Chapter 1
Introduction

1) Define Operating Systems. [2011, 2012, 2013, 2014, 2015, 2016]

Ans:- Operating Systems is a system software that manages software the software and hardware resources.

Ex: Windows, Linux, IOS

Software resources: - memory management, process, file management etc..

Hardware resources: - HDD, RAM, I/O devices.

2) Explain different (various) operating systems. [2011, 2012]

A) <u>Distributed System</u>:

- This allows numbers of computers at geographically separate locations, to work co-operatively. Data processing divided among different machines, they have own operating systems.

-This represents a configuration where in several work station PC's called intelligent notes(sites) are interconnected with mainframe and mini computers via communication network is called distributed computing environment.

<u>Major Objective of D.S</u>:

i) Resource sharing:- user at one site may be able to access the available resource of another site.

ii) Load sharing/balancing:- An overloaded sites computational tasks can be shifted to an executed by other remote sites having little or no work load at present.

iii) Reliable operations:- failure of one site need not affect other sites and remaining sites can continue operating.

iv) Communication:- processes executing at different sites can communicate to exchange any useful info.

B) <u>Real Time Distributed System</u>: [2011, 2012]

- The real time reefers to actual time in which on event takes place. Real time computer systems are loaded with RTOS, a real time O.S.

- Here time Bonds made between systems O.S.
- RTOS is designed to manage rigid and or précis time requirement of critical application.

-Real time Systems are required to manage time bound activities in application such as monitoring a satellite to be place in space orbit.

- Real systems are able to perform correct computations at given amount of time limits.
- Real systems are required to highly reliable for which they have to be compliant with fixed time restrictions.

EX: Railway Reservation systems, Rocket Launching.

3) Explain Different Operating System Services in details. [2013, 2014, 2015, 2016]

Ans:- i) Program execution(Run time execution):- Accomplish the task of loading a program into main memory partitions.

- Initiate program
- Provide for normal termination of program after successful execution.

ii) I/O operations:- Accomplish the task of device allocation and control i/o devices.

- provide for notifying errors, devices status etc.

iii) File system manipulation (Handling):- Accomplish the task of opening a file, closing the file etc and provide for creating, deleting the files.

iv) Communications:- Accomplish the task of inter-process communications either on the same or between different computer system on a computer network.

- Provide for massage passing and shared memory access in safe mode.

v) Accounting: - Accomplish the task of record keeping the system usage by how many users and for how long (duration) for billing and accounting purposes

- Maintains log system activities for performance analysis and error recovery.

vi) Error detection: - Accomplish the task of error detection and recovery if any. For instance paper jam on the printer.

- keep track of CPU, memory, I/O's device, storage device, file system networking etc..
- Report and/or deliver error massages in case of arithmetic overflow, diide by zero errors.

vii) Resource Allocation:- Accomplish the task of resource allocation to multiple jobs.

- Reclaim the allocated resource after their use or as and when the job terminates.

viii) Protecting the systems:- Provide for safe computing by employing security scheme against unauthorized access/ users.

- Authenticate legitimate users with login password and registrations.

4) Explain Components of Operating Systems details. [2011, 2012, 2015, 2016]

Ans:- i) Process management:- Process management refers to Process management. In a Process management process will be allocated to process.

- A process is running program, it may be in one of the three stage Ready, Running, Blocked
- A program (called Job or Task in O.S terminology) is a sequence of that are to be executed by CPU processor to yield desired output.

ii) Memory Management:- The major function operating system is to manage the resources of computers. Out of which main memory plays key role in the operation of a modern computers system.
- Main memory is repository of run-time programs and the relevant data being accessed and shared by CPU or processor I/O devices.

iii) File Management:- File is collection of related records about an entity is defined by a file creator, object program, txt data, error report etc.
ex :- a data file may sequence of characters that can be alphanumeric, alphabetic, pneumonic
- File management designed to provide an uniform log view.
- File management allow creation and deletion of a file.
- File management provides back-up a file for permanent storage for their feature use etc.

iv) I/O's System management:- It is also referred to as device management or simply device management. These devices include Keyboard, mouse, joystick, scanner etc.
- All these devices have can have different physical characteristics (analog, digital, magnetic)
- This are classified as 2 groups:- 1) I/O device 2) Secondary Storage device.
1) I/O device:- device drivers to control and/or to manage device characteristic
- I/O interrupt
- I/O traffic controller (program)
2) Secondary Storage:- this are like Magnetic tape, caridge etc. back-up the expensive data, main memory for additional storage requirements.
Various Functions:- → Allocation of Disk space
→ Disk scheduling (in case of share access storage devices)
→ Management of the disk space (avoiding bad block of storage space)

V) Network Management:- When computers are interconnected via communication links we have computer communication network like LAN , WAN and distributed systems as well.
- Network operating system (NOS) also manage networks routing, control data traffic.

Vi) Protecting Systems:- To manage the resources of a computer a operating system Also provides for protection mechanism.
- With multiprogramming time sharing systems might execute number of processes concurrently.
-Various functions of OS are:-
→ Provide a control access to the resource of a computer system.
→ Provide a mechanism for detecting hidden or dormant errors.
→ Provide protection schemes to distinguish between authorized and unauthorized access of the resource of P.C

Vii) Command Interpreter System:- The primitive unction of O.S is to support and user interface that determine the interaction with programmers operators or end-users.
- M.S DOS and UNIX O.S have the layer called **Shell.**

Types of User Interfaces:-
a)GUI (graphical user interface):- This are user interface commands are selected for execution by moving placing pointer on a tiny graphic images are called images.
- GUI are no need to remember and type the command line characters.
b) CUI (command line interface):- When a command are typed on keyboard and displayed on the screen at comment.
Ex: c:\> dir

5) Explain Operating System calls and services. [2011, 2012, 2014]
Ans:- The various types of System calls that can be offered by an O.S under each of these categories:

a) System calls for Process management:-
→ Create, Terminates a process
→ Load, Execute a process
→ End, Abort a process
→ Allocate, realise a process
→ Wait event, signal event

b) System calls for File management/ File manipulation:
→ Open, close a file
→Create, delete a file
→ Read, write a file
→ Get file, set a file

c) System calls for device management:
→ Request, realize an I/O device
→ Attach a device logically (mounting)
→ Read, / write device

d) System calls for Information maintains/management:
→ Get date, time, and system data.
→ Set date, time, system data
→ Get date, time, device attributes.

e) System calls for Communication management:
→ Create, delete connection
→ Send, receive massages
→ Transfer status information.

System services:

i) Language Translator: - + interpreters (for ex: java interrupts, basic interpreters)
→+ Compilers (for ex: turbo c compiler, Borland c++ compiler)
→ + Assembler (for ex: MASM, TASM, assembler for 8086 microprocessor)

ii) Loaders and Linkers:- for program loading execution. The system programs such as Absolute loaders, re-lockable loaders.

-Linkage, editors, linkers, overlay loaders help assets placing (loading) the assembled and/or compiler code (object program) into main memory and transfer CPU control to the first instructors at which the program execution has to begin.

iii) Text editors for file modification and file manipulation: - A system programs such as screen editors, line editors, Morton editors and several other text editors.

iv) Application program (system utilities for command user needs): - A system program has applications or utilities such as world processor (M.S world), web browser (net scope navigators, internet explorer (IE) M.S Excel).

- DBMS packages (ADD) for plotting.

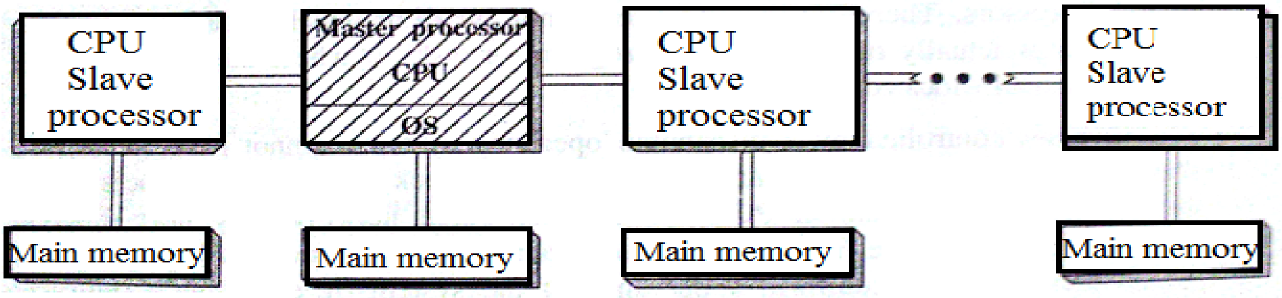
V) System programs for communication and status reporting: - Certain system program has been developed to encourage. Browsing a web page, remote computers login, remote file transfer, sending & receiving massages like e-mail facilities.

- Ex: UNIX has computing environment

6) What are Loosely Coupled and Tightly Coupled systems (Parallel systems)?
[2013]

Ans:- i) Loosely Coupled:- In this system each & every processors has separate memory (its own memory). The processors can communicate with each other, but individually processors can directly access its own 'Local Memory'.

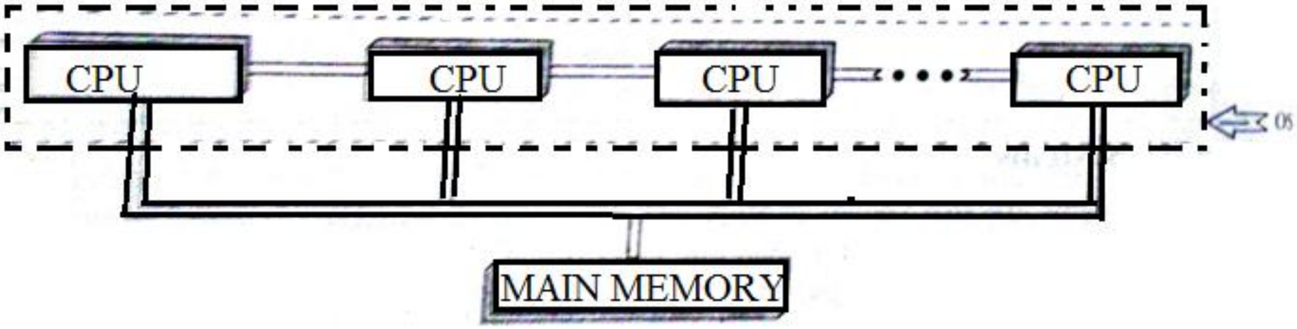
- Loosely Coupled systems employs asymmetric multiprocessing which each slave processor (CPU) is assigned a specific task and execute user job in parallel, where as a single master processor (CPU) control & co-ordinate activities of the slave processor.
- It is possible to assign several slave processor (CPU) to execute single user job in Parallel, in such case given job is broken into Modules.



Figure↑ Loosely-coupled Asymmertic Multiprocessing architecture of Parallel System

- Loosely Coupled systems can have a maximum up to 256 CPU's. Accessing there own local memory.

ii) Tightly Coupled systems: - in these systems, all processors shares a common memory and each processors can run identical copy of operating systems. Concurrently giving raises to Symmetric multiprocessing (SMP).
- With Symmetric multiprocessing, all processor (CPU) are peers (equal status) and there is no Master-slave relationship among multiprocessor.



Tightly Coupled System Multiprocessing architecture of Parallel System

- Even it can perform parallel execution of a single user job. Since processor shares common memory and executes simultaneously.
- Tightly Coupled parallel system can have a maximum of 16 pees CPU's accessing common memory, there are also know as shared memory parallel computers.

7) Explain concept of Multiprogramming and Time sharing.
Ans: - A Multiprogramming system permits several job (programs) to be executed simultaneously. In multiprogramming several jobs are executed.
- Multi programmed OS offers a convenient run-time environment and other support functions such that concurrent execution of jobs (programs).
- A single job system runs one user job at a time and performs serial processing.
- The general goal of multiprogramming is to make the most efficient use of computer system.
- Multiprogramming events makes OS to take certain decisions for the benefits of users and bring the visibility to end-users.

Time Sharing – It is the logical extension of the multiprogramming. It also refers Multi tasking. Its goal is to provide good response time to interactive sessions of users.
- A time shared OS offers interactive access time to numbers of end-users, simultaneously.
- Time Sharing support on line data processing and provides mechanism for concurrent execution of jobs
- It economically feasible due to several users interactive access and almost no wastage of CPU time.

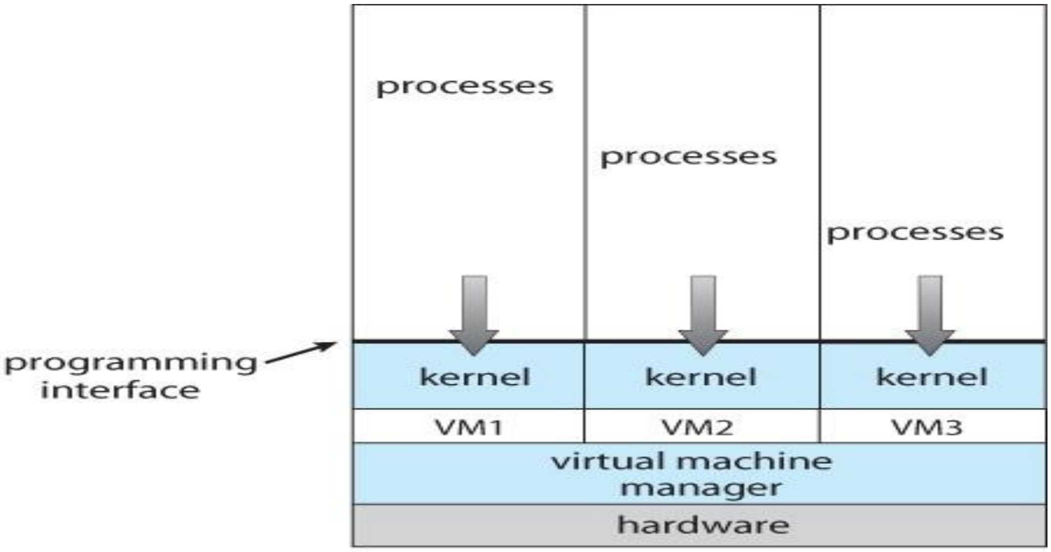
Functions Of time sharing

- it uses CPU scheduling and multiprogramming to provide each user with small portal of ime shared computer terminal on time – slice priority basis.
- it provides Multitasking and provides rapid user interactive with their terminals.
- Effective sharing and quick response
- High degree of user interaction.

8) What are Virtual Machines? Explain the benefits. [2013]

Ans:- The hierarchical structure of O.S can be extended to provide the end-user and operating system , with an illusion of running on separate extended machines. This extended machine is sometimes referred to as Virtual Machines. In essence, the resource of the computer system is shared to create virtual machine(VM's).

- A layer of VM OS called virtual machines monitor(VMM) exist & encapsulate real machine hardware in association with kernel.



Advantages / Benefits of Virtual Machines.

1) Using separate virtual machine provides higher degree of protection of various resource of system.
2) Virtual Machine allows systems development (R and D) activities to take place without affecting the normal system operation.
3) Different operating systems can run concurrently to serve the varying needs of normal system operation.
4) Example of virtual machine OS is IBM's VM O.S.
5) This concept gives rise to flexibilities.

Chapter - 2
Process Management

1) Define a process. [2011, 2014, 2015]

Ans:- The process can be defined program in execution. Sometimes it referred to as Task. It represents **user job.**

- The process corresponds to more synthesized version of job in OS environment.

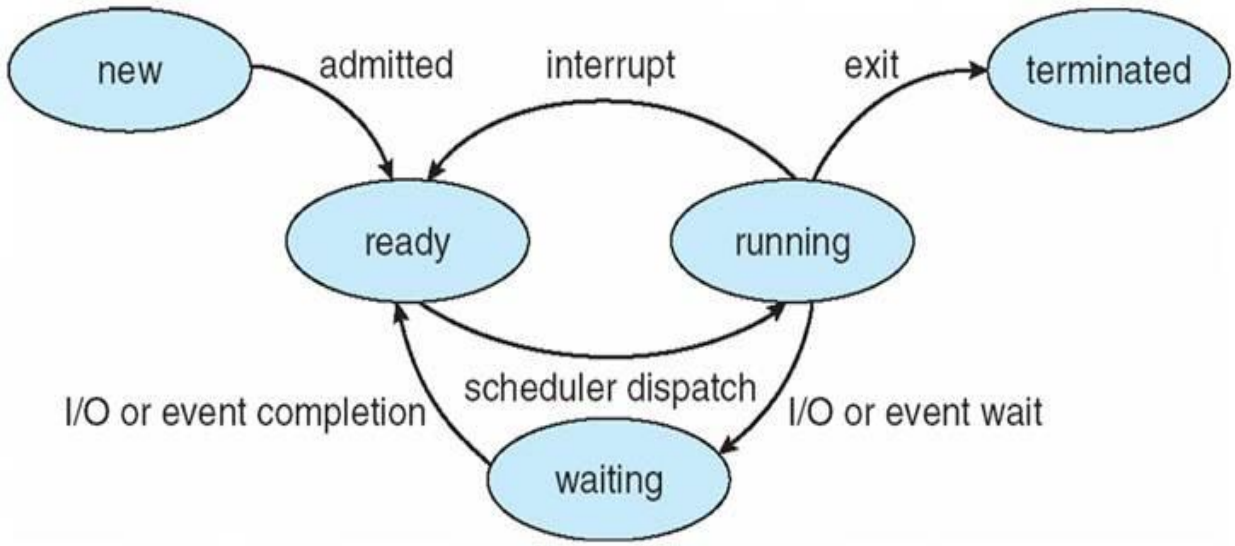
- Since process is always active or dynamic as it represents running state of program or user job.

- Here an active entity resides inside the Main Memory and finite number of sequence of steps in execution.

- It can in any one of the Ready, Running and Block states.

2) Explain the process states and operations of processes. [2011, 2014, 2015, 2016]

Process States :



1) Ready 2) Running 3) Terminated 4) Blocked or Waiting.

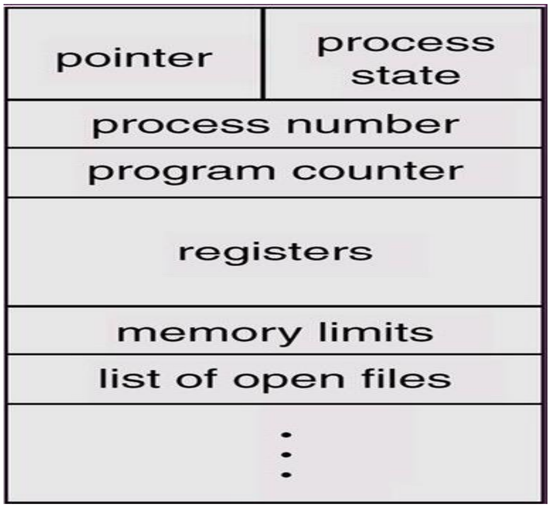
1) Ready:- The process is on the toes control over the CPU or Processer. That is, a ready process is waiting to be assigned to a processor for continuing its execution.

2) Running:- The process actually executing its instruction by CPU or processor.

3) Terminated:- The process has completely executed all instructions.

4) Blocked or Waiting: - The process must wait for some event to occur.

3) Explain in details about PCB (process control block). [2014, 2015, 2016]
Ans:-



A process switches between the running, ready and blocked states, many times, well before its formal termination.
Each time when a particular process leaves the running state and yet does not encounter its flag end, then its current state of status must be saved for feature reference.
a) Process state:- indication of the process state as Ready, Running, blocked or waiting etc..
b) Process number:- Suggest finds this process distinctly as process ID
c) Program counter:- it is register containing address of next instruction to be executed of this Process.
d) Registers:- A group of registers viz accumulators (A or AC) index registers, stack pointers general purpose registers and condition codes.
e) CPU Scheduling information: - It suggests the details about the priority of the process, pointers to ready queue scheduling.
f) Memory Management Information: - It suggests the details about Memory Usage by this process.
g) Accounting Information:- it suggest the details such as, the allocated CPU time-slice, the real-time used, process numbers, time limits and account number etc…
h) I/O device status information: - it suggest the details about allocated devices for this process, list of files opened for this process etc..

4) What is Scheduler? Explain different types of Schedulers.[2013, 2014, 2015, 2016]
Ans:- A processes Transit between various scheduling queues (job queue) while switching across the running, ready and waiting (blocked) states.
- It is the job of process management component of OS to select an appropriate process from these queues (called **scheduling**). And transfer the CPU control to such selected process (called Dispatching).

- The sole responsibility of selecting a process (for execution) in accordance with some scheduling policy-is being assigned to an OS module or to a component of process management called the Scheduler.

3 types of scheduler programs:

1) Long-term scheduler (Job scheduler)
2) Short-term scheduler (CPU scheduler or process scheduler).
3) Medium-term scheduler

1) Long-term scheduler:- Is also referred to as Job Scheduler. Long-term scheduler loads every time a new job or a program (from a hard disk) into main memory for execution. The frequency of loading programs depends heavily on the availability of space in main memory.
-In other words, Long-term scheduling carry out the selection of processes that couv turn be allowed to contend for processor assignment.

2) CPU-bound process Vs 1/0-bound process: - A CPU-bound process has minimal I/O request instructions, so that the process spends its allocated time-slice almost completely with the physical processor in order to carry out the numerical computations.
-Where as an I/O-bound process has maximum I/O request instructions so that the process spends more of its allocated time-slice in doing I/O operations rather than carrying out computations rarely with the physical processor.

3) Short-term Scheduler:- is also referred to as CPU scheduler or process scheduler, The short-term scheduler actually selects an appropriate process from among the already loaded processes in main memory, for execution and allocates the CPU.

4) Medium-term Scheduler:- give rise to an intermediate-level of scheduling in case of time-sharing systems like UNIX OS based systems.
-Accordingly, medium-term schedulers swaps-out (to send back to hard disks) some partially executed processes from main memory partitions
- Swapping eliminate overhead introduced due to high degree of multiprogramming and allow for good mixture of CPU-bound and I/0 bound processes in a freed main memory.

5) What are Co-operating Process? Explain briefly. [2011, 2012, 2014, 2016]
Ans: - The processes executing in the operating system environment can be classified into two types.
1. Independent process 2. Cooperating process
1) Independent process:- It does not share any data with any other process and it will neither affect nor be affected by any other executing processes in the system.

2) Co-operating process: - It share data with other processes and can directly share logical address space in Memory.

-Cooperating processes communicate each other via two communication schemes shared memory and message systems.

Co-operating Processes extend the following benefits:

1) Information sharing:-several users can access a shared file simultaneously.
2) Computational Speed Up: Parallel execution of sub-tasks in Multiprocessor systems.
3) Modularity:- Modular design to system functions as distinct processes or separate threads.
4) Convenience:- User can perform at his/her convenience the functions such as editing a file, printing a file, program compilation etc., in an overlapped manner.

6) Explain Inter-process Communication? [2011, 2012, 2013, 2014, 2016]

Ans:- Co-operating processes must communicate each other. For which the operating system provide a mechanism called **Inter- process communication facility (IPC)**.
- The IPC mechanism allows co-operating processes to communicate each of other and synchronize their actions without sharing the same address space.

Two Types:

1) Shared Memory System: - In this scheme, cooperating processes exchange information via shared memory variables and the OS provides shared memory.
2) Message System:- In this scheme, cooperating processes communicate each other by exchanging messages (message passing system). In this case, the OS has to bear the responsibility of providing necessary communication among cooperating processes.
- IPC scheme provides: quite helpful in distributed computing environment whey in the communicating processes are spread over across geographically distributed computers (intelligent nodes) and connected over network links.
Also, IPC mechanism is best supported by both message systems and message-passing system.

7) Explain CPU scheduling Criteria. [2011, 2012, 2015]

Ans: - CPU scheduling in general depends upon the system of priorities.
a) CPU Utilization: The primary objective is to keep CPU or processor the busiest one among all other resources of a computer system.
- The idle time of CPU can be minimized to almost zero in case of multiprogramming and Time-sharing systems.
- Typically, CPU Utilization may range from 0% to 100%. But in most practical cases it would be between 40% to 90% only.

b) Throughput: - Suggests that CPU must perform maximum computational tasks in the shortest interval of time.
- If the CPU bound process is complex and time-consuming then the system throughput may be just 1 or 2 process completion per unit of time.
- However, throughput rate may increase up to 10 or 20 process completion per unit of time, provided the processes involved in execution must be simple and short.

c) <u>Turn around Time</u>:- Defines the time elapsed between the time of submission of a process or job by a user and the time of completion of that process of job.
- Turn around time also includes waiting time in the ready queue, time spent over I/O completion, waiting to get into memory, waiting time for child process completion etc.,

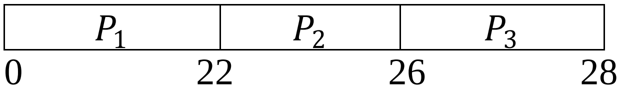
d) <u>Response Time</u>: - Defines the time it takes to commence responding to a request or a command. It's not the time interval to complete the task from the point of submission.
- The best criterion for an interactive time-sharing system is to minimize expected response time.

e) <u>Waiting Time</u>: - Suggest the time spent while waiting for I/O completion, waiting for CPU in ready queue etc., The sum total of such time quantum's expended while waiting for various subsidiary events to occur or to complete contribute towards increasing Turn around time.

8) Explain Scheduling Algorithms in brief. [2011, 2012, 2015]

Ans: - CPU scheduling or interchangeably Process scheduling is the activity that govern "switching CPU control" among various competing processes in accordance with some scheduling policies.
- The scheduling algorithms deals with the problem of choosing one process at a time from ready queue, so as to allocate the CPU to it, based on certain considerations such as shortest job, first-come, first-serve, round robin, system of priorities.
- We shall list out few scheduling algorithms to h discussed as follows:

a) <u>FCFS - First-Come, First-Served Scheduling</u>:- Name itself indicates, the process which comes first or which is the first comes the head of ready queue, will be allocated the CPU.
- Whenever a process enters the ready queue requests the CPU, Its PCB is linked at the end or tail of the queue.
- Assuming the process entering the ready queue in the sequential order p1,p2,p3… and are allocated CPU in the FCFS order.



- So the gantt chart suggest that waiting time is 0 milliseconds.

- So, average waiting time is= $\frac{48}{3}$ = 16 milliseconds.

b) SJF Shortest Job First Scheduling: - Name itself indicates, the process in ready queue having the 'Shortest next CPU burst time' will be allocated the CPU.

- This would help in selecting shortest job for allocating the CPU next, in case of long-term job scheduling in a batch system.
- Although the process enters in sequential order p1, p2, p3, p4… in like FCFS scheduling, the process will be executed in the order p4, p1, p3, p2…
- Now, respective waiting time for individual process in ready queue waiting to get CPU assignment shown below.

P_4	P_1	P_3	P_2

0 2 8 16 26

- Average waiting time is = $\frac{0+2+8+16}{4}$ == 6.5 milliseconds.

c) PR - Priority Scheduling: - Name itself indicates, the participating processes will be assign priority and each time process with highest priority will be allocated to CPU.
- Assignment as per FCFS scheduling.
- Accordingly , the respective waiting times for individual process in ready queue waiting to get CPU assignment shown using Gantt chart as follows:-

PR:1	PR:2	PR:3	PR:4	PR:5
P_3	P_2	P_1	P_5	P_4

0 2 6 11 19
22

- Now, average waiting time is = 0+2+6+11+19/ 5 = 7.6 milliseconds

<u>Conclusion</u>
- Priority scheduling can be implemented as either primitive or non-primitive algorithms
- Priority cab be assigned can be defined as either internal or external priority

d) <u>RR - Round -Robin Scheduling</u>: [2014]
- Name itself indicates, the Round Robin Scheduling suggest the CPU scheduler to go round the ready queue and allocate CPU to each process on FIFO basis.
- Working of Round Robin Scheduling:- The Round Robin Scheduling algorithm is pre-emptive. Accordingly, if the currently running process has longer CPU burst time exceeding time quantum.

12-5=7 0 0

P_1	P_2	P_3	P_1	P_1	P_1

0 5 9 12 17 22
27

So, average waiting time for RR scheduling tine is $\frac{21}{3}$ =7milliseconds.

Advantages of RR scheduling: - 1) For time sharing systems and multi user system.
2) Offers good response and turnaround time.

e) MLQ - Multilevel Queue Scheduling: [2013, 2015]

- MLQ scheduling partitions are ready queue of processers into many different queues based on certain considerations, such as Foreground or Interactive processors and Batch processors.

- MLQ scheduling algorithms permits different scheduling schemes to be employed for several different classes of processors.
- Further MLQ scheduling scheme permits among these different ready queues and is implemented by 'Fixed- priority preemptive' scheduling algorithm.

Advantages of MLQ scheduling algorithms:-1) Low scheduling overhead
2) Fixed priority scheduling
3) Each queue has its own scheduling algorithm

f) MLFQ - Multilevel Feedback Queue scheduling: - This allows processes to switch from one queue type to other.

- A process belonging to higher priority queue. Starts consuming much of CPU time, then it can be shifted down to lower-priority queue.

-Number of queues and type of scheduling algorithms used for each queue type

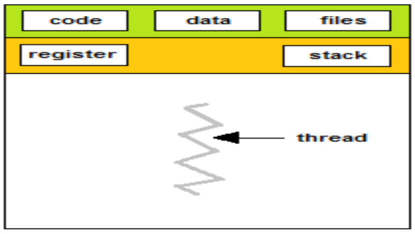
- Method used to suggest when to upgrade a low-priority process to higher priority queue.

Advantages: - 1) Most flexible CPU scheduling algorithm.
2) It permits processors to move between queues belonging to various priorities.

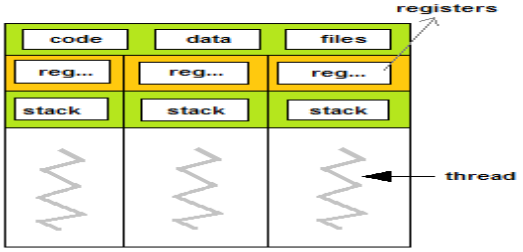
9) What is Threads? [2015]

Ans: - A thread represent small execution code segment associated with a process.

- Thread is a light-weight process (LWP). In its primitive form a thread represent basic unit of CPU utilization.
- There is one program counter, and one sequence of instructions that can be carried out at any given time.
- As each thread has its own independent resource for process execution, multiple processes can be executed parallel by increasing number of threads.



single-threaded process



multithreaded process

Advantages of Threads

- **Responsiveness:** - Speedy response to users.
- **Resource sharing:** - hence allowing better utilization of resources.
- **Economy:** - Creating and managing threads becomes easier.
- **Scalability:** - One thread runs on one CPU. In Multithreaded processes, threads can be distributed over a series of processors to scale.

- Multithreading Models

The user threads must be mapped to kernel threads, by one of the following strategies.

a) Many-To-One Model: -In the many-to-one model, many user-level threads are all mapped onto a single kernel thread.
-Thread management is handled by the thread library in user space, which is efficient in nature.

b) One-To-One Model: -The one-to-one model creates a separate kernel thread to handle each and every user thread.
-Most implementations of this model place a limit on how many threads can be created.

c) Many-To-Many Model: -The many-to-many model multiplexes any number of user threads onto an equal or smaller number of kernel threads, combining the best features of the one-to-one and many-to-one models.
-Users can create any number of the threads.

10) Explain Context Switching Concept in detail. [2012, 2013]

Ans:- CPU is about to be switched between processes, it is necessary to save the status information of currently running process soon after it is blocked for I/0 or its allocated time-slice expires.
- Thus, once after saving the status of previously running process in its corresponding PCB, the dispatcher then switches the CPU control by executing the LPS (Load Process State) instruction to load the status information saved in the PCB corresponding to the process selected (and dispatched) a fresh for execution.
- This task of saving the process state (PCB) of the old process and reloading the previously saved process sate (PCB) of the new process being scheduled for execution-is termed as
'Context Switch' or Switching Context.

Chapter -3
Process Synchronization and Deadlocks

1) What Race Condition? Explain two process solutions. [2013, 2015]

Ans: A race condition occurs whenever two threads access a shared variable at the same time.

- **Race Condition** is a situation on concurrent programming where two concurrent threads or **processes** & the resulting final state depends on who gets the recourses first.

The two process solutions are:- a) Critical section problem
b) Semaphores

2) Explain Critical Section Problem? Explain requirements to solve Critical Section Problem. [2012, 2014]

Ans: - Assuming the given system contains 'n' numbers of ready processors waiting to be assigned to CPU. For instance

S(Pi)= {P0,P1,P2,….Pn}

- The process shares the data & are said to be co-operating sequential process.

a) Entry section b) Exit section c) Remainder section

- Th**e segment** of code called Entry Section of process Pi must request the OS permission to enter in its own critical section.
- The segment of code called critical section perform the critical activities such. as accessing and/or modifying the shared global variables, writing and/or rewriting, appending, updating a file, a table of values etc.
- The segment of code called Exit section of process Pi follows critical section and concludes formal termination of critical section code (zone).
- The segment of code called Remainder section of process.
- The critical-section problem suggests that the critical section code of a process must be in use.

<u>Requirements to be fulfilled for providing solution to critical-section problem</u>:

- Mutual Exclusion: - Two processes (say P1 and P2) are said to be mutually exclusive.
-If execution of them (both P1 and P2) at the same time will not affect each other's address space-code section.

- Progress:- Assuming that if processes are not executing both in critical and remainder sections, then these processes can compete as to which one will enter its critical section next (Process Execution-progress requirement).

3) Explain Semaphores? [2013, 2014, 2015, 2016]

Ans:- A more generalized solution to the problems of synchronization and critical section problems can be given with the help of a synchronization toil called **Semaphores**.

- A semaphore S is a synchronizing variable that can hold an integer value. Accordingly, it can be initialized to a specific integer.
- A semaphore as a control variable can be accessed via two standard atomic operations.
- Each of these two operations accepts a single argument S.

<u>Wait (s)</u>:-This operation is originally termed as P (for wait -i+ to test) and can be defined with the following program segment.

```
Wait (S)
{
While(S<=0)
;
s--;  }
```

signal (S) :- This operation is originally termed as V (for signal → to increment) and can be defined with following program segment.

```
signal(s)
{
s=s+1; increment
}
```

- **Semaphores** doesn't permit the simultaneous modification of semaphore value S by more than one process, at a time
- Therefore two or more concurrently executing processes can be synchronized by sharing a common semaphore variable.

4) What is Critical Region? Explain Readers and writers problem.[2011, 2012, 2013]

Ans: - Critical Regions: - Critical regions are one of the fundamental high level synchronization construct, sometimes referred to as **'Conditional Critical Region'.**

- The critical regions construct eliminate simple errors that occur due to incorrect use of semaphores as a solutions to Critical-Section Problem.

- For instance when semaphores are used to provide solution to the critical section problem it is observed in general, the following requirements.

a) All concurrent processes share a semaphore variable mutex

b) Each & every process must execute wait (mutex) well before entering critical section and signal (mutex) thereafter.

```
do
wait (mutex);
…….
critical section
…….
```

signal (mutex)
- Consider the following situation:-
- Observe that order of execution of wait and signal operations on the semaphore variable mutex has been interchanged in a process and the execution sequence look like the one shown below :

```
 Signal(mutex)
……..
critical section
…
wait (mutex);
```

Readers and Writers Problem:

- The Readers and Writers problem highlights concurrent execution of both Reader processes and writer processes by sharing a common data file.
- Accordingly, the reading processes called Readers in short, can access the shared data file only for Reading.
- Where as the writing processes called Writers in short, can access the shared data file for possible updating i.e., insert a new record, delete a record, modify existing record, append a record to the end of the file etc..
- The necessary synchronization can be achieved by employing mutex semaphore in the reader proem code and write (mutual exclusion semaphore for writers).

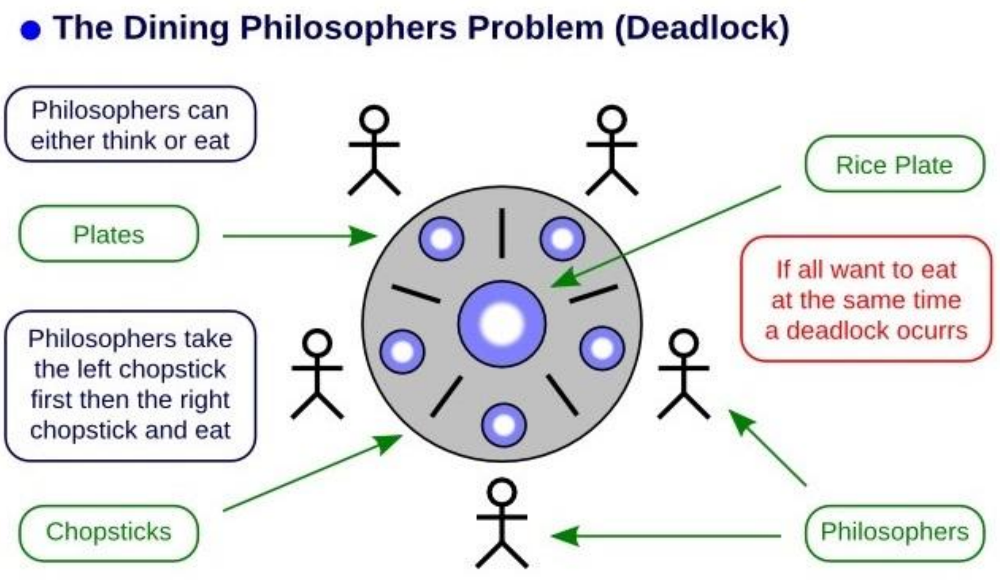
For instance

- **First Readers - Writers Problem**:- (Preference will be for Readers) Here, no reader process is made to wait has not obtained any prior permission to access shared file. The reader can proceed to execute its Read operations.

- **Second Readers-Writers Problem :** (Preference will be for writers) Here, no new reader process is allowed to access the shared file, once the writer is waiting and ready to access the shared data file for writing.

5) Explain the Dining - Philosophers Problem: [2014, 2015]

Ans:- The Dining - Philosophers problem highlights an example of large class concurrency - control problem where in, it represents need to allocate several resources among number of concurrent processors.
- The Dining - Philosopher problem suggests that at least 5 philosophers can sit across a common dining table, where in circular dining table is surrounded by 5 chairs. Each of the chairs can be occupied by individual philosophers.


The Dining Philosophers Problem (Deadlock)
Philosophers can either think or eat
Plates
Philosophers take the left chopstick first then the right chopstick and eat
Chopsticks
Rice Plate
If all want to eat at the same time a deadlock ocurrs
Philosophers

- Now, the problem starts as when philosophers became hungry and would like to eat rice!
- When philosophers are busy in thinking about ethics, research etc. they never interact with each other. But they would attempt at least to pick-up two chopsticks or spoons that are adjacent to their plates.
- In other words, synchronization problem exist and intern it may lead to deadlock &Starvation.
- To conclude, Dining - Philosopher problem could be solved by employing semaphores called ‘chopsticks’. This solution provides the necessary synchronization.

6) Explain Monitors. [2011, 2012]

Ans: - It is another fundamental high-level synchronization construct. Using monitors the synchronization mechanisms can be implemented for sharing abstract data types (variables).

<u>General Syntax of Monitor</u>

```
Monitor _name
// shared variables declaration
Procedure p.1 ( )
Procedure p2 ( )
}
Procedure 'jr. )
}
Initialization code
}
}
```

- Body of the monitor construct consist of shared variable declarations and a set of one or more user-defined functions or procedures that represent various operations to be performed on these abstract data types (i.e., on shared variables).
- In other words, the monitor is characterized by a set of user-defined or programmer defined operations that are represented via functions.

7) What is Deadlock? Explain the necessary for its occurrence. [2015, 2016]

Ans: - Deadlock is a situation in which two or more processors are waiting on same resources that are held by each other, this stage is called Deadlock.
- Deadlock situation can occur in a community of co-operating processes or among competing processes that need exclusive access to one or more resources of a computer system.
- Deadlock may be the side effects of synchronization techniques.
- In essence the need for inter-process communication among several co-operating processes gives rise to the need for synchronization.
- As we know that, a deadlock situation presents a problem or scenario where in two or more processes (P1, P2, P3, etc.,) get in to a hanging state (hold or locked or blocked state) such that each process is holding a resource that its adjacent or neigh burring process is requesting the same.

<u>Necessary conditions to Occurs Dead-Lock</u>

a) Mutual exclusion: - This states that, there is a proper understanding between two processors.
- At least one resource is held in a non-sharable mode that is only one process at a time can use the resource. If another process requests that resource, the requesting process must be delayed until the resource has been released.

b) Hold & wait:- One processor holding one resources. Another processor must wait.
- There must exist a process that is holding at least one resource and is waiting to acquire additional resources that are currently being held by other processes.

c) No preemption:- It doesn't go back for scheduling, it will wait until gets complete**.**
- Resouces cannot be preempted; that is, a resource can only be released voluntarily by the process holding it, after the process has completed its task.

d) Circular wait:- Two or Multiple processor waiting for subsequent process, this situation is called as 'Circular Wait'.
- There must exist a set {p0, p1,.....pn} of waiting processes such that p0 is waiting for a resource which is held by p1, p1 is waiting for a resource which is held by p2,..., pn-1 is waiting for a resource which is held by pn and pn is waiting for a resource which is held by p0.

Deadlock Detection

- The deadlock avoidance approach avoid the 'Unsafe States' although system might recover from them.

- When **Deadlock Detection** and Recovery techniques are employed, the system does now attempt to prevent deadlocks from Occurring. Rather, it allows deadlocks to occur, tries to detect them.
- Therefore, the detection and recovery strategy must provide for the system the following two algorithms.
- An algorithm to monitor the state of the system that verify and confirm whether the deadlock has occurred [Detection].
- An algorithm to recover from Deadlock [Recovery].

- Deadlock Detection Algorithm for Single Instance of each resource type.

This algorithm makes use of variant of resource - allocation graph known as a wait -for graph. - The deadlocked state occurs in the system if and only if the wait for graph contains a cycle.

- Deadlock Detection algorithm for multiple instances of each resource type.

- This algorithm makes use of time-varying data structures that are similar to those involved in banker's algorithm.

-The algorithm checks for every possible sequence of allocations for processes that are not completed.

- The algorithm is matrix based (the current allocation matrix C and request matrix R) depends upon comparing these vectors with available resource vector A.

-One alternative approach is to invoke detection algorithm every time when a resource request is to be processed. This method is expensive interns of considerable overhead on CPU time.

-An alternative strategy is to check for deadlock detection at every k (say once in 60) minutes or perhaps only when the CPU utilization drops down below some throughput (say below 50 percent).

<u>Recovery from Deadlock</u>

- Once the deadlock detection algorithm determines that a deadlock has been detected in a system, the system will have to be recovered from deadlock.
- For which the crude approach is to abort or kill one or more processes to break the deadlock or circular wait.
- In a second approach - one or more processes will have to be preempted, thus releasing their resources that would unblock the other deadlocked processes.

-We shall discuss and elaborate on the following 3 recovery approaches:

1) Recovery through killing processes [Process Termination].
2) Recovery through Preemption [Resource Preemption].
3) Recovery through Checkpoint / Roll back mechanism [Check pointing].

1) Recovery through Killing Processes [Process Termination]:- A simple and straightforward approach is to abort or kill all processes by rebooting the machine.

To summarized, the following methods used for recovery:-
→ Abort or Kill all deadlocked processes.
→ Abort or Kill one process.

2) Recovery through Preemption [Resource Preemption]:- Under some circumstances its possible to pull back an allocated resource from its currently executing process.
- But the ability to take away from the allocated resource forcibly well before the termination of currently holding process.

- Deadlock recovery with Resource preemption would call for 3 issues
a) Selecting a victim b) Roll back c) Starvation

Deadlock Presentation:
a) Elimination of Mutual Exclusion
b) Elimination Hold and wait
c) Elimination of No-preemption
d) Elimination circular wait.

8) Explain Banker's Algorithm. [2011, 2012, 2014]
Ans:- The banker's algorithm is the best known of the Deadlock avoidance strategies. This algorithm is first introduced by Mjkstra in 1965.
- The classic model of a state used in the deadlock avoidance strategy comes from Dijkstra's analogy of resource allocation (1968).
- It's a scheduling algorithm used for resource-allocation and deadlock avoidance. Banker's algorithm is applicable to resource allocation systems with multiple instances of each resource type.
- The simple assumption is such that bank must satisfy all of its customers by leading money to each and every customer on the basis of a line of credit.
- We know that - a bank can have a designated amount of cash at any time subject to its transaction load at that particular town or city.
- A line of credit is an agreement on maximum claim on resources by the processes. In banking environment a line of credit is an agreement by the bank.

Chapter – 4
Memory Management

1) What is Memory Management?

Ans:- Main Memory is a repository to hold the run-time program instruction and data operands to be operated upon. The function of Memory Management is to keep the track of parts of main memory that are in use, allocated memory blocks or segments.

2) Explain Page Replacement Algorithms:

Ans: - As we know that Demand Paging offers extremely large virtual memory to programmers without constraining the actual size of physical main memory
- Page replacement policy or algorithm has to determine as to which page from physical main memory must be removed (or swapped-out to hard disk).
- Therefore, page replacement is fundamental to Demand Paging. In other-words, successful implementation of Demand Paging scheme would call for now, to develop the **following algorithms.**

I. Frame-Allocation Algorithm (Fetch: and Placement Policy)
2. Page-Replacement Algorithms (The Replacement policy)

- With multiprogramming, the multiple processes must be allocated main memory simultaneously.
- The Frame Allocation has to decide how many frames to allocate at a time to each process participating in execution.
- There is good number of page-replacement Algorithms Sometimes, it quite easy to discuss and/or narrate the best possible page replacement algorithm.

- The various page replacement algorithms listed below:

a) First-in First-out: FIFO page replacement algorithm.

- FIFO page replacement algorithm replaces the page that has been resident in main memory for the longer time.
- Foe in which the O.S maintains a separate queue of pages that are all resident in physical main memory frames. The oldest page is at the head of queue (first-in).
- FIFO algorithm is simple to implement but it functional behavior cannot be incorporated or applicable to most of the programs. It is independent of locality.
- So FIFO kind of page replacement will increase page-fault and slow-down process execution.

b) Optimal page replacement algorithm.

- Optimal Page replacement algorithms suggest that - On a page fault, replace the page which will not be used or referenced.
- That is, those pages getting frequent references need not be replaced. Somehow, one should knock about a page reference which will not be referenced say until 10,100, or perhaps 1000 instructions later.
- Therefore, an optimal page must be chosen to be swapped-out from main memory.

Limitations of this Algorithm
- This algorithm is not generally realizable. Since, when a page fault occurs, the operating system has no way of knowing, in advance,
- Therefore, it is difficult to implement the Optimal Page Replacement Algorithm.

Benefits of this Algorithm
- This algorithm has lowest page-fault rate of all algorithms.
- This algorithm can be used for simulating and evaluating other new algorithm for their comparative studies.

c) Least recently used: LRU page replacement algorithm.
d) Second Chance page replacement algorithm.
e) Least recently used: LFU page replacement algorithm.
f) Most Frequently used: MFU page replacement algorithm.

3) Explain Paging (Non-Contiguous Allocation). [2014, 2015]

Ans:- Paging techniques in different forms is commonly used in many operating systems.
- The technique permits the logical address space of a process to be non-contiguous.
- According, the logical memory is broken down into fixed-size partitions called pages.
- Similarly, the physical memory that accommodates these pages is also broken into blocks of same size called frames or physical blocks (memory blocks).

- **The Basic Method:- Paging Hardware:-**The following figure illustrates the hardware support for paging memory management scheme.
- With this approach, the addresses generated by CPU contain two components.
1) Page number 'p' (in which desired instruction exists)
2) Offset ‘d’ in that page 'p' [the exact location (distant) of the desired instruction that page only].
- this hardware to perform the mapping from user (logical) address space to physical memory) address space.

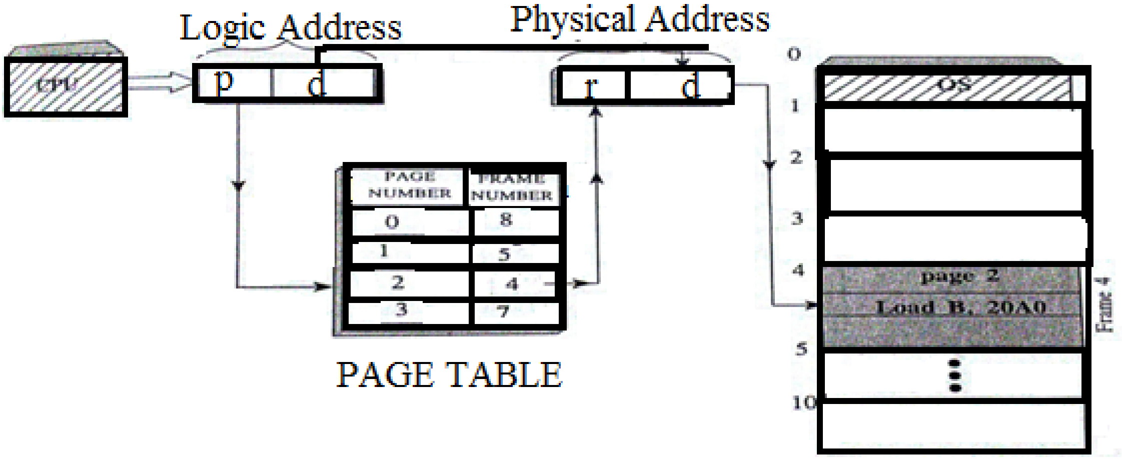


Figure :- Paging Hardware

- These registers may be either special hardware registers or some reserved area in main memory.
- Note that the page tables are much smaller in size than the given address space.
- Accordingly, the page table consists of the base address (starting address) of each page (in physical main memory).

4) Explain Advantages and Disadvantages of Paged Allocation Scheme Advantages include.

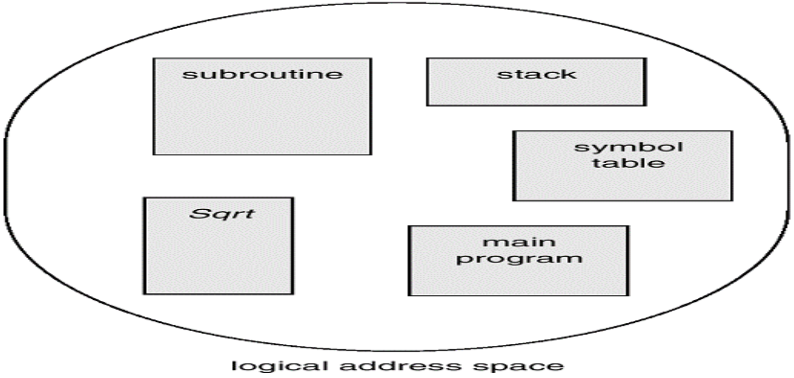
Ans:- 1) No External Fragmentation.
2) No Compaction or Relocation overhead
3) Support time sharing and virtual memory.
4) Job's partitions need not be contiguous in physical main memory.
5) Only locations in a given page must be continuous.
6) Several processes share the same physical pages.

Disadvantages:
1) Overhead associated with each address translation.
2) Extra memory or Registers required to maintain page Tables.
3) Internal Fragmentation and/or Page breakage.

5) Explain Segmentation with example. [2011, 2013, 2014, 2015]

Ans: - It is revealed from Paging Memory Management that user's view of memory is not same as the actual physical core memory.
- That is user's view is mapped onto physical main (core) memory.
- This kind of mapping accommodates the different views between logical memory and physical memory.
- But for practical reasons, the user's view (perceive) main memory as if it were a simple linear physical core memory comprising a group of variable-sized segments.
- In essence, a segment is a grouping of information that is treated as an entity that is, a segment may be one full program, a sub-program, a sub routine, a function, procedure or a program module including data structures such as tables, stacks, arrays.

- The following figures depict user’s view of a program (i.e the program' address space) as a collection of variable-sized segments:



The Basic Method: Segmentation Hardware

- Segmentation hardware implements the mapping of two-dimensional logical address into one-dimensional physical address in Main Memory.

- This mapping is implemented via a segment map or segment table.

- Since practical physical main memories are linear, it is quite necessary to use a segment map to translate the job's address space into linear memory locations.

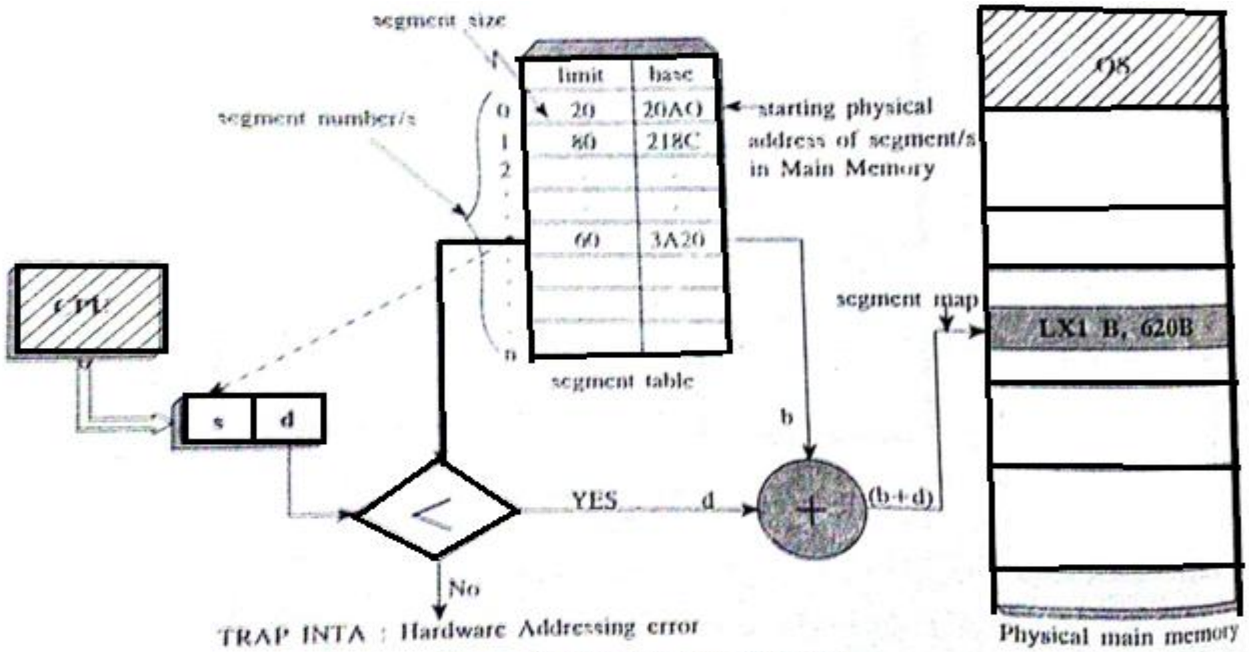


Figure ↑ Segmentation Hardware

5) Explain Optimal Page replacement algorithm.

Ans:- Optimal Page replacement algorithms suggest that - On a page fault, replace the page which will not be used or referenced.

- That is, those pages getting frequent references need not be replaced. Somehow, one should knock about a page reference which will not be referenced say until 10,100, or perhaps 1000 instructions later.

- Therefore, an optimal page must be chosen to be swapped-out from main memory.

Limitations of this Algorithm

- This algorithm is not generally realizable. Since, when a page fault occurs, the operating system has no way of knowing, in advance,

- Therefore, it is difficult to implement the Optimal Page Replacement Algorithm.

Benefits of this Algorithm

- This algorithm has lowest page-fault rate of all algorithms.

- This algorithm can be used for simulating and evaluating other new algorithm for their comparative studies.

6) Explain Swapping Process.

Ans: - 1) Swapping: - It permits interleaved execution of two or more processes in time-sharing multi-user systems. It allows main memory to be shared effectively among two or more active processes.

- The key observation about swapping technique is that if a process is not going to use the CPU for a relatively longer tune, then such process is swapped out (pulled out) to backing store.

- Swapping technique requires a secondary storage which is usually a hard disk. At any given instance of time when an executing process is blocked, this removal process is called

Swap-Out.

- After sometime, the swapped-out process has to be brought back to primary memory from secondary storage unit.
- This insertion process is termed as Swap-in.
- Swapping effect can result in noticeable increase in the response of time.

Advantages of swapping:

1) Swapping an increase the degree of multiprogramming, since, swapping can at least perform load-balancing on an oversubscribed time sharing.

Disadvantages of swapping:

1) Swapping a process, which is blocked for I/O, is a tedious job, since I/O operation may access user memory asynchronously for I/O buffers.

7) What is Virtual Memory?

Ans: - Virtual Memory is a predominant memory management strategy in modern operating systems that allows programs to run even if the program contents are partially loaded into main memory.

- The basic idea behind the ‘virtual memory’ management is that total size of the program including the data and stack requirement can exceed the amount of physical main memory.
- So complete program need not to be resident in primary memory during the execution.
- Virtual memory offers a mechanism to accommodate one or more processors into physical memory whose address space is quit larger than actual available memory.
- It extends the capabilities of the program that is only partially loaded into main memory.
- Virtual memory enables the programmers to write programs that would occupy extreme large virtual address space.
- With Virtual memory, end users will be given an impression that they have resident in main memory.

8) What is demand paging? [2014]

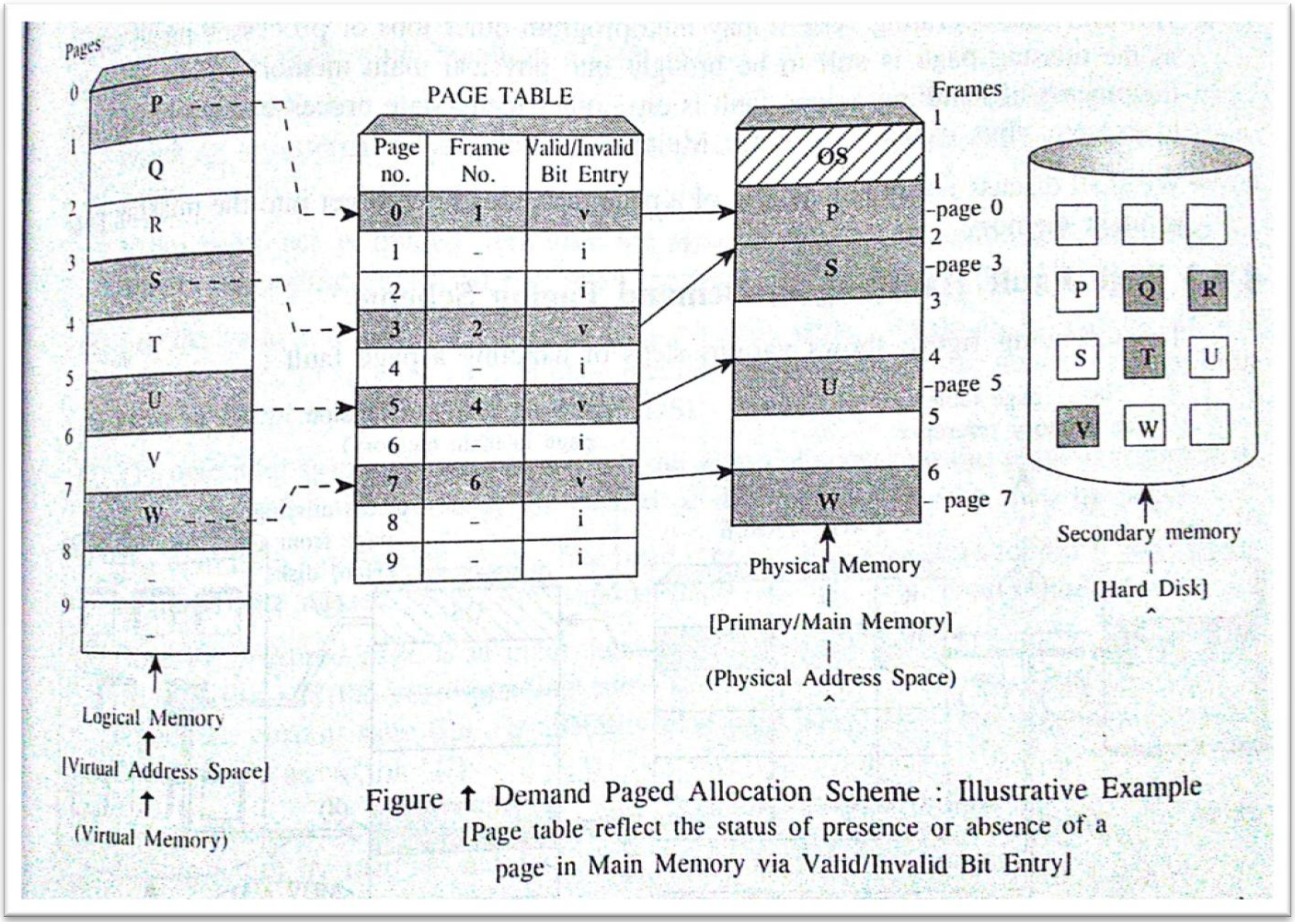
Ans:- Demand paging suggests- load pages into main memory only on demand and not all in advance. The idea is to keep only those parts or pages of the program in main memory that are currently in use and rest of the program on secondary memory.

- In many situations, program execution is not uniformly distributed over the entire address space. The execution may be localized /concentrated only in certain pages.

- Demand paging is similar that keep track status of pages as to which page must remain in physical main memory.

- Demand paging is similar to paged allocation scheme with swapping.

9) Explain paging concept with neat diagram. [2011]



PAGE TABLE

Page no.	Frame No.	Valid/Invalid Bit Entry
0	1	v
1	-	i
2	-	i
3	2	v
4	-	i
5	4	v
6	-	i
7	6	v
8	-	i
9	-	i

Figure ↑ Demand Paged Allocation Scheme : Illustrative Example
[Page table reflect the status of presence or absence of a page in Main Memory via Valid/Invalid Bit Entry]

Two types:

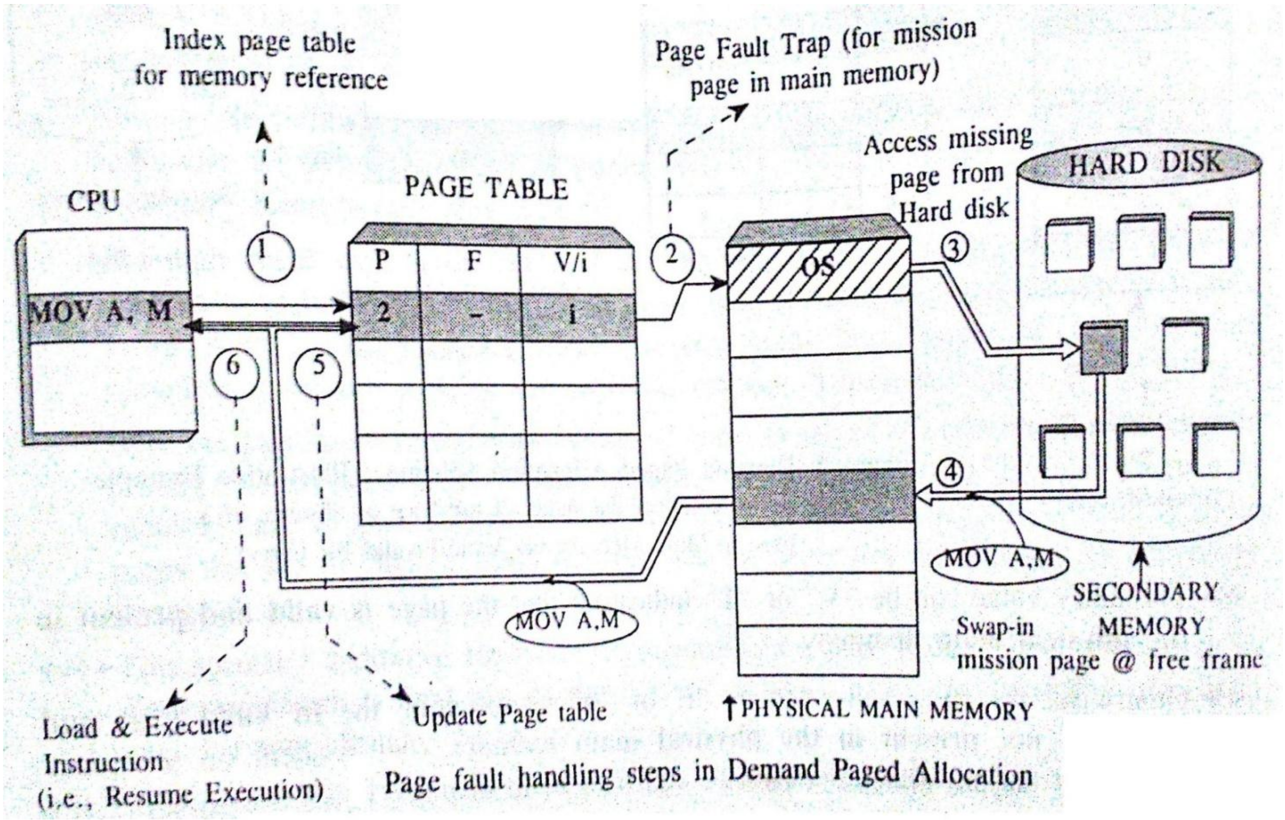
1) Hardware support:- In demand paging memory allocation systems-only part of the demand handling is performed by the Hardware.

- Majority of the work will be carried out by O.S with the help of paging algorithms.

- The high speed disk space reserved for the purpose of swapping demand pages is reffered to as **Backing store** or **swap space** & the secondary disk storage is termed **Swap Device.**

2) Demand pages allocation: Here virtual memory is larger than physical main memory. a program or job occupies 8 pages in logical physical out of which 4 pages are accommodate in physical main memory. On demand however, the locking store-hard disk contains all 8 pages of job's virtual address space as contiguous disk blocks.
- Here the page number is used as an index into 'Page table'. The page table provides mapping of each logical page into physically allocated frame or memory block (page frame) in physical main memory.

9) Explain Page Fault handling in demand paging scheme. [2012]



Page fault handling steps in Demand Paged Allocation

Ans:- We shall list out now the procedural steps being followed by an operating system while handling a page fault:-

a) Determine whether the memory reference to this page is valid or invalid cross checking an internal table associated with process control block (PCB).
- If this reference is Invalid, terminate the process. Otherwise, index the page table for the corresponding page frame in main memory,

b) For the valid reference suppose indexing the page table returns an invalid bit "0" or a "i" for missing page in main memory, then the hardware generates page fault trap interrupt to the operating system (OS).

c) The operating system (OS) now performs the action of accessing the secondary memory - hard disk and acquires (read) the needed or demanded page.

d) The operating system (OS) finds a free page frame in physical main memory and swaps-in (loads) the missed page from backing store into one such free frame. - To conclude, this is how the **Demand Paging** scheme is able to execute a process although some portions (pages) of it are not currently resident in Main Memory.

10) What Pure Demand Paging? Explain Advantages and Disadvantages.

Ans:- Demand Paging suggested that swap-in an additional page into main memory as needed, following the page fault trap to an OS. That is, to start with, at least one page of each and every process used to be in physical main memory.

- But Pure Demand Paging suggests there is no need to bring (swap-in) a page into physical main memory until it is required. That is, CPU can proceed to execute a process without any of its page in main memory.

- Pure demand page need not bring even a single page of a process physical main memory until the same is demanded to provide.

Advantages of Demand Paging:

a) Provides the programming advantages of an apparently very large memory.

b) Contributes to increase system throughput in several way.

c) Allows system to multiprogramming at the rate 25 % more users.

Disadvantages of Demand Paging:

a) Paging overhead

b) Page replacement is expensive.

c) Takes longer time to execute the process

11) Explain Internal and External fragmentation. [2011]

Ans: - Fragmentation: - It is development of unusable “holes” or “fragments”.

Internal Fragmentation:

- The problem of external fragmentation may be solved using compaction technique and paging technique.

- In technique, the job’s address space is sub-divided into equal sized pages (1000bytes). So memory allocated to a process may be little larger than requested memory size.

- If the uniform page size of 1000 bytes each say… Then individual pages are left blank with differences of 10 bytes, 15 bytes, bytes, 110 bytes etc., respectively.

- This differential (free) left - out memory holes its respective pages is termed **internal Fragments** and said to cause **Internal Fragmentation.**

- This kind of wastage in the last page is also referred to as Page breakage.

- To conclude, internal fragmentation occurs whenever page size does not match the process's requested memory occupation size.

External Fragmentation: - The major problem with multiple-partition allocation algorithms is the development of external fragments. (the phenomenon is referred to as external fragmentation).

This can be seen in the following figure:

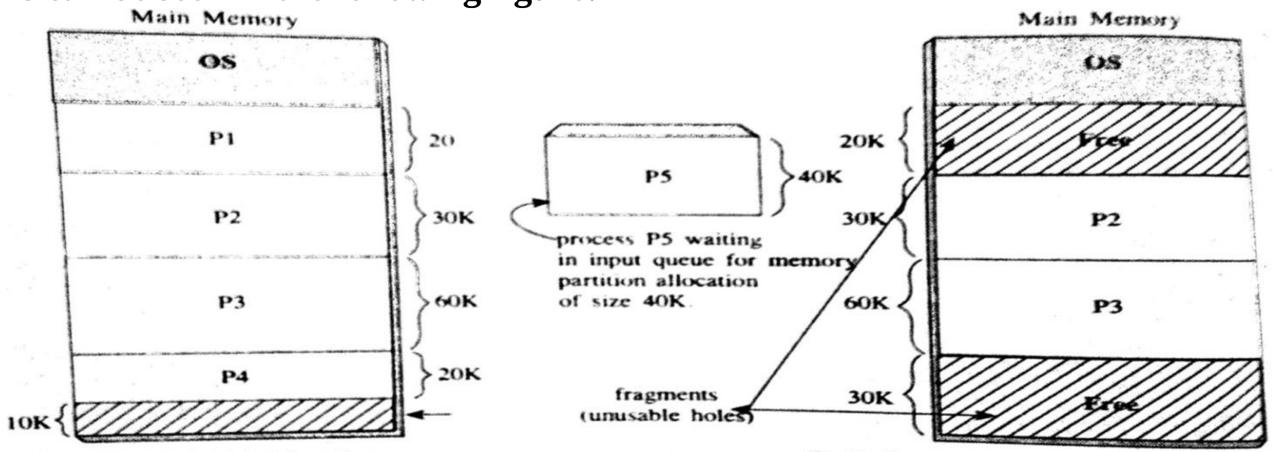

Main Memory
OS
P1
20
P2
30K
P3
60K
P4
20K
10K
P5
40K
process P5 waiting in input queue for memory partition allocation of size 40K.
fragments (unusable holes)
Main Memory
OS
20K
Free
30K
P2
60K
P3
30K
Free

Fig.(1) Initial Allocation

Fig(2) Development of external fragments after P1 and P4 are terminated

As processes (say p1,p2, p3,p4,p5….) enters & exit main from memory. The single large contiguous memory space called Hole is broken down and scattered into number of uneven sized holes.

- As the time passes i.e., more and more process are loaded into and released from main memory, the external fragmentation develops.
- Enough memory space available but scattered here and there as unusable memory chunks or memory fragments of exact sizes.

12) Explain Thrashing. [2013]

Ans:- The phenomenon of thrashing occurs whenever is a mismatch between the processes locality of references and size of the actually required physical space in main memory.

- Thrashing increases the number of page faults causing slow down of process's execution.
- Thrashing results into very low throughput of the system and sometime it may drop down to less than 1% also.
- A process is said to suffer from Thrashing if it spends more time in paging rather thean useful execution of page counts.

Strategies of Prevent Thrashing:

1) By decreasing the degree of multiprogramming, Thrashing can be controlled and CPU utilization can be maximized.

2) To conclude, Thrashing has been high page-fault rate.

3) Once the Thrashing sets-in, the CPU utilization drops down dramatically causing low system throughput.

Chapter – 5
File Management

1) What is file? [2011, 2012, 2014, 2015]
Ans:- A file is collection of related information units called Records.
- It is the way to organize and store the information logically in a group (or collection) is called File.
- In other-words, file provides a convenient way to store the information on the secondary storage disk structure and read it back in the same way.

2) Explain file attributes. [2012, 2014, 2015, 2016]
<u>File Attributes</u>:
i) Name:- The alphanumeric file name string with or without file extension and appears on the screen when listed out with **dir or Is** command of MS-DOS OS or UNIX OS respectively.

ii) Size: - The current size of file tells how big the file is at present. That is data occupation or file contents in that file is indicated in terms of bytes.

iii) Date, time and user-identification:- These attributes keep the track as to when the file was created. Its last modification or usage most recently accessed and / or modified for its end users. Such information is required to monitor the protection, security. Etc..

iv) Identifier: It is distinguishing number that uniquely the file within the file system. This identifier is not accessible or not visible to end –users.

v) Type: It suggest different file types (.obj, .bak, .bat .com).

vi) Location: It is pointer to the storage device & to the location of the file on that device.

vii) Protection: It suggests control right's such as who is allowed to perform reading a file. Writing a file etc…

3) Explain File Operations. [2011, 2012, 2014, 2015, 2016]
Ans: **File Operations are:**
i) <u>Create a file</u>:- Initially file must be created without any data in it. The create system calls declares that new file, afresh is going to exist with some attributes settings.

- File creation operation calls for:
  → Allocating free space on the disk file system.
  → Making directory entry for new file.

ii) Open a file: Most systems enforces programmer to open a file explicitly with an open system call well before using that file.
- It enables system to fetch necessary attributes of that file.

iii) Read a file: The read system call reads data from the given open file at current -file_ position pointer.

- The caller must specify the file pointer or the name of the file (location) and how much data (bytes, lines) to be read from that file and where exactly the hence read data must be copied (the buffer location).

ii) Write a file: The write system call writes data to the given file which is opened in write mode (or append mode). The writing takes place again at current write pointer. Usually, the system maintains a write pointer to the location in the file as to where exactly the next write to take place

iii) Append a file: The append system call can only add the supplementary data to the end of the file. For which caller must open a file in append mode. File appending operation is restricted form of write operation.

iv) Delete a file:- The delete system call erases the specified file and releases the file space to the disk: Whenever the file is no longer needed, file deletion operation can be carried out and the deleted file name will disappear from the directory entry.

v) Seek a file (position or Reposition within a file): The seek system call perform file seek operation and there by repositions the current file pointer to a specific value (place) in the file.
After seek operation data can be read from or written to the file from this new position. Seek operation is an useful choice for updating random access files.

vi) Truncate (contents of) a file: Whenever the structure of the file (its attributes both in number and field size) is to be preserved and only its contents are to be erased then truncate system call will do the same. So that file size is truncated to zero and file space is returned to disk.

vii) Rename a file: The rename system call allow users to change the name of an existing file. It's not file copying operation.

viii) Close a file: The close system call closes the file and delinks it from main memory and disk addresses and internal descriptors / tables.

4) Explain File Types. [2015]

Ans: File Types All major operating systems support different types of files. For instance MS-DOS and WINDOWS operating systems support Regular file types and Directories.

The Different File Types are:

i) Directories:- This is the system files that maintain the structure of the file system. Accordingly, a directory provides a mechanism for the logical association of groups of files and other nested sub-directories.

ii) Regular Files:- This is either ASCII files or binary files in general. ASCII files contain lines of text data (strings of characters).
Binary files exist in two forms i.e., the files with .com and .exe extensions.
Whereas ASCII file contents can be displayed and printed too... Further such file contents can be edited and interpreted up to the end user convenience.

iii) Character Special Files: emphasize serial input and output and are used to model serial I/O devices such as video terminals, printers and network I/O. Serial I/O suggest character-by -character input / output.

iv) Block Special Files: This are used to model disks. Since a disk is written in blocks (physical records). i.e., block-by-block input/output.

5) Explain File Structures.

Ans: i) Byte Streams kind of File Structure: Here file is an unstructured collection of bytes. O.S system does not mean or interest these 8-bit byte sequences.

ii) Record Streams kind of File Structure: A High level file system provides record-streams translation.

iii) Tree structured kind of File Structure: Here file is organized into tree of records. Records can be of both varying lengths and same lengths.

6) Explain File Access Methods in brief. [2011, 2012, 2014]

Ans: Major File access methods are:

i) Sequential Access ii) Direct Access iii) Indexed sequential Access.

i) Sequential Access: Sequential access to the file records is defend by current file position & the current position pointer index records sequence by successive addition to locate the next record in sequence.
- Sequential access suggest the processing to take place by accessing the records as they appear in sequential i.e. retrieve one record after another that are all adjacent to each other.

ii) Direct Access: In direct access, the file is organized as a numbered sequence of fixed length logical records. Such files allow arbitrary blocks to be read or write.
- The silent feature of direct access method is to provide immediate **Random access to any information.**
- Here file must contain new parameter called Block number.

iii) Indexed sequential Access : Combining Sequential access and direct access using table of indexes to form which is known as the ‘Indexed sequential Access’ and indexed sequential files.
- When a index is created, an index record for each block in the ordered file contains search key field and pointer to one or more records of individual blocks

3070 – E21 – VS BCA (R) – N – 11

FIFTH SEMESTER B.C.A DEGREE EXAMINATION, NOVEMBER 2011
(REVISED)

OPERATING SYSTEM

Time : 3 Hours Max. Marks : 80

*Answer any **five full** questions.*
Draw diagrams wherever necessary.

1. a) What is an operating system? Explain real time and distributed systems.
   b) Explain operating system calls and services. 8+8=16

2. a) Define a process. Explain the process states and operations of processes.
   b) What are co-operating processes? Explain inter-process communication. 8+8=16

3. a) Consider the following set of processes with the length of CPU burst time given in Milliseconds.

Process	Burst time	Arrival time
P_1	10	0
P_2	1	1
P_3	2	2
P_4	1	3
P_5	5	4

i. Draw Gantt chart illustrating the execution of these processes using pre-emptive SJF, FCFS and routing robin (quantum =4) scheduling algorithms.

ii. Calculate average waiting time foe each of the scheduling algorithm.

iii. What is the turn around time for process P_1 in each of the scheduling algorithm?

b) What is CPU scheduling? Explain the criteria for comparing CPU scheduling algorithm.
8+8=16

4. a) What is critical region? Explain readers and writers problem.
   b) Explain the strategies for dealing with deadlocks.
   c) Explain banker's algorithm. 8+4+4=16

5. a) Consider the following snaphot of a system with 5 processes P_1 to P_5 and 4 resource types R_1 to R_4 have instances $\langle R_1, R_2, R_3, R_4 \rangle = \langle 8, 5, 9, 7 \rangle$.

	Allocation					Max			
	R_1	R_2	R_3	R_4		R_1	R_2	R_3	R_4
P_1	2	0	1	1	P_1	3	2	2	4
P_2	0	1	2	1	P_2	1	2	5	2
P_3	4	0	0	3	P_3	5	1	1	5
P_4	0	2	1	0	P_4	1	5	3	0
P_5	1	0	3	0	P_5	3	0	3	3

i. What is the content of ‘available’ matrix?
ii. What is the content of ‘need’ matrix?
iii. Find the safe sequence, if the system is in safe state.

b) What is segmentation? Explain internal and external fragmentation. 8+8=16

6. a) What is demand paging? Explain paging concept with a neat diagram.
b) Consider the following page reference string : 1,2,3,4,5,3,4,1,6,7,8,9,7,8,9,5,4,5,4,2.
How many page faults would occur for the optimal, FIFO and LUR replacement algorithms assuming 4 frames (initially all are free). 8+8=16

7. a) What is a file? Explain the different methods to access a file.
b) Explain the file attributes and operations performed on a file. 8+8=16

8. Write short notes on any **four**:
a) Single-level v/s Two-level directory
b) Monitors
c) Process states
d) Parallel systems
e) Scheduling algorithms
f) Operating system components. 4 x 4=16

3070 – E21 – VS BCA (R) – N – 12

FIFTH SEMESTER B.C.A DEGREE EXAMINATION, NOVEMBER 2012
(REVISED)

OPERATING SYSTEM

Time : 3 Hours Max. Marks : 80

*Answer any **five full** questions.*
Draw diagrams wherever necessary.

1. a) What is an operating system? Explain Distributed and real time systems. 8+8=16
   b) Explain various operating system services and system calls.

2. a) What are co-operating processes? Explain inter-process communication. 8+8=16
   b) Define a process. Explain the different states that a process taken while execution. Explain in detail process control block.

3. a) What is critical region? Explain readers and writers problem. 6+6+4=16
   b) What is critical section problem? Explain the requirements to solve the critical section problem.
   c) Explain the Banker's algorithm (multiple process solution).

4. a) What is Scheduling? What are the criteria for CPU scheduling algorithm? 8+8=16
   b) Consider the following set of processes with the length of CPU burst time and arrival time given in milli seconds.

Process	Priority	Burst time
P_1	3	3
P_2	1	1
P_3	2	3
P_4	4	4
P_5	5	2

The processes are assumed to arrived in the order P_1, P_2, P_3, P_4 & P_5.

i. Draw Gantt chart illustrating the execution of these processes using Fc Fs, SJF a non preemptive Priority (a smaller priority number implies a high priority) and routing robin (RR) (quantum =1) scheduling algorithms.

ii. What is the turn around time of each process for each of the scheduling algorithm?

iii. What is the waiting time of each process for each of the scheduling algorithm?

5. a) Explain Resource – Allocation graph. 8+8=16
   b) Consider a system 5 processes p_0 through p_4 and 3 resource type A has 10 instances B has 5 instances and C had 9 instances.

	Allocation			Max			Available		
	A	B	C	A	B	C	A	B	C
P_0	0	1	0	7	5	3	3	3	2
P_1	2	0	0	3	2	2			
P_2	3	0	2	9	0	2			
P_3	2	1	1	2	2	2			
P_4	0	0	2	4	3	3			

i. What is the content of ‘need’ matrix?
ii. Is the system safe? Given the safe sequence if it is safe state.
iii. If a request from process P_1 arrives for (1, 0, and 2) can the request be granted immediate.

6. a) What is page fault? What are the steps following in handling a page fault? Explain with neat diagram. 8+8=16
   b) Consider the following page reference string : 7,0,1,2,0, 3,4,2,3,0,3,2,1,2,0,1,7,0,1.
   Find the number of page faults in LRU, FIFO and optimal replacement algorithm (Assumption: 3 frames).

7. a) What is a file? Explain the different methods to access a file. 8+8=16
   b) Explain the attributes and file operations.

8. Write short notes on any **four**: 4 x 4=16
   a) Context switch
   b) Operating system components.
   c) System model
   d) Inverted page table
   e) Parallel processing
   f) Monitors

8079 – E21 – VS BCA (R) – N – 13

FIFTH SEMESTER B.C.A DEGREE EXAMINATION, NOVEMBER 2013
(REVISED)

OPERATING SYSTEM

Time : 3 Hours Max. Marks : 80

Answer any ***five full*** *questions.*
Draw diagrams wherever necessary.

1. a) Define Operating System ? Explain various Operating System Services.
   b) What are loosely coupled system ?
   c) What are virtual Machines? Give their benefits.

(6+4+6=16 marks)

2. a) What is Inter-process communication ? How is IPC provided in an OS?
   b) What is a dispatcher? What is its function?
   c) Explain the term context switch.4

(10+ 3+ 3=16 marks)

3. a) Consider the following set of processes with the length of CPU burst time and arrival time given in milli seconds.

Process		Burst time	Priority
P_1	....	24	2
P_2	....	3	0
P_3	....	3	1

The processes are assumed to arrived in the order P_1, P_2, P_3

i. Draw Gantt chart illustrating the execution of these processes using FCFS, SJF, non preemptive Priority , RR (quantum =5 milliseconds).

ii. What is the waiting time and turn around time for each process of each of the scheduling algorithm

b) Explain different types of schedulers.

(10+6=16 marks)

4. a) Define Race condition? Explain two process solution.
   b) Explain Reader and Writers problem.

(8+8=16 marks)

5. a) Consider a system 5 processes p_0 through p_4 and 3 resource type A, B, C. resource type A has 10 instances, B has 5 instances and C had 7 instances.

		Allocation					Max		
		A	B	C			A	B	C
P_0	…….	0	1	0	P_0	……	7	5	3
P_1	……	2	0	0	P_1	……….	3	2	2
P_2	…......	3	0	2	P_2	…….	9	0	2
P_3	………	2	1	1	P_3	……….	2	2	2
P_4	………	0	0	2	P_4	……….	4	3	3

Answer the following questions:

i. What are the available Resources?
ii. What is the contact of matrix Need?
iii. Is the system safe? Give the safe sequence if it is in safe state.

b) Explain deadlock recovery algorithms.

(10+ 6=16 marks)

6. a) Explain Segmentation with an example.
b) Consider the following page reference string : 1,2,3,4,1,2,5,1,2,3,4,5. How many page faults would occur for the optimal, FIFO and LUR replacement algorithms, assuming three frames.
c) What is thrashing? Explain working-set model.

(6+6+4=16 marks)

7. a) Explain the following threats : Worm and Virus.
b) Explain Indexed allocation of disk space.
c) What is meant by Boot block?

(6+8+2=16 marks)

8. Write short notes on any **four**:

a) Multilevel queue scheduling.
b) Access matrix.
c) Acylic graph directory.
d) Semaphores.
e) Parallel systems

(4 +4+4+4=16 marks)

7084 – E21 – VS BCA (R) – N – 14

FIFTH SEMESTER B.C.A DEGREE EXAMINATION, NOVEMBER 2014
(REVISED)

OPERATING SYSTEM

Time : 3 Hours Max. Marks : 80

*Answer any **five full** questions.*
Draw diagrams wherever necessary.

I. a) Define Operating System ? Explain the different Operating System Services.
b) Explain operating system calls and system program.
(8+8=16)

II. a) Define a process. Explain the states transition diagram for process.
b) Explain multiprocessor system.
c) Explain in detail about PCB.
(6+4+6=16)

III. a) Differentiate long, short and medium schedulers.
b) What are co-operating Processes and IPC?
c) Consider the following set of processes with the length of CPU burst time and arrival time given in milliseconds.

Process	Burst time
P_1	24
P_2	3
P_3	3

The processes are assumed to have arrived in the order P_1, P_2, P_3 all at time

i) Draw Gantt chart illustrating the execution of these processes using FCFS, SJF, RR (Quantum =4) Scheduling.
(4+4+8=16)

IV. a) What is critical section problem? Explain the requirements to solve the critical section problem.
b) Explain the Banker's algorithm (multiple process solution).
c) What is Semaphore? Explain dinning philosophers problem.
(4+8+4+16)

V. a) State and explain the condition that must hold simultaneously for a system to be in deadlock state.
b) Consider a system 5 processes p_0 through P_4 and 3 resource type A, B,C. Resources type A has 10 instances B has 5 instances and C had 7 instances.

	Allocation						Max	Available	
	A	B	C	A	B	C	A	B	C
P_0	0	1	0	7	5	3	3	3	2
P_1	2	0	0	3	2	2			
P_2	3	0	2	9	0	2			
P_3	2	1	1	2	2	2			
P_4	0	0	2	4	3	3			

Answer the following questions using Banker's algorithm.

i. What is the content of 'Need' matrix?

ii. Is the system safe? Given the safe sequence if it is safe state.

iii. If a request from process P_1 arrives for (1, 0, and 2) can the request be granted immediately.

(8+8=16)

VI. a) Explain paging.

b) Define segmentation concept.

c) Consider the following page reference string :7,0,1,2,0,3,4,2,3,0,3,2,1,2,0,1,7,0,1. Find the number of page faults in LUR, FIFO & optimal replacement algorithms (Assuming : 3 frames)

(4+4+8=16)

VII. a) What is a file? Explain the different methods to access a file.

b) Explain the attributes and file operations.

(8+8=16)

VIII. Write short notes on any **four**:

a) Distributed system

b) Parallel processing

c) RR Scheduling

d) Deadlock avoidance

e) Demand paging

f) Directory structure.

(4 x4=16)

5074 – E21 – VS BCA (R) – N – 15

FIFTH SEMESTER B.C.A DEGREE EXAMINATION, NOVEMBER 2015
(REVISED)

OPERATING SYSTEM

Time : 3 Hours Max. Marks : 80

*Answer any **five full** questions.*
Draw diagrams wherever necessary.

I. a) Define Operating System ? Explain Operating System Services in detail.
b) Explain operating system components in detail.
(8+8=16)

II. a) Define a process. Explain various in process control block (PCB) in detail.
b) What is thread? Explain benefits and multithreading models.
(8+8=16)

III. a) Explain scheduling the criteria used for CPU scheduling.
b) Consider the following set of processes with the length of CPU burst time given in Milliseconds.

Process	Burst time	Priority
P_0	24	2
P_1	3	0
P_2	3	1

The processes are assumed to have arrived in the order P_0, P_1, P_2

i. Draw Gantt chart illustrating the execution of these processes using FCFS, SJF, Priority and RR (quantum =5 milliseconds).

ii. Calculate average waiting time and turn around time. (6+10=16)

IV. a) Define race condition. Explain two process solution.
b) Explain Dining-philosophers problems. (8+8=16)

V. a) Explain paging.
b) Explain segmentation with example.
c) Consider the following page reference string :
1,2,3,4,1,2,5,1,2,3,4,5.

How many page faults would occur for the optimal, FIFO and LUR replacement algorithms, assuming three frames. (4+6+6=16)

VI. a) What is a file? Explain the following in detail:
i. File attributes
ii. File operations
iii. File types

b) What is deadlock? Explain the necessary conditions for its occurrence. (10+6=16)

VII. a) Consider a system 5 processes p_0 through P_4 and 3 resource type A, B,C. Resources type A has 10 instances B has 5 instances and C had 7 instances.

Process	Allocation	Maximum	Available
	A B C	A B C	A B C
P_0	0 1 0	7 5 3	3 3 2
P_1	2 0 0	3 2 2	
P_2	3 0 2	9 0 2	
P_3	2 1 1	2 2 2	
P_4	0 0 2	4 3 3	

Answer the following questions using Banker's algorithm.

i. What is the content of 'Need' matrix?

ii. Is the system safe? Given the safe sequence if it is safe state.

b) Explain with an example how resource allocation graph is used to describe deadlock. (10+6=16)

(4 X 4=16)

8. Write short notes on any **four**:

a) State transition diagram for process.

b) Multiprocessor system.

c) Multilevel queue scheduling.

d) Semaphores.

e) Schedulers.

6026 – E21 – VS BCA (R) – N – 16

FIFTH SEMESTER B.C.A DEGREE EXAMINATION, NOVEMBER 2016
(REVISED)

OPERATING SYSTEM

Time : 3 Hours Max. Marks: 80

*Answer any **five full** questions.*
Draw diagrams wherever necessary.

1. a) Explain abstract view of the components of a computer system.
b) Define Operating System? Explain the components of operating system in detail.
(6+10=16)

2. a) List and explain O.S services in detail.
b) Explain process states with neat diagram.
c) Explain Schedulers. (6+6+4=16)

3. a) Consider the following set of processes with the length of CPU burst time in milli seconds and priority.

Process	Burst time	Priority
P_1	6	1
P_2	8	3
P_3	7	0
P_4	3	2

The processes are assumed to arrived in the order P_1, P_2, P_3, $P_{4.}$

i. Draw Gantt chart illustrating the execution of these processes using FCFS, SJF, Priority and Round Robin (Time Quantum =4 milli seconds).
ii. Calculate average waiting time and average turn around time.
iii. Specify which algorithm results in minimal average waiting time.

b) List and explain Scheduling Criteria. (10+6=16)

4. a) Explain Interprocess Communication (IPC) in detail.
b) Explain with neat diagram multithreading models. (8+8=16)

5. a) What is fragmentation? Explain.
b) Consider the following page reference string (with frames size 3)
7,0,1,2,0,3,0,4,2,3,0,3,2,1,2,0,1,7,0.
How many page faults would occur for the page FIFO, LUR and optimal replacement algorithms?
(6+10=16)

6. a) Explain file attributes and operations on file.
b) Explain necessary conditions for deadlock occurrence.
(8+8=16)

7. a) Explain banker’s algorithm.
   b) Describe Resource allocation graph with example.
   c) Explain disk formatting.

(8+4+4=16)

8. Write short notes on any **four**:
   a) Time sharing system.
   b) Co-operating processes.
   c) Process Control Blocking (PCB).
   d) Semaphores.
   e) Segmentation.
   f) Scheduling Queues. (4+4+4+4=16)