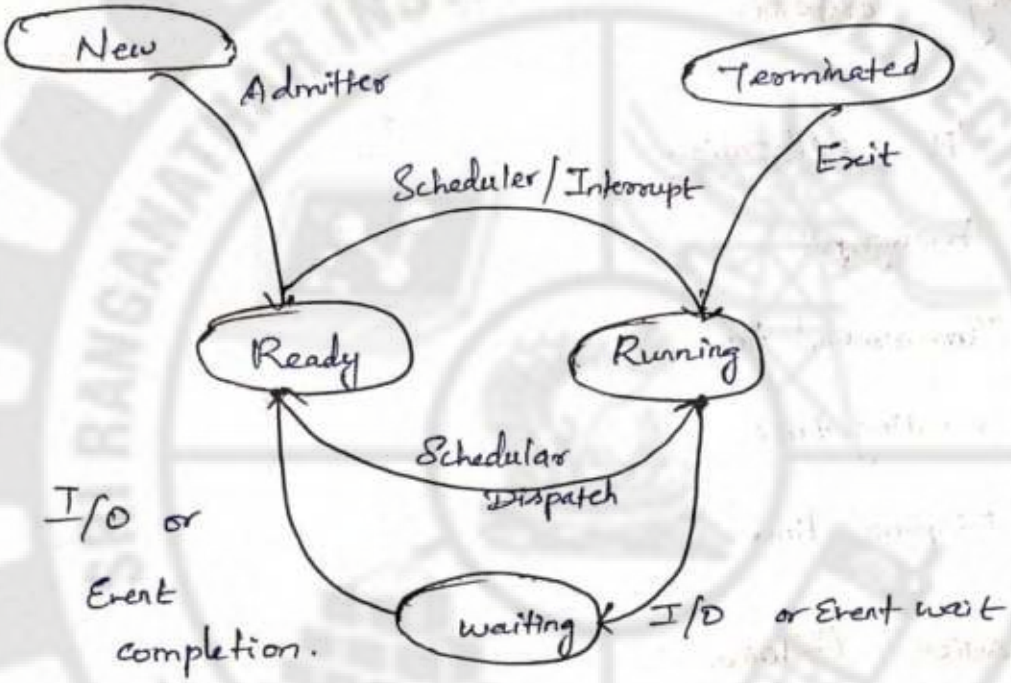
- Each process is represented by a process control block (PCB) which contains information such as:
  - state of the process (ready, running, blocked)
  - register values
  - priorities and scheduling information
  - memory management information
  - accounting information such as open files
  - allocated resources.

Process life cycle (Model)

A process moves through various states during its life:

- new,
- ready,
- running,
- blocked and
- done states and
- transitions between them

Continue..



Process States:

- new : The process is being created
- running: Instructions are being executed.
- waiting: The process is waiting for some event is to occur.
- ready: The process is waiting to be assigned to a processor.

terminated: The process has finished execution.

2.2. Process and Processor Scheduling

Scheduling criteria

- CPU utilization
- Throughput
- Turnaround time
- waiting time
- Response time

Optimization Criteria

- Max CPU utilization
- Max throughput
- Min turnaround time
- Min waiting time
- Min response time

Scheduling Algorithm.

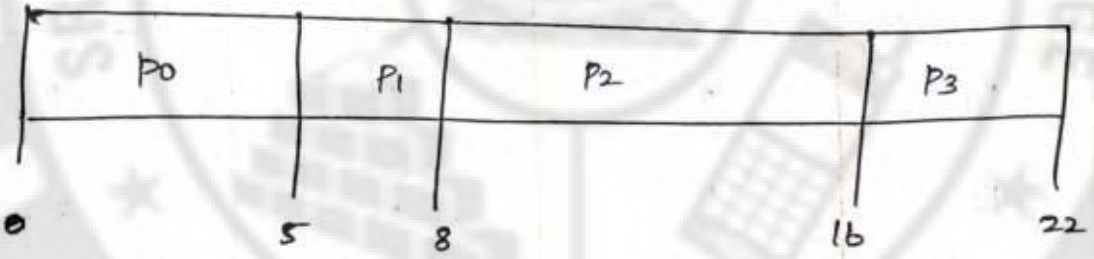
- First Come First Serve (FCFS) Scheduling
- Shortest-Job-First (SJF) Scheduling
- Priority Scheduling
- Round Robin (RR) Scheduling
-

FCFS:

- Jobs are executed on first come, first serve basis.
- Easy to understand and implement

- Poor in performance as average wait time is high.

Process	Arival time	Execute time	Service Time
P_0	0	5	0
P_1	1	3	5
P_2	2	8	8
P_3	3	6	16



wait time each process is following.

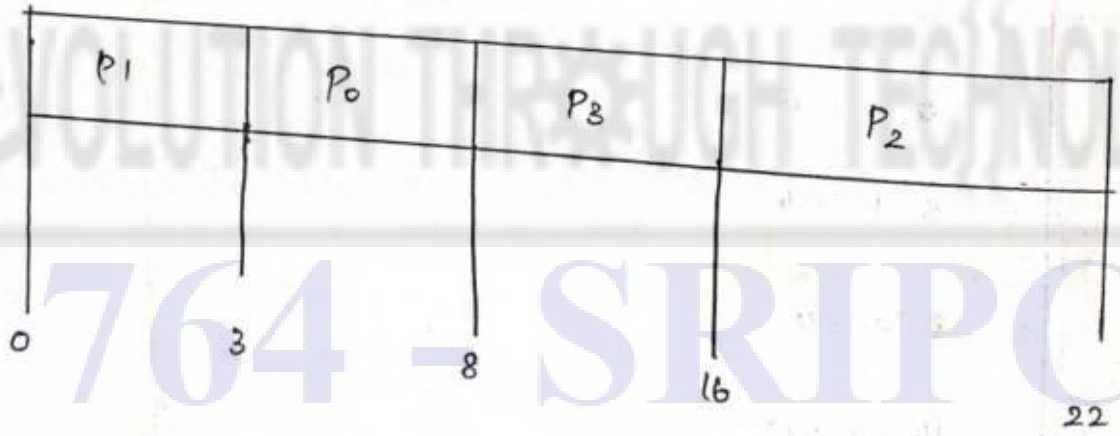
Process	Wait time : Service time - arrival time
P_0	0 - 0 = 0
P_1	5 - 1 = 4
P_2	8 - 2 = 6
P_3	16 - 3 = 13

Average wait time : (0+4+6+13)/4 = 5.55

Shortest Job First (SJF)

- Best approach to minimize waiting time
- Impossible to implement
- Processor should know in advance how much time process will take.

Process	Arrival Time	Execute Time	Service Time
P_0	0	5	0
P_1	1	3	3
P_2	2	8	8
P_3	3	6	16



Average wait Time : $(3+0+14+5)/4 = 5.50$

SRTN (Shortest Remaining Time Next)

It is the Preemptive version of SJF

Also known as Shortest Remaining time First (SRTF) is a scheduling method that is a preemptive version of shortest job next scheduling.

Priority Based Scheduling.

Each process is assigned a priority. Process with highest priority is to be executed first and so on.

Processes with same priority are executed on first come first serve basis.

Priority can be decided based on memory requirements, time requirements or any other resource requirenat

Round Robin Scheduling

Each process is provided a fix time to execute called quantum.

Once a process is executed for given time period. Process is preempted and other process executes for given time period.

Context switching is used to save states of preempted processes.

2.3 IPC (Inter Process Communication)

- In multiprogramming OS more than one process may be running simultaneously. Such processes can communicate with each other, such type of communication is called Inter Process communication.
- IPC is useful in a distributed environment where the communication process may reside on different machine connected with a network.

Example of IPC:

- A shell pipeline in UNIX: ls | wc -l
- Printing a paper in network
- Chat or mail server

Issues Related to IPC.

- How one process can pass information to other process:
  - To share information.

- Two or more process should not get into each other's way:
- Proper sequencing should be maintained in execution of procer where dependencies are present

Advantages and Disadvantage

Advantage:

- Information sharing
- Computation speed up (increase)
- Modularity
- Convenience
  - Many tasks on which work can be done at same time.

Disadvantage:

- Race condition problem
- Process synchronization should be there.

Process Synchronization.

- Process synchronization is a mechanism to ensure systematic sharing of resources among concurrent process.
- Two issues related to process synchronization.
  - Two or more process should not come into each other's way.

- Proper sequencing should be maintained during execution.
- In IPC one problem arises called Race condition.

Race condition / Racing problem :

It is situation where two or more process are reading / writing some shared data and final results depends on relative order of their execution, is called race condition.

At = 1000, At the end of two process A should be 1100. but if P_0 and P_1 permitted to execute in any arbitary fashion then output will not be same.

P_0	P_1
Read (A);	Read (A)
A : A - 100	A : A + 200;
Write (A)	Write (A)

Possibility :1

P_0	P_1
Read (A) :	—
—	Read (A)
A : A - 100	—
Write (A)	—
—	A : A + 200 :
—	Write (A)

Here, Answer will be 1200

Possibility :2

P_0	P_1
Read (A) :	—
	Read (A)
	A : A + 200 :
	Write (A)
A : A - 100	
Write (A)	

- Here two process are reading and writing common variable 'A'. The final value depends on relative execution order of P_0 and P_1, such situation is called race condition.

o This implies that concurrent process are racing with each other to access a shared resource in arbitrary order and procedure wrong final results, so race condition must be avoided.

Critical Section:

A critical section is a code segment of a process where the shared resources are accessed

It is also known as critical region

Only one process should be allowed to execute in its Critical section.

It is the responsibility of an OS to ensure that no more than one process is present in its critical section.

Requirement of Cafical Section:

Solution:

An ideal critical section solution meet the following three condition:

Mutual exclusion:

At any time, at most one of the co-operating process should be executing in its critical section.

Progress.

When no any process is running in its critical section and other process wants to enter its critical section, it should be allowed immediately

If more than one process is waiting to enter their critical sections, one of them should be selected to enter.

Bounded waiting:

When more than one process is waiting to enter their critical section, one of them is selected to enter in critical section.

Other processes simply wait. But such type of waiting must be limited. There should not be starvation.

It should not be such that one process waits forever and other get entry in critical section more than once.

Other processes allowed to enter into their CS before a requesting process is granted to enter in its CS.

Critical Section Solution.

- General Structure

```
do
{
    Entry Section
    Critical section
    Exit section
    Remainder section
} while (1);
```

Process execution divided into four section for co-operating processes.

- Entry Section:

This section is executed when a process wants to enter its critical section.

Eligibility of process to enter CS is checked here. If all requirements are satisfied process is allowed to enter else it will wait till the entire requirements are met.

- Critical section:

In this code segment, process accesses a shared resource

Only one co-operating process will be in its CS and enjoying exclusive access to the shared resource.

Exit section:

This code segment is executed when a process exits its CS

In this process performs certain operations, indicating its exit from the CS

This process enables one of the exit waiting process to enter its CS.

Remainder Section:

This is the remainder part of process's code.

In this section, process performs tasks which are independent from other processes. Means they do not include use of any shared resources.

Some tools are available to implement entry and exit section such as semaphore, mutex and monitor.

Semaphore

- Semaphores are used to indicate the status of shared resources whether they are free or being used.
- They can be used to control the entry of a Process in its critical section.

Semaphore Operations.

Semaphores can be accessed only through the two operations:

Down and Up

Down (or wait) and Up operation:

```
Down (s)
{
  while (s<=0)
  sleep ();
  s--;
continue
}.
```

This operation checks to see if the semaphore value is less than or equal to zero.

```
Up(S)
{
    S++;
    Wakeup();
}
```

Semaphore Types:

- Binary Semaphore
  - Simpler to implement
  - It is used to provide exclusion mutually
- Counting semaphore

  It is used to manage multiple instance of a resource.

Monitor.

A monitor is a synchronization construct that allows threads to have both mutual exclusion and the ability to wait (block) for a certain condition to become true.

Monitors also have a mechanism for signalling other threads that their condition has been met.

A monitor consists of a mutex (lock) object and condition variables.

2.4 Deadlock.

Whenever process needs any resource, it request for it. OS grants Resource if it is free. If resource is not free, process will wait for it to be free

Eg: Traffic Jam.

Device allocation:

- process 1 requests tape drive 1 & get it
- Process 2 requests tape drive 2 & get it
- Process 1 requests tape drive 2 but is blocked
- Process 2 requests tape drive 1 but is blocked

A set of process is deadlocked if each process in the set is waiting for an event that only another process in the set can cause.

A process request the resources, the resources are not available at that time, so the process enter in to the waiting state.

The requesting resources are held by another waiting process, both are in waiting state, this situation is said to be "Deadlock".

Necessary condition for deadlock.

- Deadlock system must satisfy the following 4 Conditions. Those are:
  - mutual exclusion
  - Hold and wait
  - No preemption
  - Circular wait

Resource allocation graph.

- Deadlocks can be described accurately using a directed graph called a system resource allocation graph.
- This graph consists of a set of vertices V and a set of edges E.
- The set of vertices V is partitioned into two different types of nodes:

$P = \{P_1, P_2, P_3 \ldots P_n\}$, the set of all active processes in the system.

$R = \{R_1, R_2 \ldots R_m\}$, the set of all resource types in the system.

The set of edges E is in two different types:

- A directed edge from process Pi to resource type Rj is denoted by $P_i \rightarrow R_j$, it called request edge. It means that process Pi requested an instance of resource type Rj & is waiting for that resource.

- A directed edge from resource type Rj to process Pi is denoted by $R_j \rightarrow P_i$, It is called assignment edge. It means that an instance of resource type Rj has been allocated to process Pi.

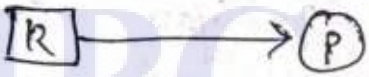
Resource allocation modeling using Graphs.

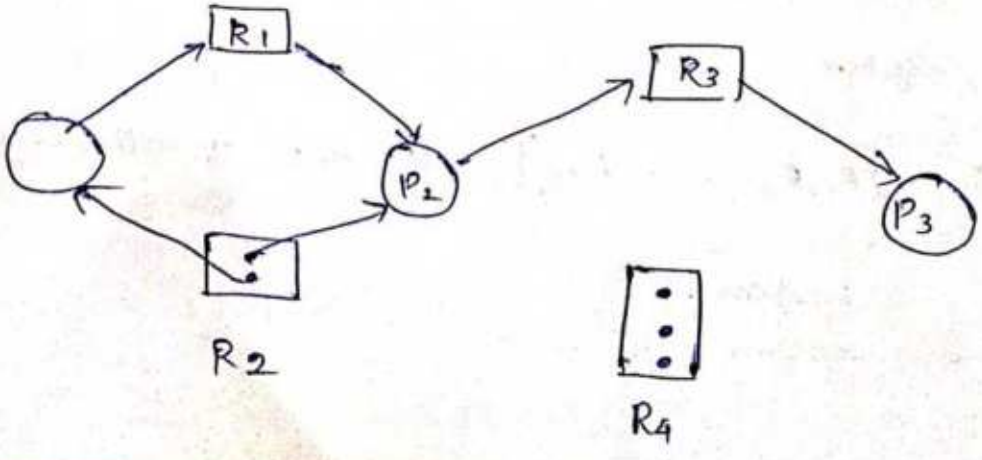
Nodes: resource [R]

Process (P)

Area: resource requested (request edge)

resource allocated: (assignment edge)





The sets of

Process $P = \{P_1, P_2, P_3\}$,

resource $R = \{R_1, R_2, R_3, R_4\}$,

Edges $E = \{P_1 \rightarrow R_1, P_2 \rightarrow R_3, R_1 \rightarrow P_2, R_2 \rightarrow P_2, R_2 \rightarrow R_1, R_3 \rightarrow P_3\}$

Resource instances - 1 instance of resource type R_1, 2 instances of resource type R_2, 1 instance of resource type R_3 and 3 instances of resource type R_4.

If a RAG contains no cycles, then no process in the system is deadlocked. If the graph contains a cycle, then deadlock may exist

Deadlock detection.

Some system grants the available resources to requesting process freely and then check for deadlock.

To detect deadlock, Resource Allocation Graph is construction including all processes and resources. If there exists a cycle, then deadlock is there.

Recovery from deadlock.

Once deadlock has been detected, some strategy is needed for recovery. The various approaches of recover from deadlock are:

- Recovery through preemption
- Recovery through rollback.
- Recovery through process termination.
- Abort all deadlocked process
- Abort one by one process until the deadlock is eliminated.

Deadlock avoidance

It is one of the methods of dynamically escaping from the deadlocks, the word dynamically means "online".

Safe state means "no deadlock will happen; even we allocate the resources to requesting processes".

Unsafe means the deadlocks may happen if grant the resources. Consider the below figure for better understanding.

Deadlock prevention:

Deadlock prevention is same as take the preventive methods before attacking the deadlock.

1. Mutual exclusion:
2. Hold and wait
3. No preemption
4. Circular wait.

UNIT-III

Memory Management

3.1 Memory management

3.2 Contiguous allocation.

i) Partitioned memory allocation

ii) Fixed & variable partitioning

iii) Swapping

iv) Relocation

v) Protection and Sharing

3.3 Non contiguous allocation

i) Page allocation

ii) Segmentation

iii) Virtual memory

3.1 Memory Management

Memory is 2'd major component in computer system after processors.

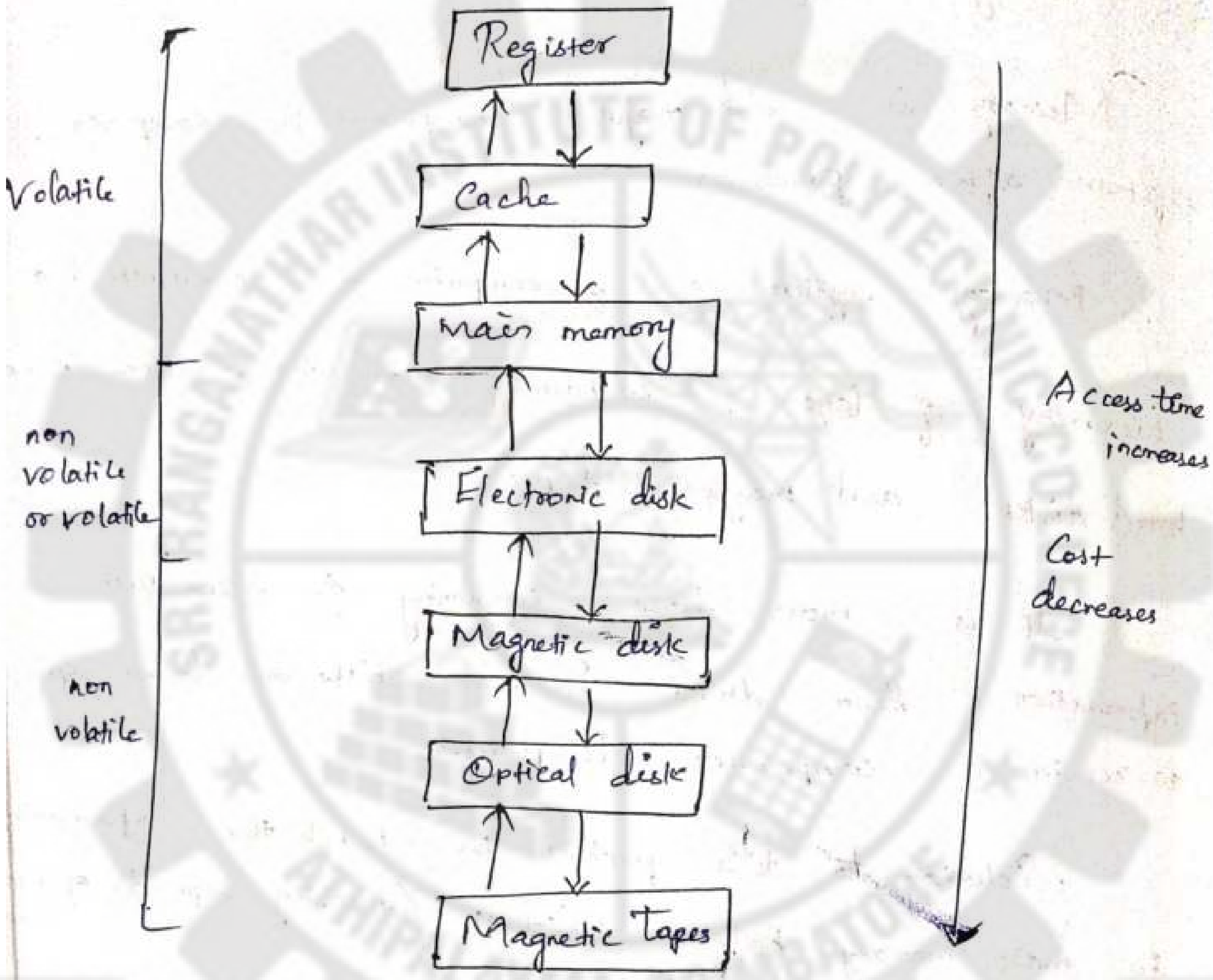
Memory system of a computer is constructed as hierarchy of layers containing registers, cache memory, hard disks and magnetic tapes

It is more time consuming to access such information from disks because disks are lower speed devices in comparison to processors.

Solution to this problem is Fetch the information in main memory before using it (accessing speed of main memory before using it).

Processors access this information from memory. This raises various issues like memory-allocation and de allocation, protection sharing etc.

So much memory management is needed.



Main memory:

It is work hose of the monory System. It is also known as physical memory.

Main memory is the form of chips internally in computer system.

It is volatile it means it needs to have electical power in order to maintain its information.

Random: Random means that any piece of date from this memory can be accessed quickly in constant time.

Main memory has limited storage capacity (64-512 MB now 1 GB.

Entire main memory consider as sequential list of bytes. Each byte has an address that is used to locate it this address is called physical address.

The addresses for most computer system starts at 0 and go up in sequence until each byte has an address.

Main memory

Processes:

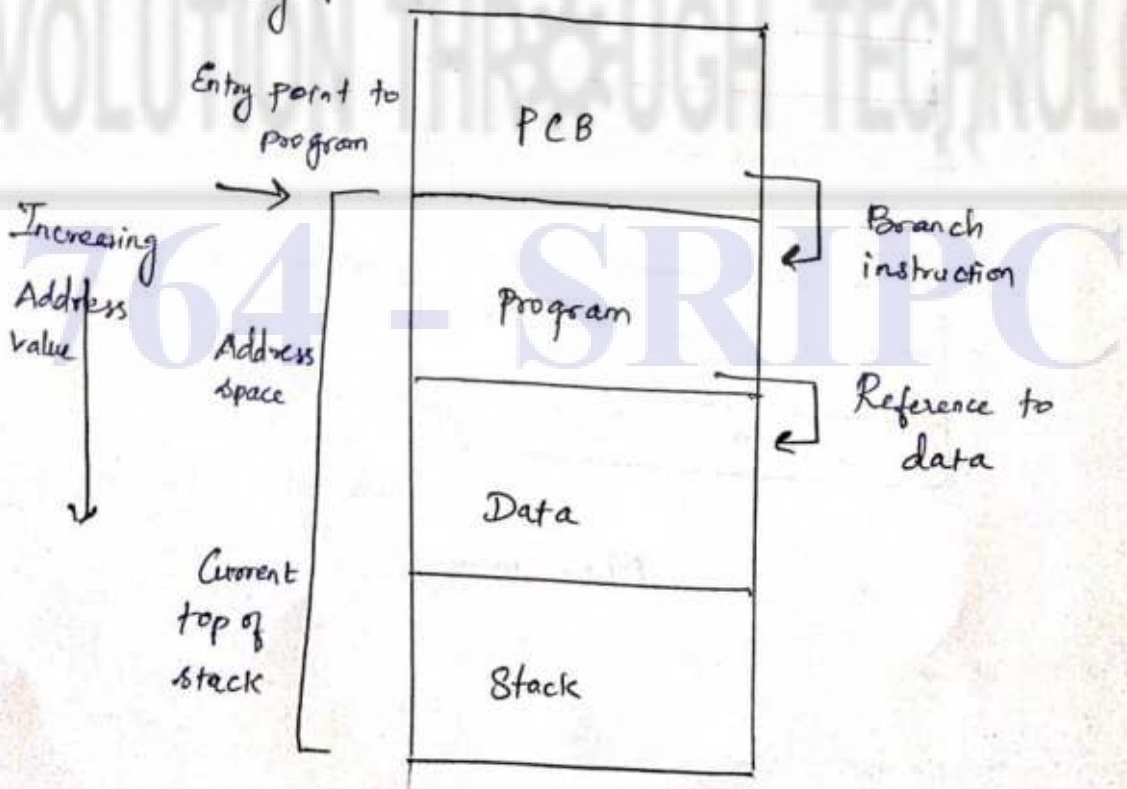
Process is instance of computer program that is being executed.

Process Contains program code, data and stack, the three parts consists an address space of the process

Address space is a sequential list of bytes, which is called logical address.

Depending upon the instruction and need, various data are used

For any process to be executed it should be in main memory.



Difference between logical address & physical address

Physical address allocated to the process may not exactly match the logical addresses of process so there should be some mapping between physical address and logical address, so memory manager is required.

Memory manager or management

MMU (Memory Management Unit) is a part of OS that deals with main memory.

MMU is memory management of main goal is to efficiently utilize the main memory among various simultaneously executing processes

Requirement and functionalities of MMU.

It performs operation like,

1. Fetch
2. Placement
3. Replacement
4. Sharing
5. Address Translation
6. Protection

Memory Allocation:

Memory should be allocated to various process and files as per requirements

When there is no need allocated memory should be freed.

While allocating memory two goals should be fulfil

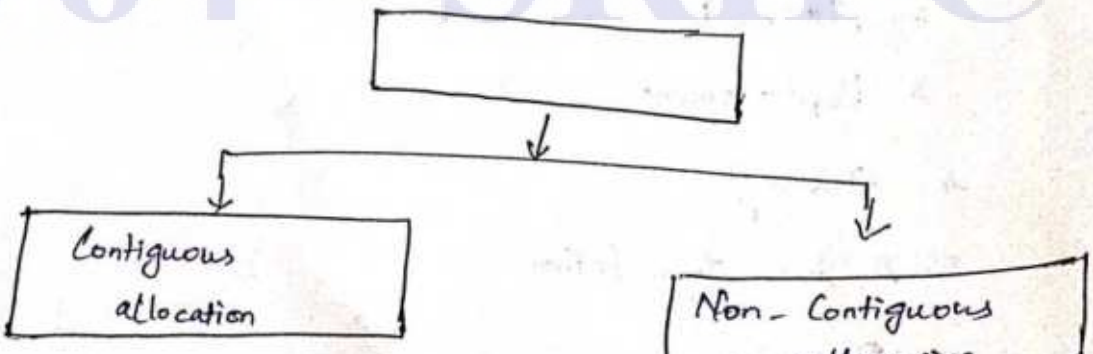
1. High Utilization:

- Maximum possible memory should be utilized.
- No any single process of memory should be wasted.

2. High Concurrency

Maximum processes should be in main memory.

When more and more processes in main memory, CPU will remain busy most of the time in executi one of the processes, resulting in better through put



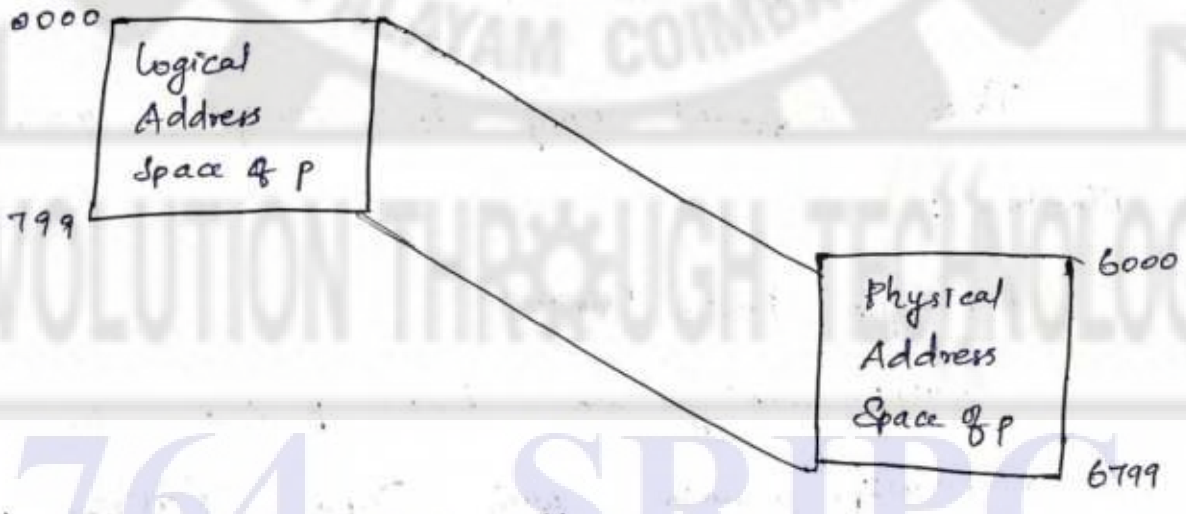
1. Single process monitor (SPM)
2. Multiprogramming with fixed partition.

1. Paging
2. Segmentation

Contiguous memory allocation.

- It is simple and old allocation method.
- It is not used in modern Operating System
- In this, each processes occupies a block of contiguous memory location in main memory.
- Entire processes kept together in contiguous section of memory.
- Here logical address space is not divided into any partition

Physical Address space will be contaguous without any gaps.

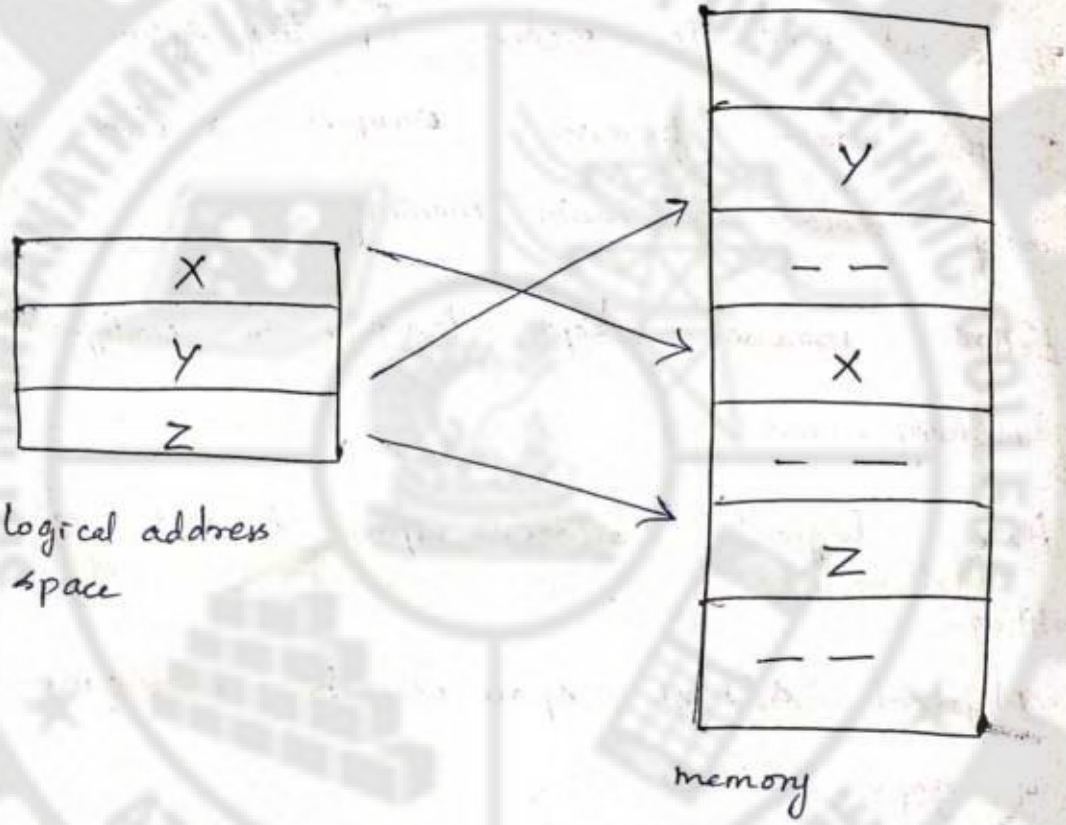


Non-contaguous memory allocation.

This is used by modern Operating system

Here logical address space of processes is divided into partition and for each partion contiguous chuck of free memory is allocated.

Physical Address space will not be contiguous now.



Contiguous memory allocation.

Three different ways for contigious memory allocation :

- Single Process Monitor
- Multiprogramming with fixed partition.
- Multiprogramming with Dynamic partition.

1- Partitioned memory allocation :

Single Process Monitor (SPM) :

. It is simplest possible memory management scheme

. Only one process is allowed to run. No more than one process can run simultaneously.

Only one process can run at a time

When user types a command the Operating System copies requested program from disk to memory and executes it.

When process finishes, the Operating system displays a prompt character and waits for a new command.

When Operating System loads new program in main memory, overwriting first one and execute it.

II. Multiprogramming with fixed (static) partition.

This method allows multiple processes to execute simultaneously

Memory is shared among Operating System and various simultaneously running processes.

Memory is divided into fixed partition size can be equal or unequal for different partition

Each partition is accommodating exactly one process.

There are two possible ways to implement method with a queue

1. Using multiple input queues
2. Using single input queue.

Using Multiple Input Queues.

For each partition separate queue is maintained

Whenever any process comes, it is put in a queue for smallest partition large enough to hold it.

When partition becomes free process from the front end of the queue is allocated that partition.

Using Single Input Queue.

Only single queue is maintained for all partition become free, process from queue is selected, this can be hold by that partition.

Process can be selected in two ways: Just select process which is near to front of the queue. Large partition will be wasted for small process.

Search engine queue and select largest process which can fit in that partition.

II. Multiprogramming with dynamic partition (variable partition):

Here memory is shared among Operating System and various simultaneously running processes.

Here memory is not divided into any fixed size partition. And number of partition is not fixed, process is allocated exactly as much memory as it requires.

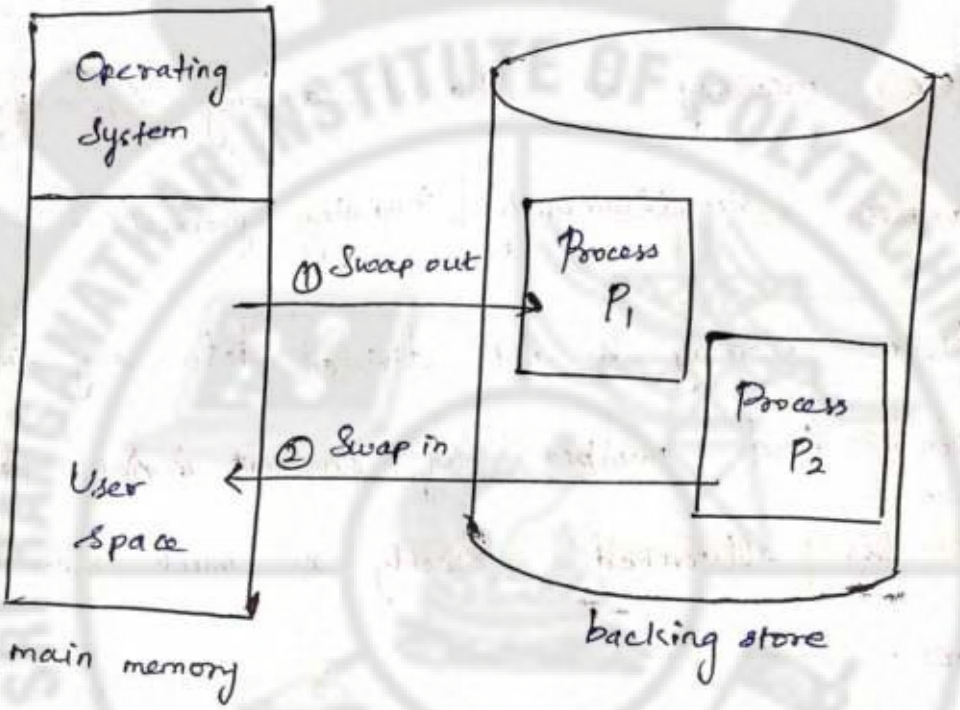
If enough free memory is not available to fit the process, process needs to wait until require memory becomes variable.

III. Swapping.

Swapping is a technique in which processes are moved between main memory and disk.

Swapping uses some of the portion of secondary storage as a backing store; this area is called swap area.

Operation of moving processes from swap area to main memory is swap-in.



iv. Memory Relocation

A process can loaded in any partition in main memory.

Address in logical address space and physical Address Space is not a same here.

Logical Address Space specifies the location of instructions and data within process address space.

Physical Address Space specifies actual location in main memory.

& logical Address (5) should be converted to actual Physical Address (1000+5) = 1005.

v. Memory Protection:

Multiprogramming allows running more than one process run simultaneously.

So memory is shared among processes as well Operating System.

Protection of both the problem:-

Two register are used. These register are called limit register and base register.

Limit register are used to store size of process.

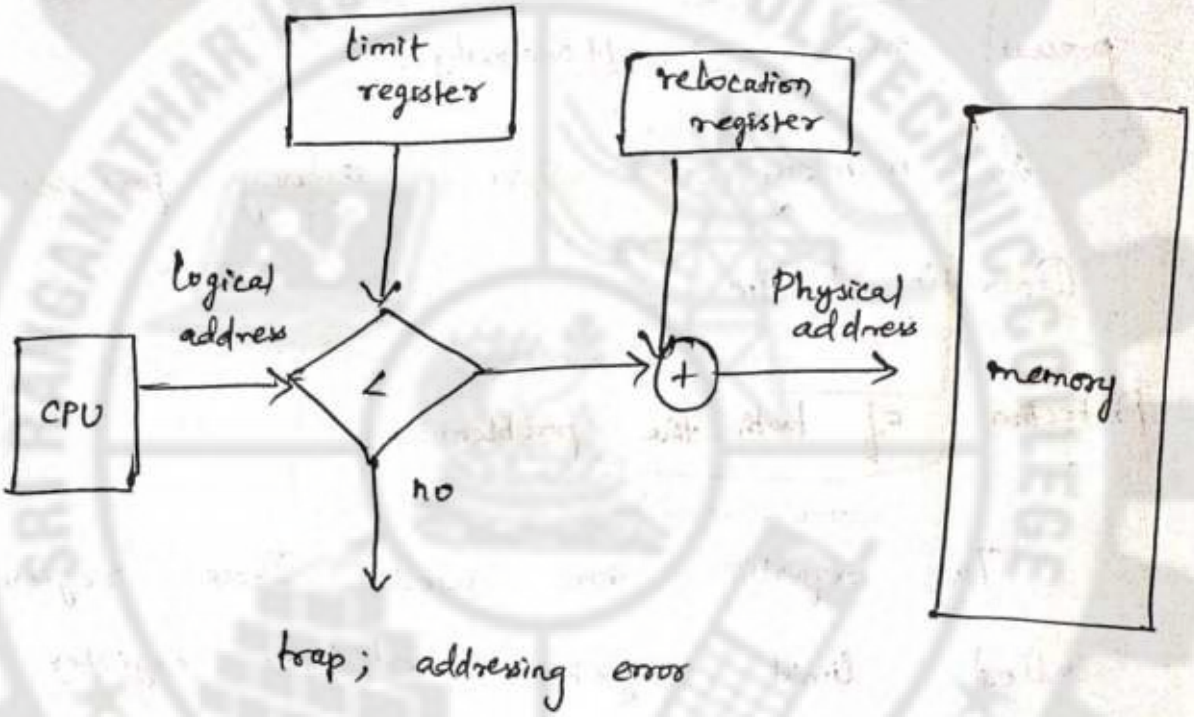
Base register are used to store starting location of process in main memory.

Whenever any process is loaded in main memory, its staring in base register and size stored in limit register.

CPU generates logical address start from 0 and goes up to size of process.

Thus in contiguous memory allocation, the problem of memory relocation and protection can be overcome by maintaining two registers.

limit register and base register:



Strategies to select partition:

Whenever any process enters in a system required free memory is allocated for that process.

All free memory, also called holes, is searched to

1) First fit
2) Next fit
3) Best fit
4) Worst fit

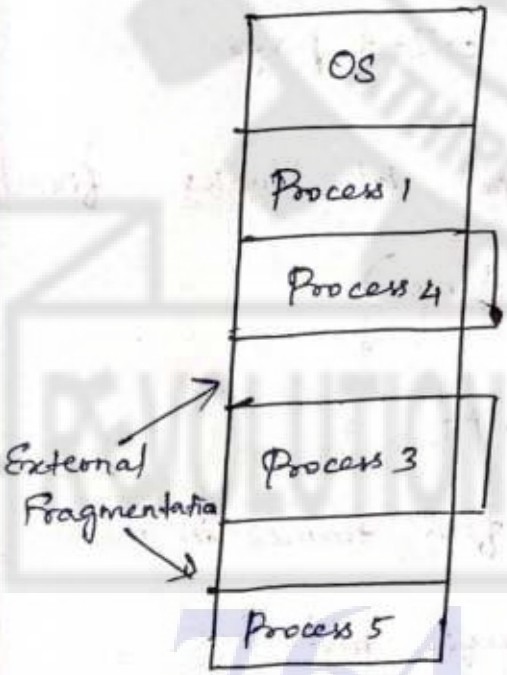
Fragmentation

Memory is allocated when process enters in the system and released when it terminates.

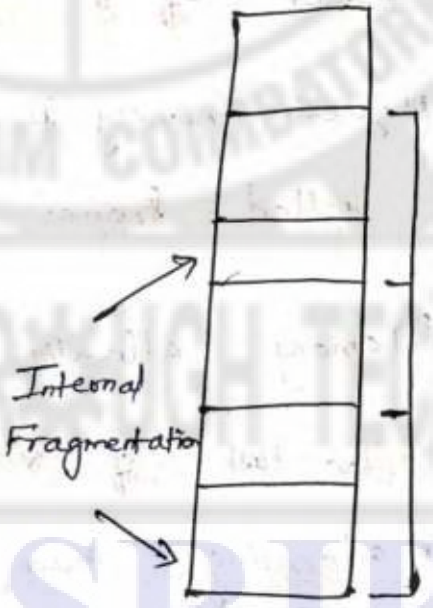
We can't utilize (100%) full memory due to some problems.

Two types of fragmentations:

- External Fragmentation
- Internal Fragmentation.



External Fragmentation



Internal Fragmentation.

Non - contiguous memory Allocation Technique.

Most modern operating system use this method for allocating memory.

For each partition contiguous chunk (group) of free memory is allocated in physical memory.

1) Paging allocation Technique

It is non - contiguous allocation technique

logical address space of a process is divided into blocks, of fixed size, called pages

physical memory is divided into blocks fixed size; called frames (page frames)

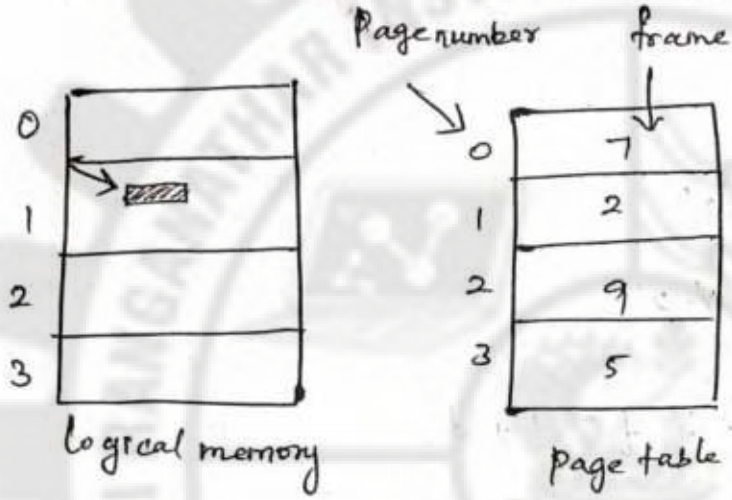
Memory allocation refers to,

- The task of finding free frames in memory,
- allocating them to pages and
- keep track of which frame belongs to which page.

Page table:

Operating system maintains a table, called page table for each process.

It is indexed by a page number.



Implementation of Paging technique.

Page number, which gives the number of a page

An offset, which gives the actual location within a page.

Page no (p)	offset (d)

Logical address (L):

Page table is used to implemant paging

Fraime no. indicats actual frame on physical memory in which page is stored.

Physical Address (PA):

frame no (f)	offset (d)

Address translation:

Page size and frame size $= 2^n$ bytes.

Logical address (L):

- Page no (p) : $L / 2^n$
- offset (d) : $L \% 2^n$

Physical address (P) : $f * 2^n + d$

Advantage of paging:

- Swap-in and swap-out operations are very fast.
- Here, available memory is divided in to fixed sized block and no wastage of memory between partitions, so there is no External fragmentation.

II. Segmentation

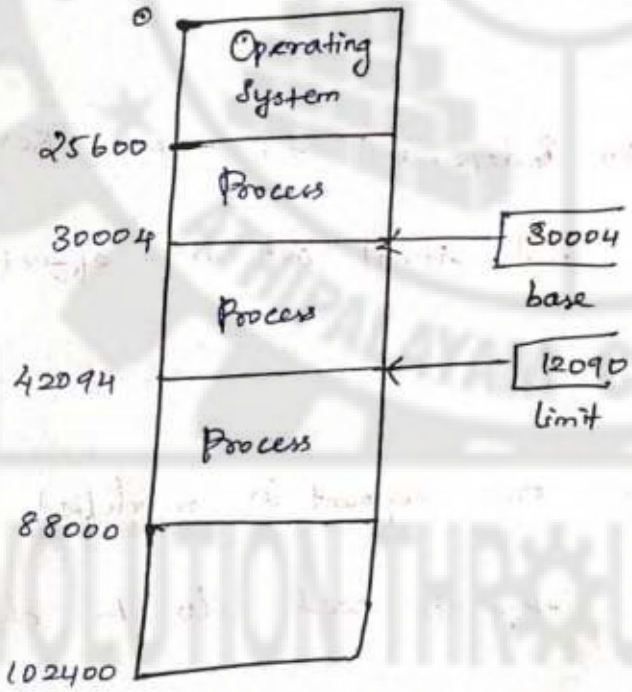
Logical address space of a process is divided into blocks of varying size, called segment. Each segment contains a logical unit of a process.

Logical unit can be main function, user defined function, procedure, stack, array, symbol table.

Each segment can be considered as a independent address space.

OS maintains segment table for each process. Segment table contains two information about each processs.

- Size of segment
- Offset, which gives actual location within segment.



base and limit of segment

Implementation of segment:

Segment table is used to implement segmentation. Each entry of the segment table has a segment base and segment limit.

A segment base contains the starting physical address where the segment resides in memory.

Logical address (LA)

During the process execution, a CPU generates a logical address (L) to access instruction or data from a particular location.

If offset is not within limit than illegal address error will be generated.

Advantage:

All segments are independent from each other, so segments can grow and shrink without affecting other segments.

If procedure in one segment is modified and recompiled, no other segments need to be changed or recompiled.

No internal fragmentation because segments are allocated exactly as much memory as required.

Disadvantage:

It is still expensive to allocate contiguous free memory to segments.

III Virtual memory

It is a technique that allows a process to execute even though it is partially loaded in main memory.

Virtual memory removes the requirement that an entire process should be in main memory for its execution.

Process size can be larger than the size of main memory.

Advantage:

Programs are not controlled by physical memory size. A process larger than the memory can also execute.

There is no need to keep an entire process in memory; more and more process can be contain in memory.

UNIT-4

File and disk management

Syllabus:

5.1 Overview of linux

5.2 Installation and upgrade

5.3 Introduction to shell and commands

5.4 Editing files with vi, vim, gedit, gcc

5.5 Linux Shell

4.1. File management

4.1.1 User view of file system

1.2 Attributes and Operations

1.3 File system Design

1.4 Disk Space

4.2 Directory Structure

4.3 Disk Organization.

3.1 Physical structure

3.2 Logical structure

3.3 Addressing.

4.4. Security And Protection Mechanism.

4.1.1 User view of file system.

File :

File is a named collection of related information which is stored in a secondary storage device.

It is used to store the information permanently.

File System :

A part of an operating system which deals with files is known as file system or file manager.

It manage all the file.

4.2.2 Attributes and operations.

Attributes :

1. Name
2. Identifier
3. Type
4. Location
5. Size
6. Protection
7. Usage Count
8. Time, date and User identification

Operations :

1. Create
2. Delete
3. Open
4. Close

5. Read
6. Write
7. Append
8. Seek
9. Get Attributes
10. Set Attributes
11. Rename

4.3.3. File System Design

- Byte Structure
  - Data stored as an unstructured sequence of bytes.
  - Any type of data can be stored.
- Record Sequence

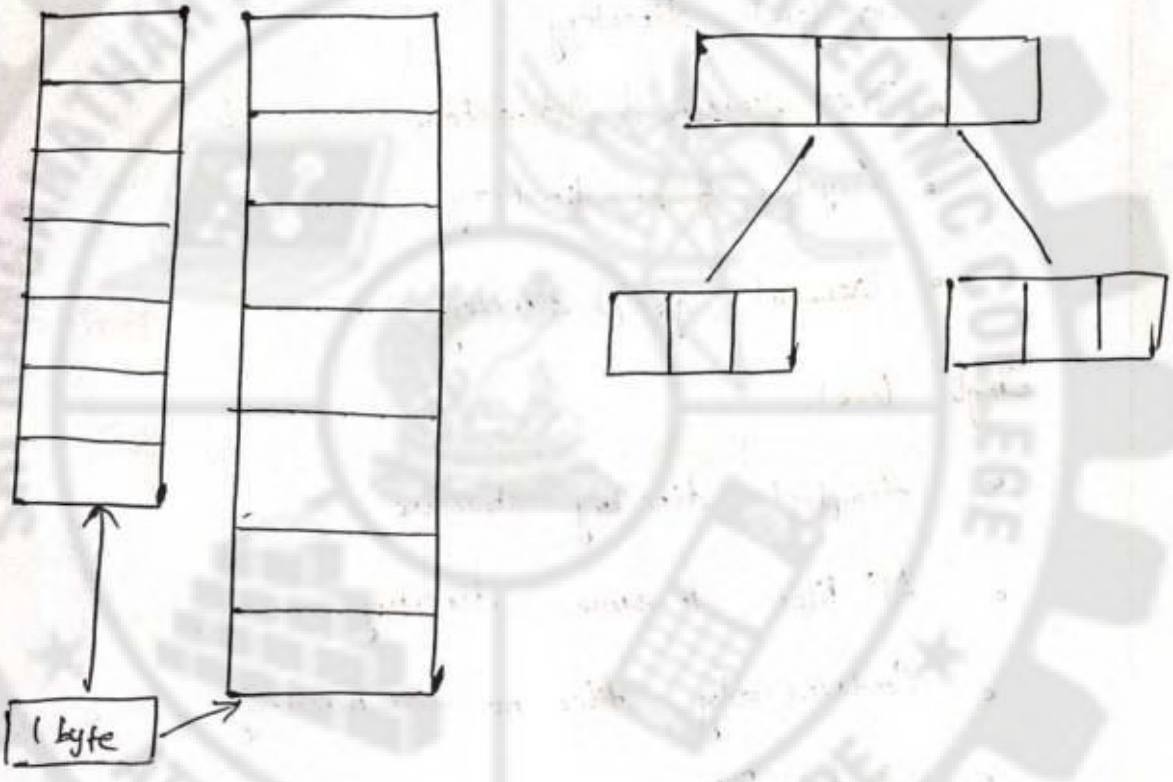
Data stored as a sequence of fixed length record has its own structure.

Suited for database related application.

- Tree

Data stored as a tree of records
Each record has its own structure
Used with large mainframes.

Byte Structure Tree structure.



4.1.4 Disk space

1) Regular File
2) Directories
3) Character special files
4) Block special files

4.2. Directory Systems

- What is directory?

Directory operation

- Create, delete, open, close, read, write, rename, link.

Directory structure

- Single level directory
- Two-level directory
- Tree structured directories
- Acyclic graph directory
- General graph directory

Single level.

- Simplest directory structure
- All files in same directory
- Contains only files no sub directory
- Used in Pc

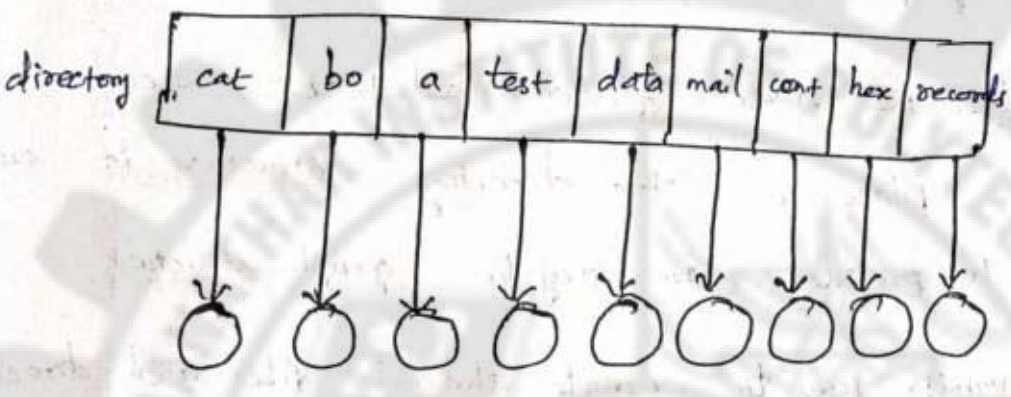
Advantage:

- Simple
- Easy to search

Disadvantage.

- Not suitable for multiuser system
- No. of files becomes too large
- Diff files cant be grouped.

Single Level:



Two level

Each user has private directory to avoid file collision.

Two level directory.

- Root directory
- User file directory.

Tree Structured :

- User can create their own sub directory
- Known as hierarchical system
- Each file has unique file path.

Advantage

- Avoid file collision
- Used multi users system
- Different files can be grouped.

Disadvantage :

- Sharing of files and directory is not possible

Acyclic graph.

When the same field need to be accessed in more than one place in the directory structure it can be useful to provide an acyclic graph structure.

Permits user to create shared files and director shared files and directory created using links.

Directory structure does not contain cycle.

Advantage

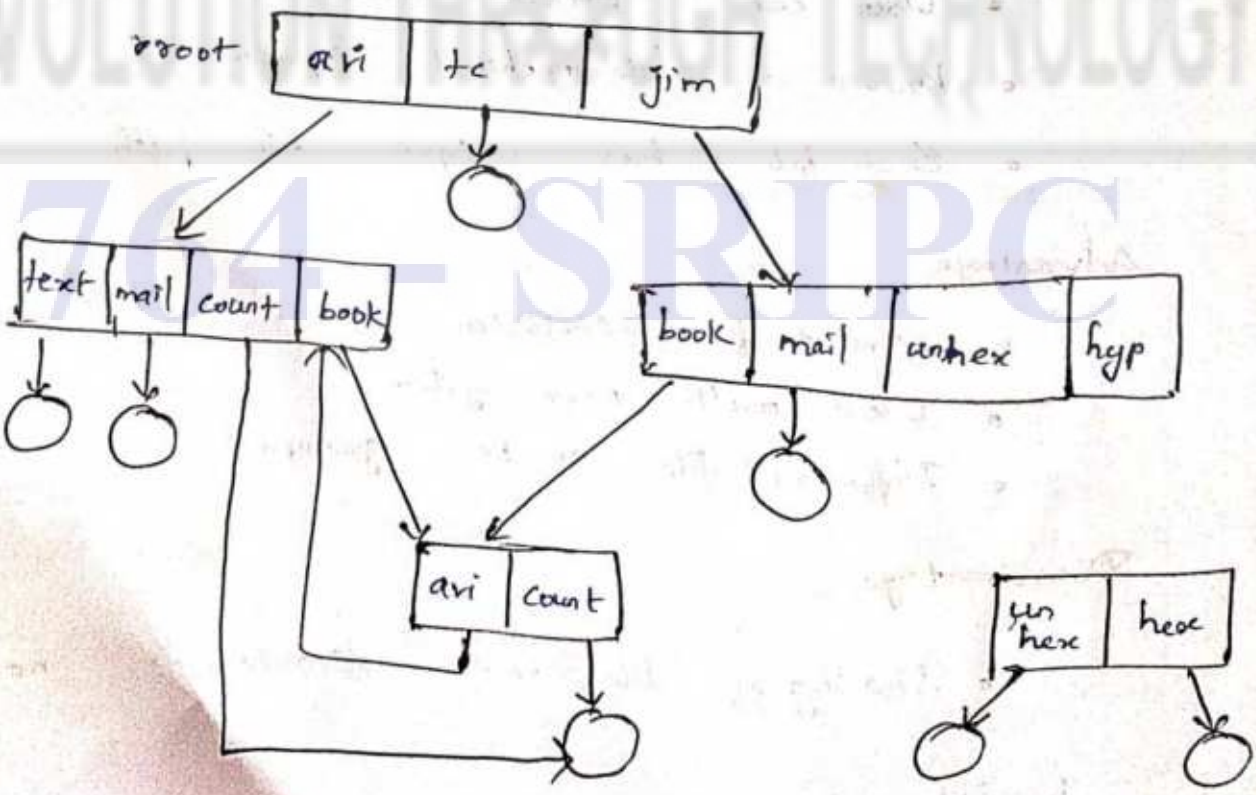
- Sharing of files and directory.

Disadvantage.

Deletion of files or dir. Is complex.

There are no cycle in directory

File may have multipath.



4.3.1 Physical Structure

Hard disk is a sealed circuit, it contains one or more platters.

Platters.

Each platters contains magnetic coating on both of its surfaces. Platters are stacked one on top of another.

Each platter contains two surfaces.

They rotate together round a central spindle.

Rotation speed is normally 3600, 5400 or 7200 rpm.

Disk contain 1 to 8 platters.

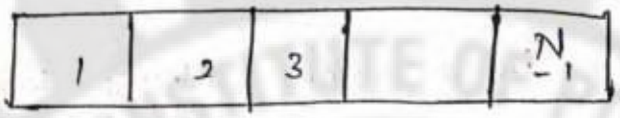
A group of tracks with the same radius are called cylinders. each track is on different platters.

Logical Address.

Disk is considered as large one dimensional array of fixed size logical block.

Size of blocks is varying from system to system.

Unique no. are used to uniquely identify on a disk.



Logical structure of disk.

4.3.3. Addressing.

To read and write to and from disk, OS tells the drive the required position.

The drive moves the head to the right position and either reads or write.

Address are given to identify particular sector.

CHS (Cylinder - Head - sector)

Cylinder are numbered from 0 to some maximum

Sector on each track are numbered from 0 to some maximum.

This method was used on earlier system.

LBA (Logical Block Addressing)

This needs method identifies sector by simply specifying the sector number

Each sector on disk is assigned a sequential number starting from 0.

Disk I/O

To read / write operation, OS needs to provide location (address).

Address is provided to hard disk controller.

If address is CHS, than there is no need of such type of translation.

Once location is identified, three steps are followed to perform read / write operation.

Hart till specified sector is directly comes above / below the read / write head. Time required for this operation is called rotational latency.

Read / write data.

4.4. Security and protection mechanism.

- Reliability

It means safety from physical damage.

Files may be deleted accidently.

- Protection.

Protection means safety from improper access.

- Password
- Access control.

UNIT-5

Linux Basic

Syllabus:

5.1 Overview of Linux

5.2 Installation and Upgrade

5.3 Introduction to shell and commands

5.4 Editing files with vi, vim, gedit, gcc

5.5 Linux Shell

5.1 Overview of Linux.

Linux was first developed by Linus torvalds

Linux is the most famous free and open source OS.

Open source OS are available in source code format rather than as compiled binary code.

There are many versions of Linux famous versions are,

Red hat linux → Fedora Centos

Debian → Ubuntu, Linux mint, Lingspire, PCLinux

Linux can be found on wide range of devices - from personal computers, mobiles, tablets and embedded systems to main frame computers and super computers.

5.2 Installation and Upgrade

There are four different types of linux installations.

1. Dual Boot

Computer System can have more than one operating system

User has to decide which operation system to boot into during boot process.

2. Live CD/DVD booting

- Linux can be run as a completely bootable operating system from cd/dvd.
- Operating system runs using those files.

3. Virtual Installation.

Linux can be run as a virtual machine inside another operating system.

Free Installation

It is the most popular installation. It installs linux as the only operating system of the computer.

Linux Architecture:

It is also known as the layered structure of linux.

1. Hardware:-

Bottom layer is hardware

Those devices provides various services.

Linux Operating System.

Next higher layer in Linux OS

It is called system kernel.

Kernel manages all the underlying hardware

It hides complex details of H/W

Main service of OS.

Standard library

Contains a set of procedures, one procedure per system call.

4. Standard utility program.

Program makes user tasks simpler.

User interacts with program, program interacts with OS to get services from OS.

Users:

Top most layers.

User program come in this layer.

They interacts with system either by library Procedure to involce system call or by using utility Program such as shell.

Kernel

Kernel is core part of any OS

Kernel manages all the underlying hardware

It directly interacts with the H/W, it hides complex details of H/W

When user needs to use any H/W, it has to use services (system call) provided by kernel.

5.3 Introduction to shell and commands.

Pwd- Print working directory

CD - Change directory

Mkdir- Make directory

Rmdir- Remove a directory or an entire directory tree.

Ls - Display the contents of a directory

Cat - To open the file, create a new file and append the data in files.

Copy- Places a copy of files in a different folder.

Directory related commands.

- % mkdir newdir
- % rmdir dirname
- % rm -r dirname

CD and PWD

Changes your current directory to new one.

% cd / some / other / dir

Absolute path

% cd subdir

% cd

% pwd

MAN

Manual pages

The first command to remember

To read about man itself type:

% man man.

NOTE: unfortunately there's no

% man woman ...

PASSWD

Change your login password

A very good idea after you got a new one.

Depending on a privilage, one can change user's and group passwords as well as real name, login shell, etc.

% man passwd

DATE

Guess what :-)

Displays dates in various formats

% date

% date -u

- in GMT

% man date

CAL

Calender

- for month
- entire year

- Years range : 1 - 9999
- No year 0
- Calender was corrected in 1752 - removed 11 days.
- % cal — current month
- % cal 2 2000 — Feb 2000, leap year
- % cal 2 2100 — not a leap year.
- % cal 2 2400 — leap year
- % cal 9 1752 — 11 days skipped
- % cal 0 — error
- % cal 2002 — whole year.

History :

- % cal — Current month
- % cal 2 2000 — Feb 2000, leap year
- % cal 2 2100 — not a leap year
- % cal 2 2400 — leap year
- % cal 9 1752 — 11 days skipped
- % cal 0 — error
- % cal 2002 — whole year

LS

Last directory contents

Has whole bunch of options, see man is for details

- % ls
- % ls -a
- % ls -A

CAT

- % cat
- % cat file1 [file2]...
- % cat > filename
- To exit cat or cat > filename type Ctrl+D to indicate EOF (End of file)

COPY

- % cp [options] <source> <destination>
- % cp file1 file2
- % cp file1 [file2]... /directory

MV

- % mv <source> <destination>
- % mv file1 dir/
- % mv file1 file2
- % mv file1 file2 dir/
- % mv dir1 dir2

RM

- % rm file1 [file2] ...
- % rm -r dir1 [dir2] ...
- % rm -r file1 dir1 dir2 file4

CHMOD

Changes file permissions

Possible invocations

% chmod 600 filename

-rw---1 user group 2785 feb 8 14.18 file name

% chmod u+rw file name

% chmod u+x myshellscript.

-rwx---1 user group 2785 feb 8 14:18 myshellscript

ECHO and GREP

- % echo "blah-foo" | grep blah
- % echo "blah-Foo" | grep zee
- See a separate grep tutorial.

PS

- Use ps to report the status of a proces
- /export/home/morris 07> ps

Output:

PID	TTY	TIME	CMD
19834	pts/7	0-05	bash
19954	pts/7	0.04	ps

SORT

- The sort command sorts data
- sort [-dfnr24] [infile...] [-o outfile]
- >sort name -o sorted_names
- or
- >sort names> sorted_names

KILL :

> If the process will not end use the '-9' or
> -KILL to kill the process immediately.
> >kill -919954
> >kill -KILL 19954

5.4 VI

- short for : visual editor
  available on all UNIX systems
- Original vi part of BSD Unix
  - written by BILL Joy to 1976
- man derived, improved version available
- Open source vim (vi improved) is part of GNU/Linux
- Vi has multiple modes of operation:
  - Input mode, command mode, last line mode.

VIM .

- Syntax : vim [options] [filename]
- Filename is optional
- With it, it opens that file for editing
- Without, it opens a default screen
- Many options available, most commonly used ones are for file recovery.

GEDIT

- It is similar to notepad editor of windows OS but it is more powerful than notepad.
- It is free software
- It is support multi language spell checking and a flexible plug in system.

GCC

- GCC refers to GNU compiler collection
  GCC is a compiler for c, c++, java and other programming languages.

  Compile the program by following code.

  Gcc test.c -o test.out

  Run the program by following commands

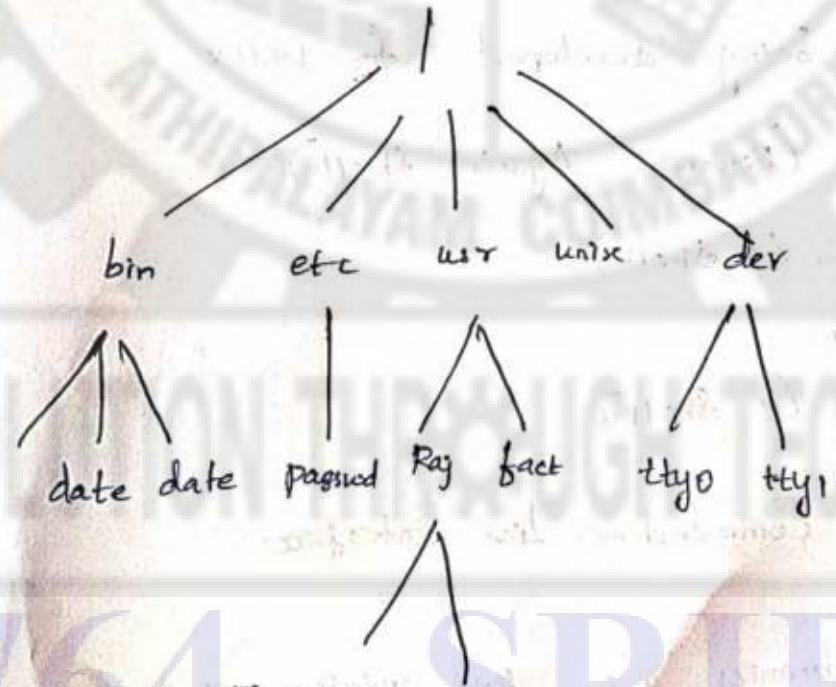
  ./test.out

Directory

It is a container for other files and sub-directories They provide a hierarchical file structure. They organize files in well manner.

Directory is a file whose data is sequence of entries. Each entry contains a file name and a unique identification number called I-node number.

Directory Structure



- Top of file is called root (defined by /)
- The name of file is given by path name. It describes location of a file and directory.

- Absolute path :
- C:\prog\cpp\test.cpp
- Relative path :
- ./test.cpp

5.5. Linux Shell

A command is a program which interacts with the kernel to provide the environment and perform the function called for by the user.

- Sh (Bourne Shell) The sh shell was the earliest shell, being developed for UNIX
- bash (Bourne-Again Shell)
- csh (c shell)
- tsch
- zsh (Z Shell)

Linux Command – Line Interface.

- Libraries are pre-written code "pieces" that application programmers use in their programs.
- Utilities maintaining the file system, editing text files managing running processes, and Installing new software packages.