Important Instructions to examiners:

1) The answers should be examined by key words and not as word-to-word as given in the model answer scheme.
2) The model answer and the answer written by candidate may vary but the examiner may try to assess the understanding level of the candidate.
3) The language errors such as grammatical, spelling errors should not be given more Importance (Not applicable for subject English and Communication Skills).
4) While assessing figures, examiner may give credit for principal components indicated in the figure. The figures drawn by candidate and model answer may vary. The examiner may give credit for any equivalent figure drawn.
5) Credits may be given step wise for numerical problems. In some cases, the assumed constant values may vary and there may be some difference in the candidate's answers and model answer.
6) In case of some questions credit may be given by judgment on part of examiner of relevant answer based on candidate's understanding.
7) For programming language papers, credit may be given to any other program based on equivalent concept.

1. a) Attempt any three of the following: Marks 12

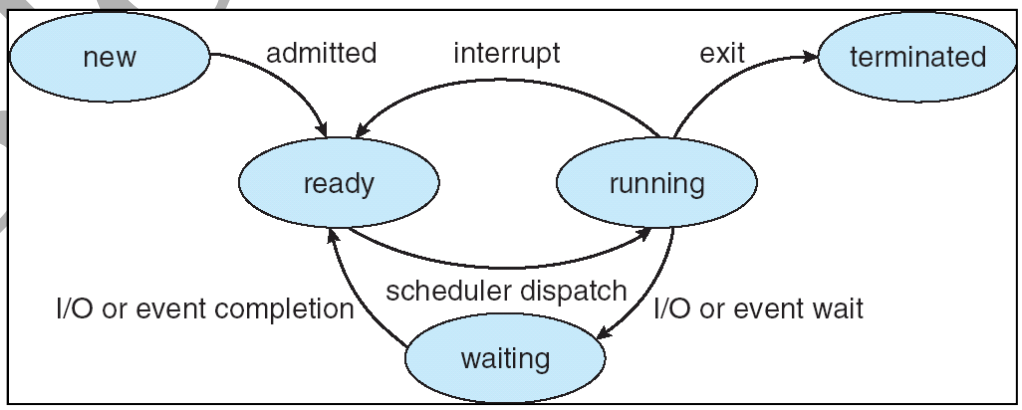
a) What is process? Explain the different process state with the help of state diagram.

(Definition - 1 Mark, Explanation - 1 Mark, Diagram – 2 Marks)

Ans: Process is a program in execution. A process does not mean only program but it could contain some part called as text section. It may contain the current activity represented by the value of the program counter & the contents of CPU register.

Process States A process is typically in one of the three states

1. Running: has the CPU
2. Blocked: waiting for I/O or another thread
3. Ready to run: on the ready list, waiting for the CPU



During the lifespan of a process, its execution status may be in one of four states: (associated with each state is usually a queue on which the process resides)

New: The process being created is available in the new state. It is the new state because the system is not permitted it to enter the ready state due to limited memory available in the ready queue. If some memory becomes available, then the process from the new state will go to ready state.

Ready State: The process which is not waiting for any external event such as I/O operation and which is not running is said to be in ready state. It is not in the running state because some other process is already running. It is waiting for its turn to go to the running state.

Running State: The process which is currently running and has control of the CPU is known as the process in running state. In single user system, there is only one process which is in the running state. In multiuser system, there are multiple processes which are in the running state.

Blocked State: The process is currently waiting on external event such as an I/O operation is said to be in blocked state. After the completion of I/O operation, the process from blocked state enters in the ready state and from the ready state when the process turn will come it will again go to running state.

Terminated / Halted State: The process whose operation is completed, it will go the terminated state from the running state. In halted state, the memory occupied by the process is released.

b) State generation of operating system. Describe any one generation.

(List (three generation)-1 Mark, Explanation of any one generation - 3 Marks)

Ans: Generations of operating system

1. The 1940's - First Generations
2. The 1950's - Second Generation
3. The 1960's - Third Generation
4. The 1980's-The Fourth Generation

- First generation 1945 – 1955 - vacuum tubes, plug boards

The earliest electronic digital computers had no operating systems. Machines of the time were so primitive that programs were often entered one bit at time on rows of mechanical switches (plug boards). Programming languages were unknown (not even assembly languages).

The 1950's - Second Generation

- Second generation 1955 – 1965 - transistors, batch systems

By the early 1950's, the routine had improved somewhat with the introduction of punch cards. The General Motors Research Laboratories implemented the first operating systems in early 1950's for their IBM 701. The system of the 50's generally ran one job at a time. These were called single-stream batch processing systems because programs and data were submitted in groups or batches.

The 1960's - Third Generation

- Third generation 1965 – 1980 - ICs and multiprogramming

The systems of the 1960's were also batch processing systems, but they were able to take better advantage of the computer's resources by running several jobs at once. So operating systems designers developed the concept of multiprogramming in which several jobs are in main memory at once; a processor is switched from job to job as needed to keep several jobs advancing while keeping the peripheral devices in use.

The Fourth Generation

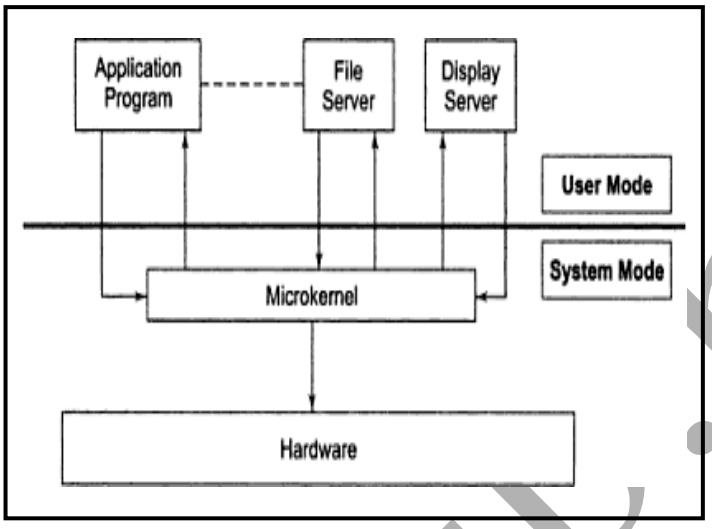
- Fourth generation 1980 – present personal computers

With the development of LSI (Large Scale Integration) circuits, chips, operating system entered in the system entered in the personal computer and the workstation age. Microprocessor technology evolved to the point that it becomes possible to build desktop computers as powerful as the mainframes of the 1970s.

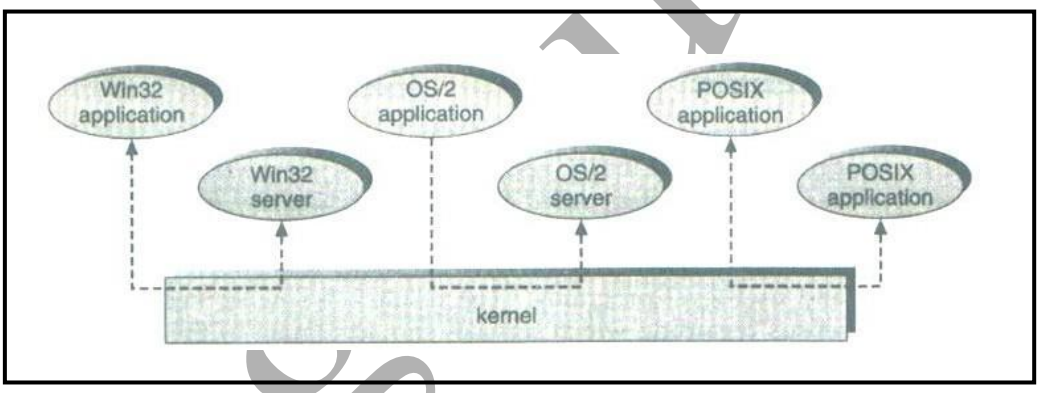
c) Draw Microkernel OS structure.

(Microkernel OS structure any one diagram - 4 Marks)

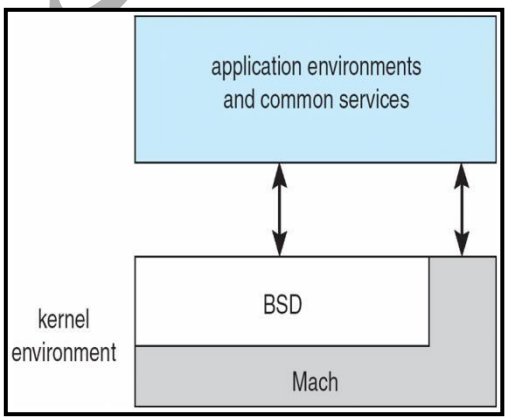
Ans:



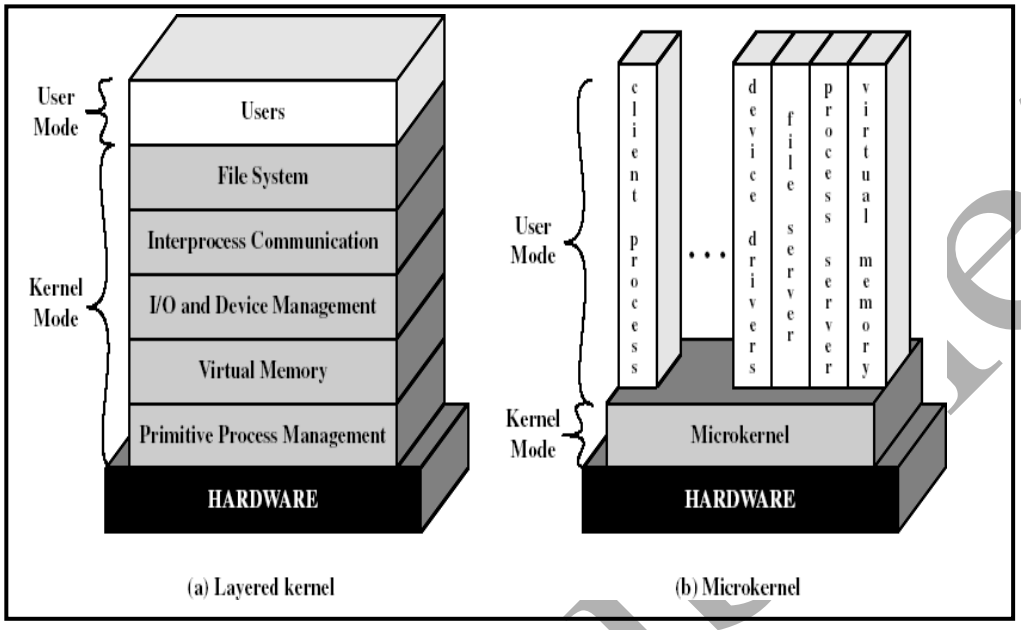
OR



OR



OR



d) Explain First come First serve scheduling algorithm.

(Explanation - 3 Marks, Example – 1 Mark)

Ans:

- The process that requests the CPU first is allocated the CPU first.
- It is nonpreemptive algorithm.
- Can easily be implemented with a FIFO queue.
- When a process enters the ready queue, its PCB is linked onto the tail of the queue.
- When CPU is free, it is allocated to the process at the head of the queue.

As other jobs come in, they are put onto the end of the queue. When running process is blocked by any reason, the first process from the queue will be executed next. When a blocked process becomes ready, it is put at the end of the queue.

Average waiting time is long in FCFS.

Example:

Process	Burst Time
P1	24
P2	3
P3	3

- Suppose that the processes arrive in the order: *P1, P2,P3.*The Gantt Chart for the schedule is:

- Waiting time for P_1 = 0; P_2 = 24; P_3 = 27
- Average waiting time: (0 + 24 + 27)/3 = 17

b) Attempt any one of the following: Marks 06

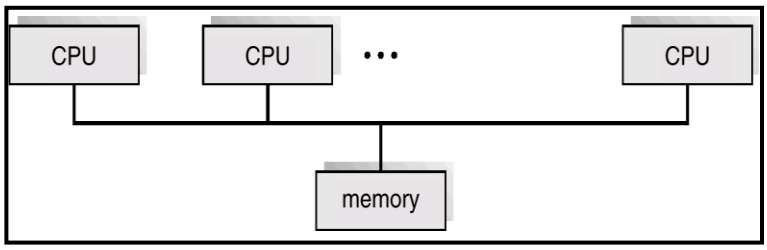
a) Describe following system:

1) Multiprocessor system

2) Batch operating system

Ans: 1) Multiprocessor Systems: (Explanation-2 Marks, Diagram-1 Mark)

- Multiprocessor systems with more than on CPU in close communication.
- Tightly coupled system – processors share memory and a clock; communication usually takes place through the shared memory.



- Advantages of Multiprocessor system:
  - Increased throughput
  - Economical
  - Increased reliability
  - graceful degradation
  - fail-soft systems

Symmetric multiprocessing (SMP)

- Each processor runs and identical copy of the operating system.
- Many processes can run at once without performance deterioration.
- Most modern operating systems support SMP

Asymmetric multiprocessing

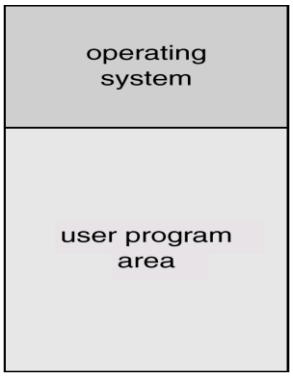
- Each processor is assigned a specific task; master processor schedules and allocated work to slave processors.
- More common in extremely large systems.

2) Batch operating system (Explanation-2 Marks, Any Diagram with batch operating system-1 Mark)

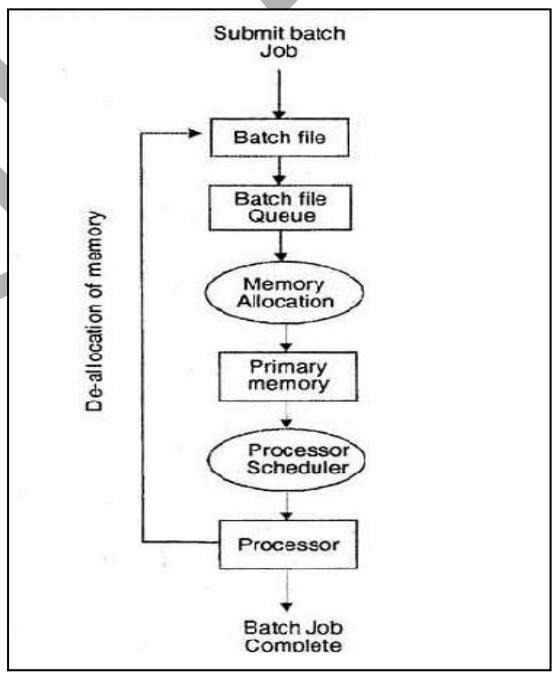
- A batch operating system normally reads a stream of separate jobs (from a card reader. For example), each with its own control cards that predefine to prevent errors and improper use of the computer. It is concerned with the operation and control if I/O devices.
- A batch system is one in which jobs are bundled together with the instruction necessary to allow them to be processed without intervention. Often jobs of a similar nature can be bundledtogether to further increase economy.

- Common input devices were card readers and tape drives. The basic physical layout of the memory of batch job computer is shown in fig.
- The OS was simple, its major task was to transfer control from one job to the next. The job was submitted to the computer operator in form of punch cards. At some later time the output appeared.
- The OS was always resident in memory. Often magnetic tapes and drums were used to store intermediate data and compiled programs.

  Example: Payroll system, stock control and billing systems.



OR



b) Describe the term:

1) Scheduling queues

2) Scheduler

3) Context switch

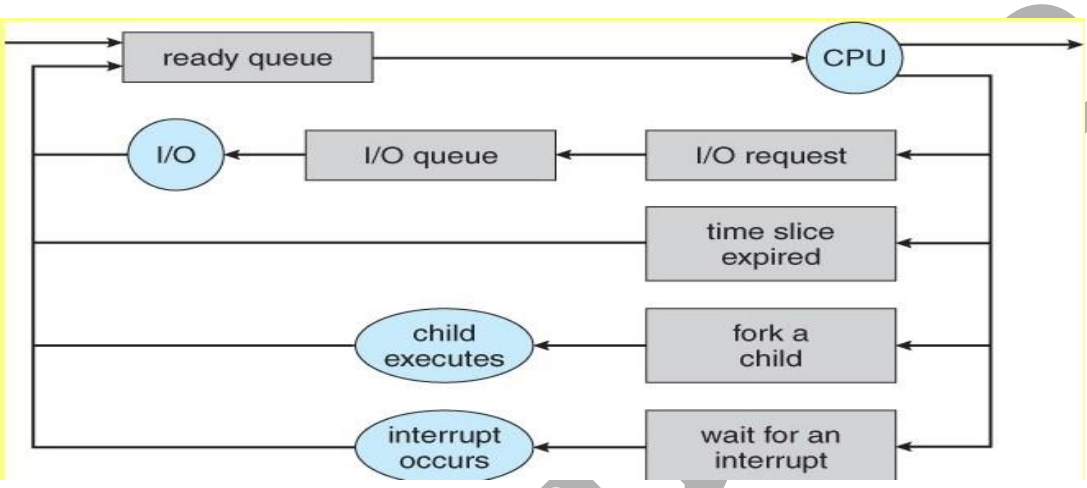
1) Scheduling queue

(Description of queues- 2 Marks)

Ans: **ready** queue : The processes that are residing in main memory and are ready and waiting to execute are kept on a list called the ready queue.

job queue : As processes enter the system they are put into a job queue.

device queue : The list of processes waiting for a particular I/O device is called device queue.



2) Scheduler

(Description of Scheduler- 2 Marks)

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

3) Context switch

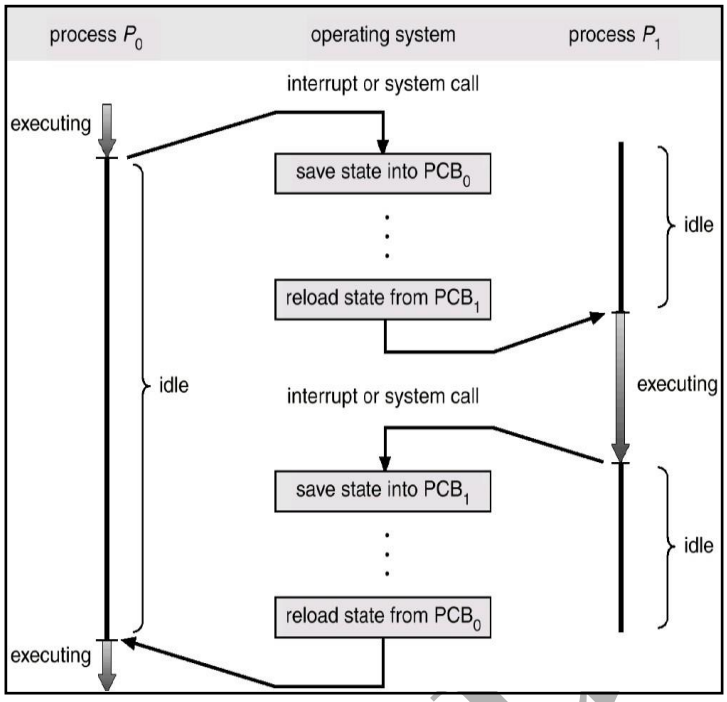
(Description - 2 Marks)

Switch the CPU to another process requires saving the state of current process and loading the saved state for new process. This time is known as a context switch. The context switch represented in PCB.

Saves context of old process in its PCB and loads context of new process schedule to run.

Pure overhead

Depend on hardware support



2. Attempt any four of the following: Marks 16

a) Explain Real time OS with the help of diagram. List its type.

(Explanation - 2 Marks, Any Diagram showing application of real time OS - 1 Mark, List - 1 Mark)

Ans: A real time system has well defined fixed time constraints. Processing should be done within the defined constraints -Hard and Soft real time system.

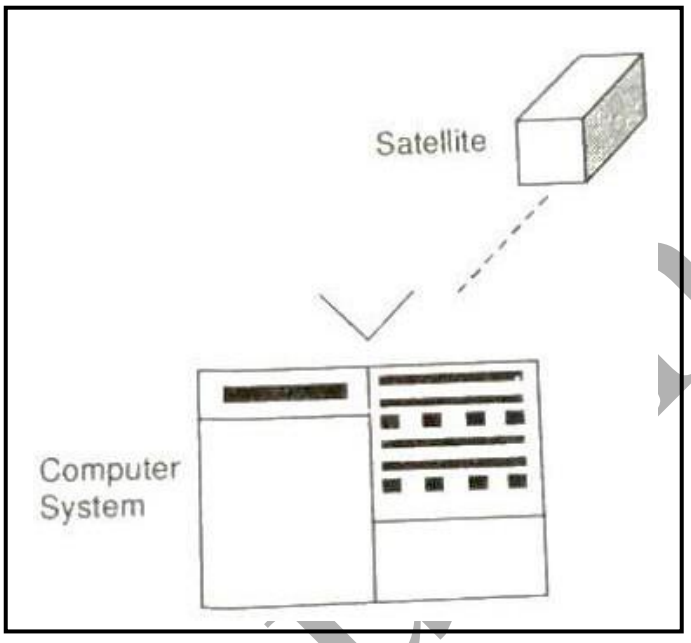
Example – Flight Control System

All tasks in that system must execute on time.

Example: Satellite application of real time OS-

The satellite connected to the computer system sends the digital samples at the rate of 1000 samples per second. The computer system has an application program that stores these samples in a file. The sample sent by the satellite arrives every millisecond to the application. So computer must store or respond the sample in less than 1 millisecond. If the computer does not respond to the sample within this time, the sample will lost.

Some of the examples of Real time systems are: A web server, A word processor, An audio/video media center, A microwave oven, A chess computer.



Satellite Application of real time OS

Types of real time operating system

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

b) List services provided by OS and explain any four of them.

(List – 2 Marks, Any four – 2 Marks (½ Mark each))

Ans: Operating Services:

1. Program execution
2. I/O operations
3. File-system manipulation
4. Communications
5. Error detection
6. Accounting

1. **Program execution**

   The operating system loads the contents (or sections) of a file into memory and begins its execution. A user-level program could not be trusted to properly allocate CPU time.

2. **I/O operations**

   Disks, tapes, serial lines, and other devices must be communicated with at a very low level. The user need only specify the device and the operation to perform on it, while the system converts that request into device- or controller-specific commands. User-level programs cannot be trusted to access only devices they should have access to and to access them only when they are otherwise unused.

3. **File-system manipulation**

   There are many details in file creation, deletion, allocation, and naming that users should not have to per-form. Blocks of disk space are used by files and must be tracked. Deleting a file requires removing the name file information and freeing the allocated blocks. Protections must also be checked to assure proper file access. User programs could neither ensure adherence to protection methods nor be trusted to allocate only free blocks and deallocate blocks on file deletion.

4. **Communications**

   Message passing between systems requires messages to be turned into packets of information, sent to the net-work controller, transmitted across a communications medium, andreassembled by the destination system. Packet ordering and data correction must take place.

   Again, user programs might not coordinate access to the network device, or they might receive packets destined for other processes.

5. **Error detection**

   Error detection occurs at both the hardware and software levels. At the hardware level, all data transfers must be inspected to ensure that data have not been corrupted in transit.

   All data on media must be checked to be sure they have not changed since they were written to the media. At the software level, media must be checked for data consistency; for instance, whether the number of allocated and unallocated blocks of storage match the total number on the device. There, errors are frequently process-independent (for instance, the corruption of data on a disk), so there must be a global program (the operating system) that handles all types

of errors. Also, by having errors processed by the operating system, processes need not contain code to catch and correct all the errors possible on a system.

6. **Accounting**

We may want to keep track at which users use how much and what kind of computer resources. What was the login time for a particular user; is he working on the system right now, what is the process -1 D for the user, all such in formations we can manage using accounting service provided by many multiuser systems. This record keeping may be for the purpose of paying for the system & its operation, or simply for accounting usage statistics.

c) Define PCB (Process Control Block) with suitable diagram.

(Explanation- 2 Marks, Diagram – 2 Marks)

Ans: **PCB** is a record or a data structure that is maintained for each and every process. Every process has one PCB that is associated with it. A PCB is created when a process is created and it is removed from memory when process is terminated.

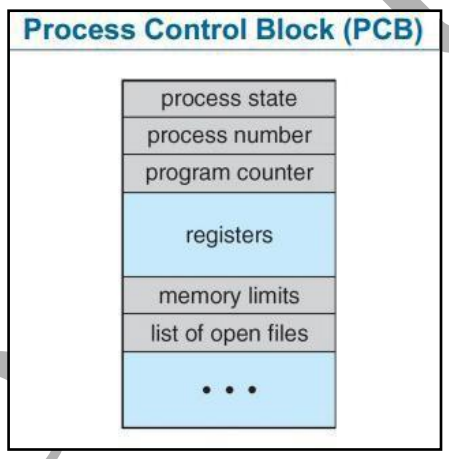
A PCB may contain several types of information depending upon the process to which PCB belongs. The information stored in PCB of any process may vary from process to process.

In general, a PCB may contain information regarding:

1. **Process Number**: Each process is identified by its process number, called process identification number (PID). Every process has a unique process-id through which it is identified. The process-id is provided by the OS. The process id of two process could not be same because ps- id is always unique.
2. **Priority:** Each process is assigned a certain level of priority that corresponds to the relative importance of the event that it services Process priority is the preference of the one process over other process for execution. Priority may be given by the user/system manager or it may be given internally by OS. This field stores the priority of a particular process.
3. **Process State:** This information is about the current state of the process. I.e. whether process is in new, ready, running, waiting or terminated state.
4. **Program Counter:** This contains the address of the next instruction to be executed for this process.
5. **CPU Registers:** CPU registers vary in number and type, depending upon the computer architectures. These include index registers, stack pointers and general purpose registers etc.

When an interrupt occurred, information about the current status of the old process is saved in registers along with the program counters. This information is necessary to allow the process to be continued correctly after the completion of an interrupted process.

6. **CPU Scheduling Information**: This information includes a process priority, pointers to scheduling queues and any other scheduling parameters.
7. **Memory Management Information**: This information may include such information as the value of base and limit registers, the page table or the segment table depending upon the memory system used by operating system.
8. **Accounting**: This includes actual CPU time used in executing a process in order to charge individual user for processor time.
9. **I/O Status**: It includes outstanding I/O request, allocated devices information, pending operation and so on.
10. **File Management:** It includes information about all open files, access rights etc.



d) Define following terms:

1) Dead line **2) Response time**

3) Throughput **4) Turnaround time**

(Explanation of each – 1 Mark)

Ans:

1) Dead line- It is defined as the time limit to complete the process execution.

2) Response time – time from the submission of a request until the first response is produced (time it takes to start responding, but not the time it takes to output that response)
3) Throughput – Number of processes that are completed per unit time
4) Turnaround time – the interval from the time of submission of a process to the time of completion. (Sum of the periods spent waiting to get into the memory, waiting in the ready queue, executing on CPU and doing I/O)

e) Describe function of OS in detail.
(Any four functions - 1 Mark each)

Ans: The major functions of an operating system are:

1. **Resource Management :**
   This function of OS allocates computer resources such as CPU time, main memory, secondary storage and input and output devices for use.
2. **Data management**: It observes input and output of the data and their location, storage and retrieval.
3. **Task management**: Task is a collection of one or more related programs and their data. This function prepares, schedules, controls and monitors jobs submitted for execution to ensure the most efficient processing.
4. **Allocation of Resources:** Handles system resources such as computer's memory and sharing of the central processing unit (CPU) time by various applications or peripheral devices
5. **Communication between User and Computer** : Provides a user interface, e.g. command line, graphical user interface (GUI)
6. Operating system enables startup application programs. OS must have text editor, a translator and an editor.
7. Operating system provides number of services such as for the programmer it provides utilities ie debugger, editors, file management which refers to the way that the operating system manipulates, stores, retrieves and saves data. It interprets the commands executed by the user. It handle disk input/output settings.

OR

1. **Process Management** – Managing the programs that are running.
2. **Memory Management** – Managing and rationing the memory between processes and data.

3. **Storage Management** – Managing the permanent Storage of data on disks or other media
4. **I/O Management** – Managing the input and output
5. **Device / Resource Management** – Managing devices and resources and allowing the users to share the resources
6. **Security and Protection** – Securing the system against possible unauthorized access to data or any other entity. Protecting the parts of the system against damage.
7. **Booting the System and getting it ready to work.**
8. **Data communications –** Providing interface to connect to other computers or allowing others to connect

f) Describe creation and termination operation on process.

(Description of Process creation – 2 Marks, Process Termination -2 Marks)

Ans: Process creation

Create Process Operating system creates a new process with the specified or default attributes and identifier. A process may create several new subprocesses.

Syntax for creating new process is:

CREATE (processed, attributes)

Two names are used in the process they are parent process and child process.

Parent process is a creating process. Child process is created by the parent process. Child process may create another subprocess. So it forms a tree of processes. When operating system issues a CREATE system call, it obtains a new process control block from the pool of free memory, fills the fields with provided and default parameters, and insert the PCB into the ready list. Thus it makes the specified process eligible to run the process.

When a process is created, it requires some parameters. These are priority, level of privilege requirement of memory, access right, memory protection information etc. Process will need certain resources, such as CPU time, memory, files and I/O devices to complete the operation. When process creates a subprocess, that subprocess may obtain its resources directly from the operating system. Otherwise it uses the resources of parent process.

When a process creates a new process, two possibilities exist in terms of execution.

1. The parent continues to execute concurrently with its children.

2. The parent waits until some or all of its children have terminated.

For address space, two possibilities occur:

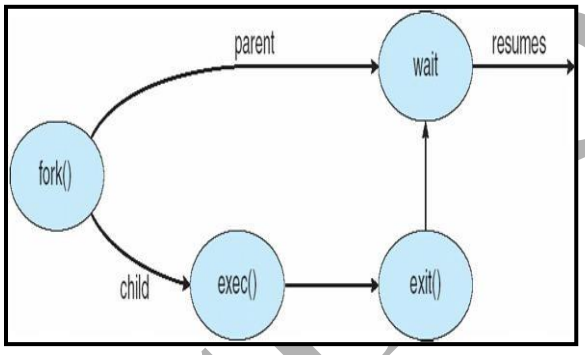
1. The child process is a duplicate of the parent process.

2. The child process has a program loaded into it.

For example: UNIX examples

i) fork system call creates new process

ii) exec system call used after a **fork** to replace the process' memory space with a new program



Process Termination:
Process executes last statement and asks the operating system to delete it (**exit**)

i) Output data from child to parent (via **wait**)

ii) Process' resources are deallocated by operating system

DELETE system call is used for terminating a process. A process may delete itself or by another process. A process can cause the termination of another process via an appropriate system call. The operating system reacts by reclaiming all resources allocated to the specified process, closing files opened by or for the process. PCB is also removed from its place of residence in the list and is returned to the free pool. The DELETE service is normally invoked as a part of orderly program termination. Following are the resources for terminating the child process by parent process.

3. Attempt any four of the following: Marks 16

a) What is system call? Explain use of any two categories of system call.

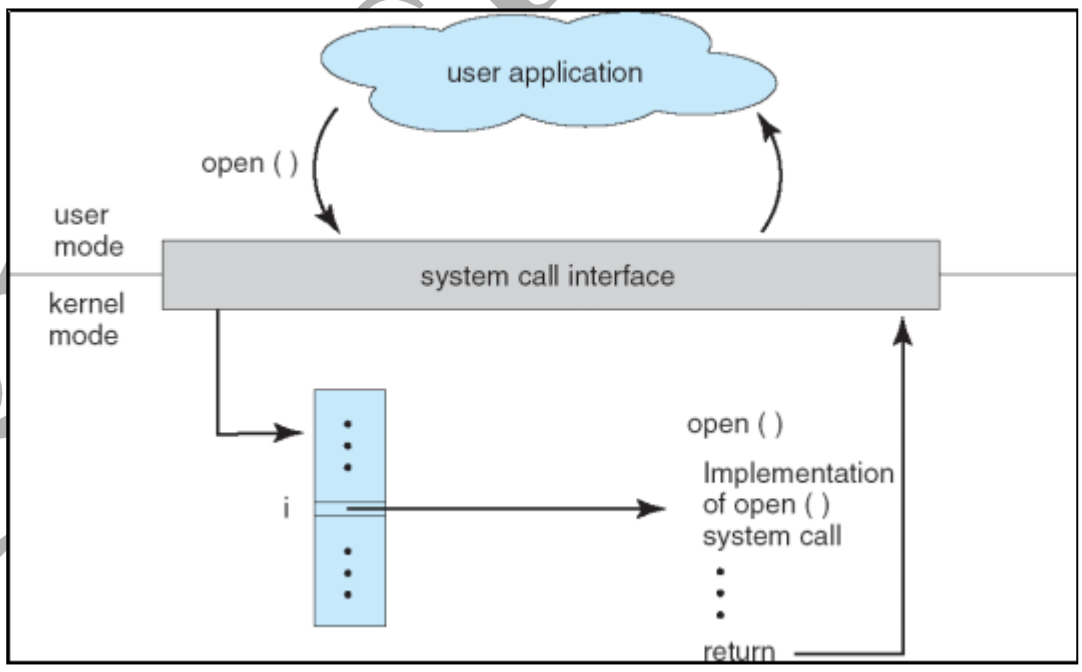
(System call Description - 2 Marks, Use of any two System calls categories – 2 Marks (Each category – 1 Mark))

Ans: System Calls: System calls are programming interface to the services provided by the operating system.

Implementation:

1. Number to number the system calls, each system call associated with a particular number.
2. System call interface maintains a table indexed according to these numbers.
3. The system call interface invokes intended system call in operating system kernel & returns status of the system call and any return values.
4. The caller needs to know nothing about how the system call is implemented. Just needs to obey.

   API and understand what OS will do as a result call.
5. Most details of operating system interface hidden from programmers by API. It is managed by run-time support library.



System calls can be grouped into the following major categories.

a. Process or Job control

b. File Management

c. Device Management

d. Information Maintenance

System calls related to process control: End, Abort Load, Execute Create process, Terminate process Ready process, Dispatch process Suspend, Resume Get Process attribute, set attribute Wait for time Wait event, signal event

System calls Related to File management: Create file, delete file Open file , Close file Create directory Read, write, Reposition Get file attribute , set file attribute Create a link Change the working directory

System calls Related to Device Management: Request a device, Release a device Read, Write, Reposition Get device attribute, set device attribute

System calls Related to Information Maintenance: Get Time or Date, Set Time or Date Get System data, Set system data Get process, file or device attributes Set process, file or Device attributes.

b) What is multithreading? Explain with suitable diagram.

(Definition - 1 Mark, Explanation – 1 Mark, Diagram – 2 Marks)

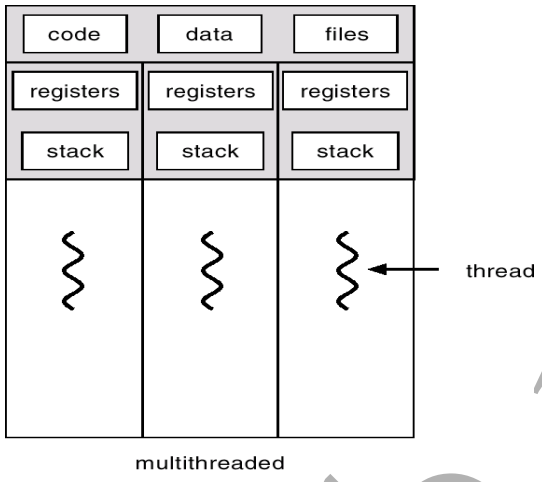
Ans: Multithreading: Refers to the ability to an O.S to support multiple threads of execution within a single process. In a multi-threaded environment multiple processes and multiple threads can be considered as in case of multiuser O.S. such as UNIX.

System provides support to both user and kernel threads, resulting in different types of multithreading models

1) Many to One model

2) One to One model

3) Many to Many model



Threads share the memory and the resources of the process to which they belong. The benefit of code sharing is that it allows an application to have several different threads of activity all within the same address space.

c) Explain Round robin algorithm with suitable example.

(Explanation - 2 Marks, Example - 2 Marks)

Ans: Round Robin Scheduling

Each process gets a small unit of CPU time (**time quantum**), usually 10-100 milliseconds. After this time has elapsed, the process is preempted and added to the end of the ready queue.

If there are *n* processes in the ready queue and the time quantum is *q*, then each process gets 1/*n* of the CPU time in chunks of at most *q* time units at once. No process waits more than (*n*-1)*q* time units.

Performance

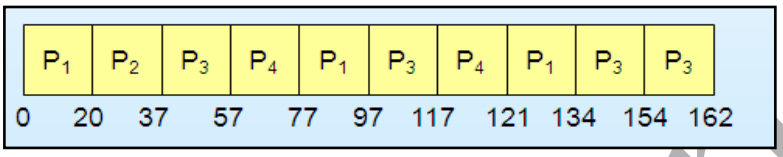
q large $\Rightarrow$ FCFS

q small $\Rightarrow$ *q* must be large with respect to context switch, otherwise overhead is too high

Example of RR with Time Quantum = 20

Process	Burst Time
P_1	53
P_2	17
P_3	68
P_4	24

The Gantt chart is:



Typically, higher average turnaround than SJF, but better response

d) Explain any four operations which can be performed on file.

(Any four operation - 1 Mark each)

Ans: File Operations

A file is an abstract data type. To define a file properly, we need to consider the operations that can be performed on files.

Six basic file operations. The OS can provide system calls to create, write, read, reposition, delete, and truncate files.

Creating a file. Two steps are necessary to create a file.

1. Space in the file system must be found for the file.

2. An entry for the new file must be made in the directory.

Writing a file. To write a file, we make a system call specifying both the name of the file and the information to be written to the file. The system must keep a write pointer to the location in the file where the next write is to take place. The write pointer must be updated whenever a write occurs.

Reading a file. To read from a file, we use a system call that specifies the name of the file and where (in memory) the next block of the file should be put. The system needs to keep a read pointer to the location in the file where the next read is to take place.

Because a process is usually either reading from or writing to a file, the current operation location can be kept as a per-process current-file-position pointer.

Both the read and write operations use this same pointer, saving space and reducing

system complexity.

Repositioning within a file. The directory is searched for the appropriate entry, and the current-file-position pointer is repositioned to a given value. Repositioning within a file need not involve any actual I/O. This file operation is also known as a file seek.

Deleting a file. To delete a file, we search the directory for the named file. Having found the associated directory entry, we release all file space, so that it can be reused by other files, and erase the directory entry.

The five operations described comprise only the minimal set of required file operations. More commonly, we shall also want to edit the file and modify its contents. A special case of editing a file is appending new information at the end of the file. Copies of the file can also be created, and since files are named object, renaming an existing file may also be needed. If the file is a binary object format, we may also want to execute it. Also of use are facilities to lock sections of an open file for multiprocess access, to share sections, and even to map sections into memory or virtual-memory systems.

This last function allows a part of the virtual address to be logically associated with section of a file. Reads and writes to that memory region are then treated as reads and writes to the file. To that memory region are then treated as reads and writes to the file, greatly simplifying file use.

Truncating a file. The user may want to erase the contents of a file but keep its attributes. Rather than forcing the user to delete the file and then recreate it, this function allows all attributes to remain unchanged (except for file length) but lets the file be reset to length zero and its file space released.

e) Describe architecture of UNIX OS with the help of diagram.

(Description - 2 Marks, Diagram -2 Marks)

Ans: The kernel of UNIX is the hub of the operating system: it allocates time and memory to programs and handles the file store and communications in response to system calls. As an illustration of the way that the shell and the kernel work together, suppose a user types rm my file (which has the effect of removing the file **my file**). The shell searches the file store for thefile containing the program rm, and then requests the kernel, through system calls, to execute the program rm on my file. When the process rm my file has finished running, the shell then returns

the UNIX prompt % to the user, indicating that it is waiting for further commands.

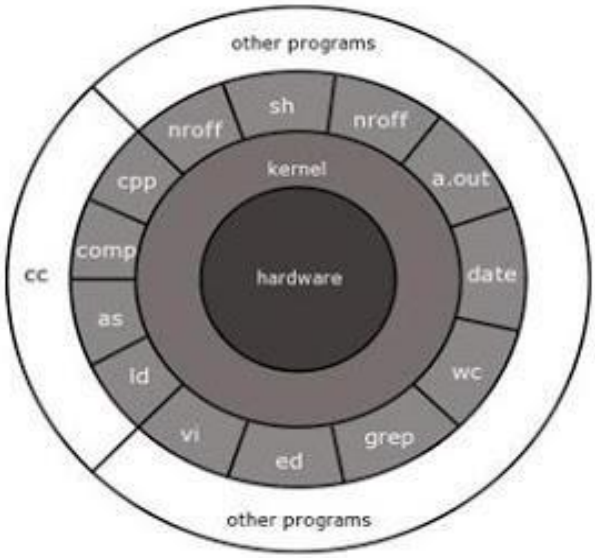
Amongst the functions performed by the kernel are:

- Managing the machine's memory and allocating it to each process.
- Scheduling the work done by the CPU so that the work of each user is carried out as efficiently as is possible.
- Organizing the transfer of data from one part of the machine to another.
- Accepting instructions from the shell and carrying them out.
- Enforcing the access permissions that are in force on the file system

The shell:

The shell acts as an interface between the user and the kernel. When a user logs in, the login program checks the username and password, and then starts another program called the shell. The shell is a command line interpreter (CLI). It interprets the commands the user types in and arranges for them to be carried out. The commands are themselves programs: when they terminate, the shell gives the user another prompt (% on our systems). The user can customise his/her own shell, and users can use different shells on the same machine. The shell keeps a list of the commands you have typed in. If you need to repeat a command, use the cursor keys to scroll up and down the list or type history for a list of previous commands. You can use any one of these shells if they are available on your system. And you can switch between the different shells once you have found out if they are available.

- **Bourne shell (sh)**
- **C shell (csh)**
- **TC shell (tcsh)**
- **Korn shell (ksh)**



4. a) Attempt any three of the following: **Marks 12**

a) List components of OS. Explain any one in detail.

(List - 1 Mark, Description of any one - 3 Marks)

Ans: List of System Components:

1. Process management
2. Main memory management
3. File management
4. I/O system management
5. Secondary storage management

Process Management

The operating system manages many kinds of activities ranging from user programs to system programs like printer spooler, name servers, file server etc. Each of these activities is encapsulated in a process. A process includes the complete execution context (code, data, PC, registers, OS resources in use etc.).

The basic unit of software that the operating system deals with in scheduling the work done by the processor is either a **process** or a **thread**, depending on the operating system.

It's tempting to think of a process as an application, but that gives an incomplete picture of how processes relate to the operating system and hardware. The application you see (word processor or spreadsheet or game) is, indeed, a process, but that application may cause several other processes to begin, for tasks like communications with other devices or other computers. There are also numerous processes that run without giving you direct evidence that they ever exist. A process, then, is software that performs some action and can be controlled by a user, by other applications or by the operating system.

It is processes, rather than applications, that the operating system controls and schedules for execution by the CPU. In a single-tasking system, the schedule is straightforward. The operating system allows the application to begin running, suspending the execution only long enough to deal with interrupts and user input.

The five major activities of an operating system in regard to process management are

Creation and deletion of user and system processes.

Suspension and resumption of processes.

A mechanism for process synchronization.

A mechanism for process communication.

A mechanism for deadlock handling.

Main-Memory Management

Services provided under Memory Management are directed to keeping track of memory and allocating/deallocating it to various process. The OS keeps a list of free memory locations.Before a program is loaded in the memory from the disk, this MM consults the free list, allocates the memory to the process, depending upon the progrsm size and updates the list of free memory.

Primary-Memory or Main-Memory is a large array of words or bytes. Each word or byte has its own address. Main-memory provides storage that can be access directly by the CPU. That is to say for a program to be executed, it must in the main memory.

The major activities of an operating in regard to memory-management are:

Keep track of which part of memory are currently being used and by whom.

Decide which process are loaded into memory when memory space becomes available

Allocate and de allocate memory space as needed.

File Management A file is a collected of related information defined by its creator. Computer can store files on the disk (secondary storage), which provide long term storage. Some examples of storage media are magnetic tape, magnetic disk and optical disk. Each of these media has its own properties like speed, capacity, data transfer rate and access methods. A file systems normally organized into directories to ease their use. These directories may contain files and other directions. The five main major activities of an operating system in regard to file management are

1. The creation and deletion of files.
2. The creation and deletion of directions.
3. The support of primitives for manipulating files and directions.
4. The mapping of files onto secondary storage.
5. The backup of files on stable storage media.

I/O System Management I/O subsystem hides the peculiarities of specific hardware devices from the user. Only the device driver knows the peculiarities of the specific device to whom it is assigned.

Secondary-Storage Management

Systems have several levels of storage, including primary storage, secondary storage and cache storage. Instructions and data must be placed in primary storage or cache to be referenced by a running program. Because main memory is too small to accommodate all data and programs, and its data are lost when power is lost, the computer system must provide secondary storage to back up main memory. Secondary storage consists of tapes, disks, and other media designed to hold information that will eventually be accessed in primary storage (primary, secondary, cache) is ordinarily divided into bytes or words consisting of a fixed number of bytes. Each location in storage has an address; the set of all addresses available to a program is called an address space. The three major activities of an operating system in regard to secondary storage management are:

1. Managing the free space available on the secondary-storage device
2. Allocation of storage space when new files have to be written.
3. Scheduling the requests for memory access.

b) Describe four conditions for dead locking.

(Four conditions - 1 Mark each)

Ans: **1. Mutual Exclusion:** The resources involved are non-shareable.

At least one resource (thread) must be held in a non-shareable mode, that is, only one process at a time claims exclusive control of the resource. If another process requests that resource, the requesting process must be delayed until the resource has been released.

2. Hold and Wait: Requesting process hold already, resources while waiting for requested resources.

There must exist a process that is holding a resource already allocated to it while waiting for additional resource that are currently being held by other processes.

3. No-Preemptive: Resources already allocated to a process cannot be preempted.

Resources cannot be removed from the processes are used to completion or released voluntarily by the process holding it.

4. Circular Wait

The processes in the system form a circular list or chain where each process in the list is waiting for a resource held by the next process in the list.

c) Describe types of scheduler used in scheduling.

(Types - 1 Mark, Description - 1 Mark each)

Ans: Schedulers are special system softwares which handles process scheduling in various ways. Their main task is to select the jobs to be submitted into the system and to decide which process to run. Schedulers are of three types

- Long Term Scheduler
- Short Term Scheduler
- Medium Term Scheduler

Long Term Scheduler

It is also called job scheduler. Long term scheduler determines which programs are admitted to the system for processing. Job scheduler selects processes from the queue and loads them into memory for execution. Process loads into the memory for CPU scheduling. The primary objective of the job scheduler is to provide a balanced mix of jobs, such as I/O bound and processor bound. It also controls the degree of multiprogramming. If the degree of multiprogramming is stable, then the average rate of process creation must be equal to the average departure rate of processes leaving the system.

On some systems, the long term scheduler may not be available or minimal. Time-sharingoperating systems have no long term scheduler. When process changes the state from new to ready, then there is use of long term scheduler.

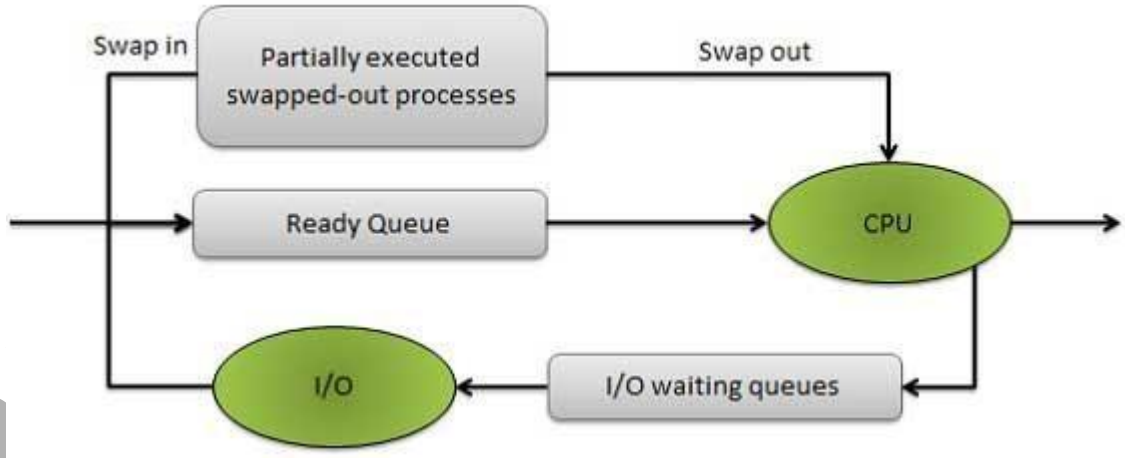
Short Term Scheduler

It is also called CPU scheduler. Main objective is increasing system performance in accordance with the chosen set of criteria. It is the change of ready state to running state of the process. CPU scheduler selects process among the processes that are ready to execute and allocates CPU to one of them.

Short term scheduler also known as dispatcher, execute most frequently and makes the fine grained decision of which process to execute next. Short term scheduler is faster than long term scheduler.

Medium Term Scheduler

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



Running process may become suspended if it makes an I/O request. Suspended processes cannot make any progress towards completion. In this condition, to remove the process from memory and make space for other process, the suspended process is moved to the secondary storage. This

process is called swapping, and the process is said to be swapped out or rolled out. Swapping may be necessary to improve the process mix.

d) Compare paging and segmentation.

(Any four points - 1 Mark each (any other valid point shall be considered))

Ans:

Paging	Segmentation
Paging divides the computer's primary memory into fixed-size units called page frames, and the program's address space into pages of the same size.	Segmentation is the only memory management technique that does not provide the user's program with a 'linear and contiguous address space.".
The hardware memory management unit maps pages to frames.	Segments are areas of memory that usually correspond to a logical grouping of information such as a code procedure or a data array.
The physical memory can be allocated on a page basis while the address space appears contiguous.	Segments require hardware support in the form of a segment table which usually contains the physical address of the segmentin memory, its size, and other data such as access protection bits and status.
Pages are used for swapping or managing memory.	Small pieces called segments are used for memory management.
Page is indicated by its number and offset.	Segment is indicated by segment number and its offset
Page table is formed	Segment table is formed.
Do not support user's view of memory.	Supports user's view of memory.

b) Attempt any one of the following: Marks 06

a) **What are the different file allocation method? Explain any one in detail with example.**

(Listing of methods - 1 Mark, Explanation of any one - 3 Marks, Example with diagram - 2 Marks)

Ans: An allocation method refers to how disk blocks are allocated for files. Different file allocation methods are:

(a) Contiguous allocation

(b) Linked allocation

(c) Indexed allocation

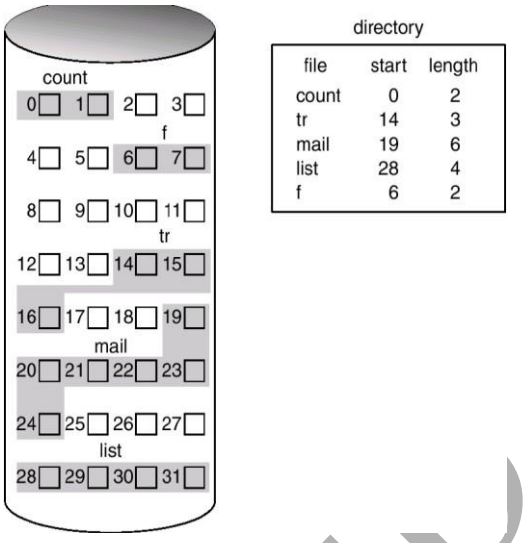
(a) Contiguous Allocation

The contiguous allocation method requires each file to occupy a set of contiguous address on the disk. Disk addresses define a linear ordering on the disk. Notice that, with this ordering, accessing block b+1 after block b normally requires no head movement. When head movement is needed (from the last sector of one cylinder to the first sector of the next cylinder), it is only one track. Thus, the number of disk seeks required for accessing contiguous allocated files in minimal, as is seek time when a seek is finally needed. Contiguous allocation of a file is defined by the disk address and the length of the first block. If the file is n blocks long, and starts at location b, then it occupies blocks b, b+1, b+2, …, b+n-1. The directory entry for each file indicates the address of the starting block and the length of the area allocated for this file.

The difficulty with contiguous allocation is finding space for a new file. If the file to be created is n blocks long, then the OS must search for n free contiguous blocks. First-fit, best-fit, and worst-fit strategies (as discussed in Chapter 4 on multiple partition allocation) are the most common strategies used to select a free hole from the set of available holes. Simulations have shown that both first-fit and best-fit are better than worst-fit in terms of both time storage utilization. Neither first-fit nor best-fit is clearly best in terms of storage utilization, but first-fit is generally faster.

These algorithms also suffer from external fragmentation. As files are allocated and deleted, the free disk space is broken into little pieces. External fragmentation exists when enough total disk space exists to satisfy a request, but this space not contiguous; storage is fragmented into a large number of small holes.

Another problem with contiguous allocation is determining how much disk space is needed for a file. When the file is created, the total amount of space it will need must be known and allocated. How does the creator (program or person) know the size of the file to be created. In some cases, this determination may be fairly simple (e.g. copying an existing file), but in general the size of an output file may be difficult to estimate.



Contiguous Allocation

(b) Linked allocation

The problems in contiguous allocation can be traced directly to the requirement that the spaces be allocated contiguously and that the files that need these spaces are of different sizes. These requirements can be avoided by using linked allocation.

In linked allocation, each file is a linked list of disk blocks. The directory contains a pointer to the first and (optionally the last) block of the file. For example, a file of 5 blocks which starts at block 4, might continue at block 7, then block 16, block 10, and finally block 27. Each block contains a pointer to the next block and the last block contains a NIL pointer. The value -1 may be used for NIL to differentiate it from block 0.

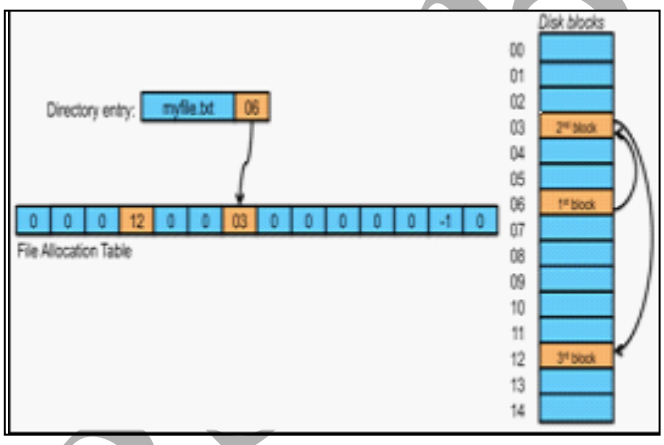
With linked allocation, each directory entry has a pointer to the first disk block of the file. This pointer is initialized to nil (the end-of-list pointer value) to signify an empty file. A write to a file removes the first free block and writes to that block. This new block is then linked to the end of the file. To read a file, the pointers are just followed from block to block.

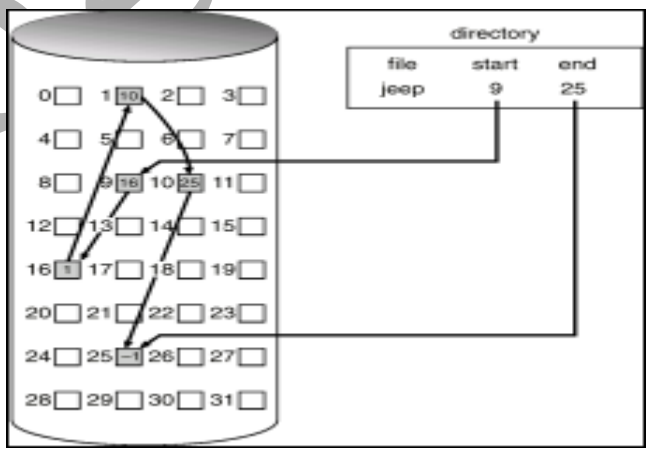
There is no external fragmentation with linked allocation. Any free block can be used to satisfy a request. Notice also that there is no need to declare the size of a file when that file is created. A file can continue to grow as long as there are free blocks.

Linked allocation, does have disadvantages, however. The major problem is that it is inefficient to support direct-access; it is effective only for sequential-access files. To find the ith block of a file, it must start at the beginning of that file and follow the pointers until the ith block is reached.

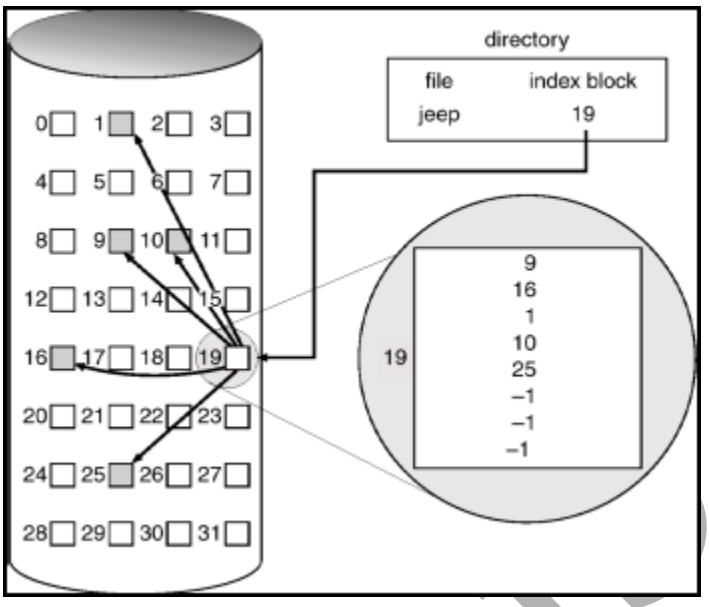
Note that each access to a pointer requires a disk read.

Another severe problem is reliability. A bug in OS or disk hardware failure might result in pointers being lost and damaged. The effect of which could be picking up a wrong pointer and linking it to a free block or into another file.





(c) Indexed allocation



The indexed allocation method is the solution to the problem of both contiguous and linked allocation. This is done by bringing all the pointers together into one location called the index block. Of course, the index block will occupy some space and thus could be considered as an overhead of the method.

In indexed allocation, each file has its own index block, which is an array of disk sector of addresses. The ith entry in the index block points to the ith sector of the file. The directory contains the address of the index block of a file. To read the ith sector of the file, the pointer in theith index block entry is read to find the desired sector.

Indexed allocation supports direct access, without suffering from external fragmentation. Any free block anywhere on the disk may satisfy a request for more space.

b) Consider the following set of processes with arrival time as below:

Process	Priority	Arrival Time	CPU Burst(in ms.)
P1	2	0	5
P2	3	1	6
P3	4	2	7
P4	1	3	3

Find out average waiting by using

1) Non-Preemptive SJF

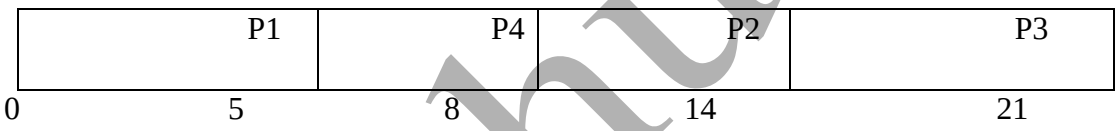
2) Preemptive SJF

3) FCFS

(Gantt chart – 1 Mark each, Average waiting time - 1 Mark for each method)

Find out average waiting time by using

1) Non-preemptive SJF

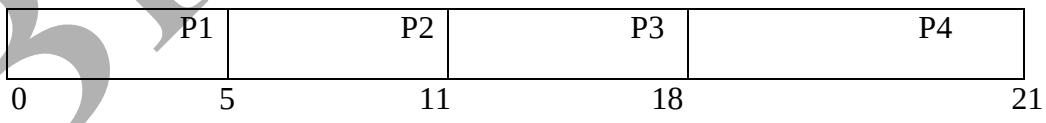


AWT= (0-0)+(8-1)+(14-2)+(5-3)/4 = 5.25 ms

2) Preemptive SJF

AWT= (0-0)+(8-1)+(14-2)+(5-3)/4 = 5.25 ms

3) FCFS



AWT = (0-0)+(5-1)+(11-2)+(18-3)/4=7 ms

5. Attempt any two of the following: Marks 16

a) Explain:

1) Monolithic OS structure

2) Layered OS structure

(Structure Diagram -2 Marks each, Structure Explanation -2 Marks each)

Ans: Layered OS structure

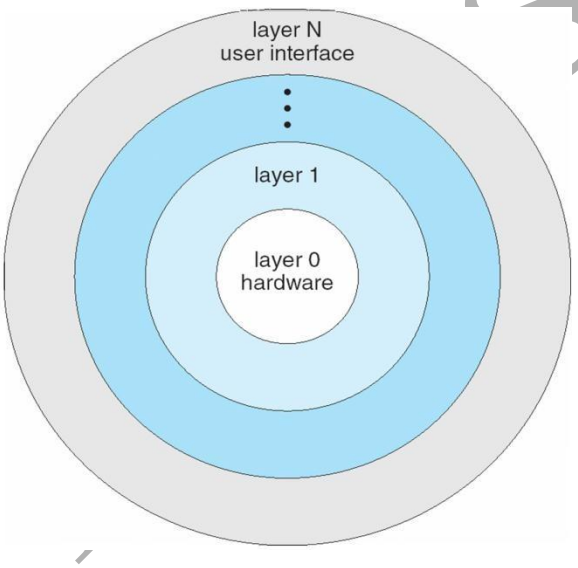
The operating system is divided into a number of layers (levels), each built on top of lower layers. The bottom layer (layer 0), is the hardware; the highest (layer N) is the user interface.

With modularity, layers are selected such that each uses functions (operations) and services of only lower-level layers.

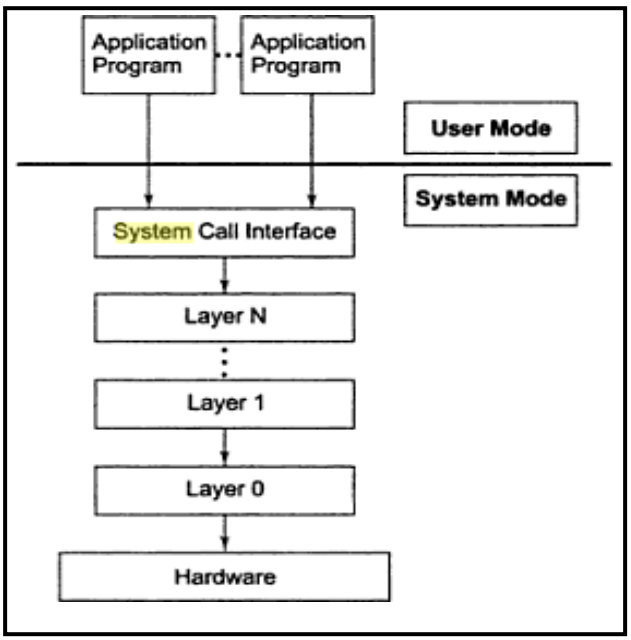
Main advantage of the layered approach is modularity.

The major difficulty with the layered approach invokes the careful definition of the layers, because a layer can use only those layers below it

Layered Operating System.(Any one diagram from the following)

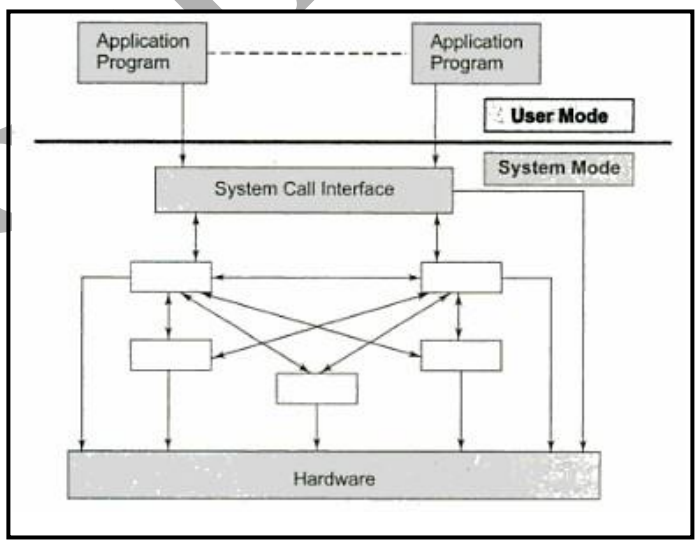


OR

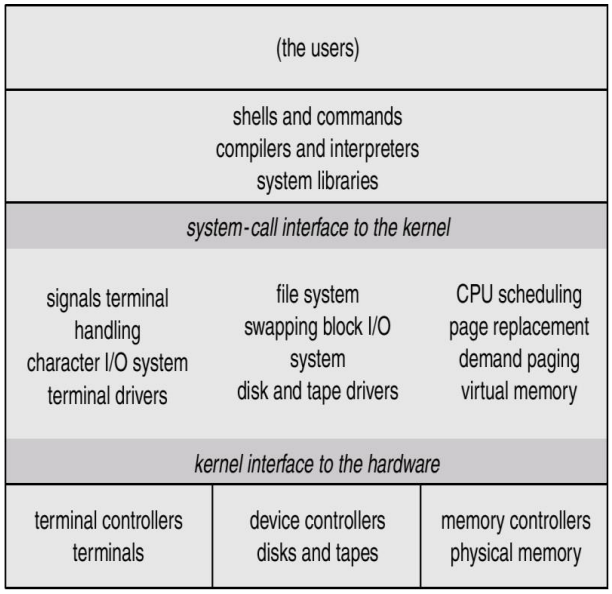


Monolithic Systems: This approach well known as "The Big Mess". The structure is that there is no structure. The operating system is written as a collection of procedures, each of which can call any of the other ones whenever it needs to. When this technique is used, each procedure in the system has a well-defined interface in terms of parameters and results, and each one is free to call any other one, if the latter provides some useful computation that the former needs. For constructing the actual object program of the operating system when this approach is used, one compiles all the individual procedures, or files containing the procedures, and then binds them all together into a single object file with the linker. In terms of information hiding, there is essentially none- every procedure is visible to every other one i.e. opposed to a structure containing modules or packages, in which much of the information is local to module, and only officially designated entry points can be called from outside the module. However, even in Monolithic systems, it is possible to have at least a little structure. The services like system calls provide by the operating system are requested by putting the parameters in well-defined places, such as in registers or on the stack, and then executing a special trap instruction known as a kernel call or supervisor call.

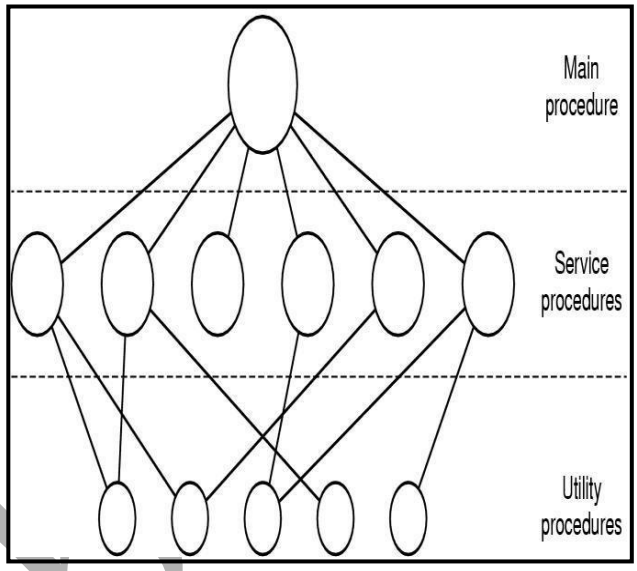
Simple structuring model for a monolithic system.(Any one diagram from the following)



OR



OR

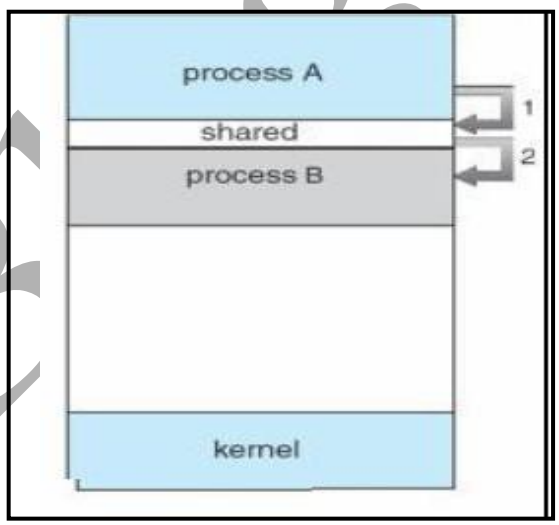


b) With suitable diagram explain interprocess communication model.

(Two models – 8 Marks (Each model - 4 Marks))

Ans: Inter-process communication: Cooperating processes require an Inter- process communication (IPC) mechanism that will allow them to exchange data and information. There are two models of IPC

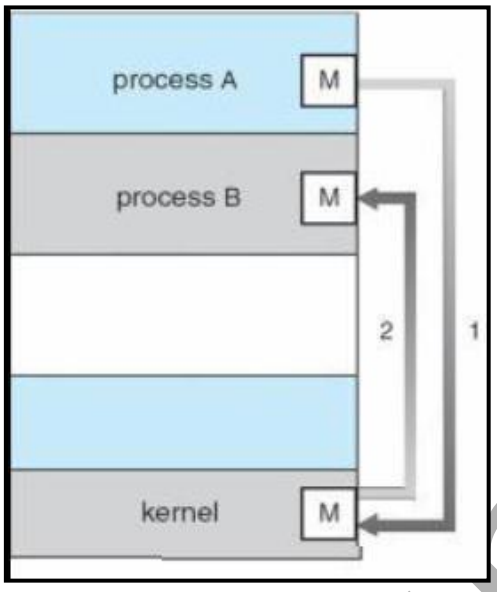
a. Shared memory: (Diagram - 2 Marks, Explanation – 2 Marks)



In this a region of the memory residing in an address space of a process creating a shared memory segment can be accessed by all processes who want to communicate with other processes. All the processes using the shared memory segment should attach to the address space of the shared memory. All the processes can exchange information by reading and/or writing data in shared memory segment. The form of data and location are determined by these processes who

want to communicate with each other. These processes are not under the control of theoperating system. The processes are also responsible for ensuring that they are not writing to the same location simultaneously. After establishing shared memory segment, all accesses to the shared memory segment are treated as routine memory access and without assistance of kernel.

b. Message Passing: (Diagram - 2 Marks, Explanation – 2 Marks)



In this model, communication takes place by exchanging messages between cooperating processes. It allows processes to communicate and synchronize their action without sharing the same address space. It is particularly useful in a distributed environment when communication process may reside on a different computer connected by a network. Communication requires sending and receiving messages through the kernel. The processesthat want to communicate with each other must have a communication link between them. Between each pair of processes exactly one communication link.

***c)* What is partitioning? Explain concept of variable memory partitioning with example.**
(Description of partitioning- 2 Marks, Concept of variable partitioning- 3 Marks, Example with Diagram - 3 Marks)

Ans: Partitioning is part of Memory Management System. A new magnetic disk is a blank slate: it is just a platter of a magnetic recording material. Before a disk can store data, it must be divided into sectors that the disk controller can read and write, this is called low level or physical formatting. Partitioning types: fixed partition (static partitions) and variable partition (dynamic partitions).

• Partition main memory into a set of non-overlapping memory regions called partitions.

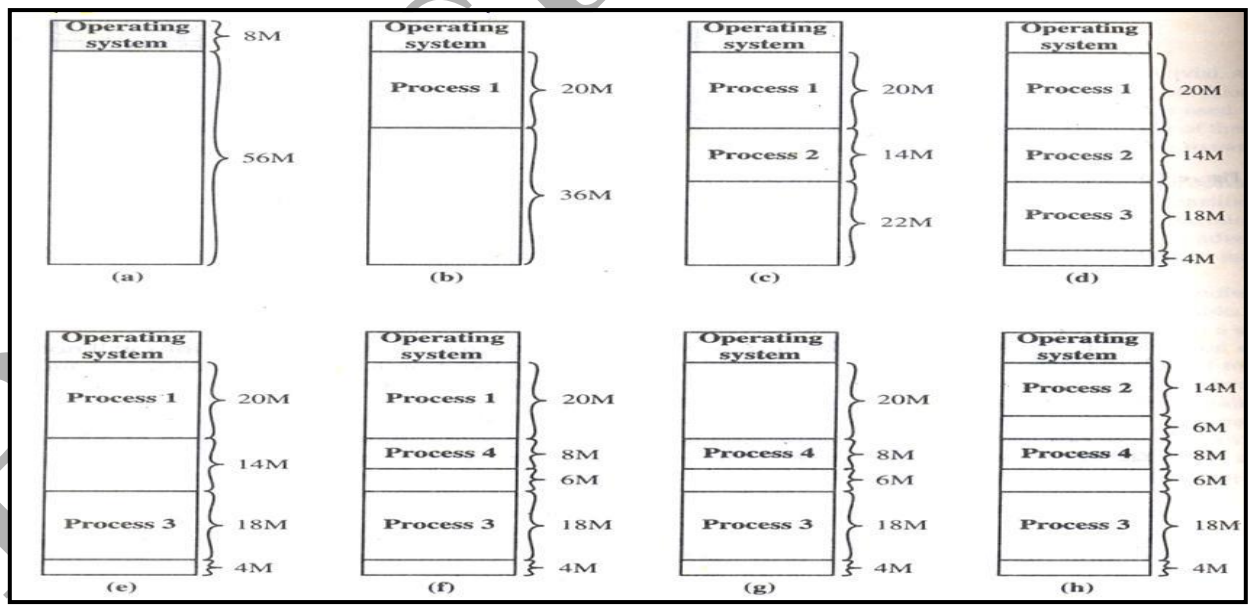
Variable partitioning:-

When a process enters in main memory, it is allocated exact size that is required by that process. So in this method, partitions can vary in size depending on memory space required by a process entering in main memory. Operating system maintains a table indicating which parts of memory are available and which are occupied. When new process arrives and it needs space, system searches for available memory space in main memory. If it is available then memory is allocated to the process by creating a partition in memory. Like this depending on size of process and available memory, partitions take place in main memory.

For example:-Consider following table with process and memory space.

Process	Memory space
P1	20 M
P2	14 M
P3	18 M
P4	8 M
P5	10 M

Process of memory allocation:-



Total memory size is 64 M .From this 8 M partition is occupied by operating system and remaining can be partitioned as per the size of processes.

Fig a: - the operating system is loaded in the memory. The entire remaining space is free.

Fig b: - process p 1 is loaded with 20 M memory size as space is available in memory. So loading process p1 create one partition in memory.

Fig c: - process p 2 is loaded with 14 M memory size as space is available in memory. So loading process p 2 creates one more partition in memory with 14 M size.

Fig d: - process p 3 is loaded with 18 M memory size as space is available in memory. So loading process p 3 creates one more partition in memory.

Fig e & f: - consider Process P 2 is currently blocked. After some time process P 4 with high priority is ready for execution with memory of 8M.the existing free space is less than the required space. With priority scheduling, suppose P 2 is having less priority over P4 then system performs swapping of process P2 and process P 4.in this case, space occupied by process P2 is released i.e. 14 M and P 4 occupies 8 M in that free space as shown in fig f.

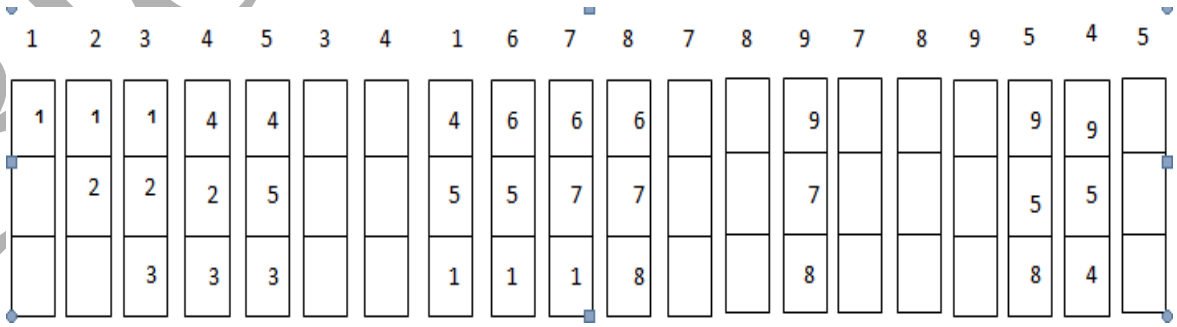
Fig g: - Process P1 completes its job and then it releases its occupied space of 20 M.

Fig h:-Process P2 can be loaded again in the memory in the free partition released by process P 1.but P 2 requires only 20 M, so the free space of 20 M is divided into two partitions of 14 M occupied by P2 and 6 M free space.

6. Attempt any four of the following: Marks 16

a) Consider the following page reference string-1,2,3,4,5,6,7,8,7,8,9,7,8,9,5,4,5. How many page faults occur for FIFO replacement algorithm assuming 3 frames?
(correct frame sequence – 3 Marks, number of Page faults – 1 Mark)

Ans:



(* indicates page hit)

Page fault 12

Page hit 08

b) State and describe different memory management technique.

(State two techniques – 1 Mark, Description of Fixed partitioning – 1 ½ Marks, Description of Variable partitioning – 1 ½ Marks)

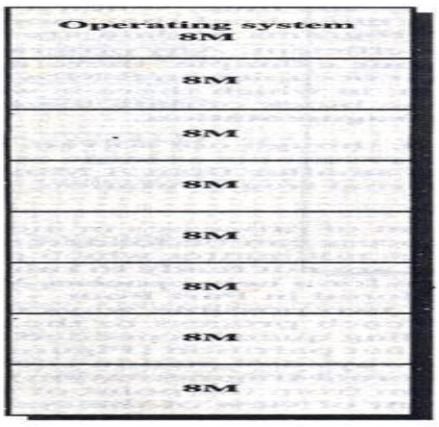
Ans: Memory management techniques:

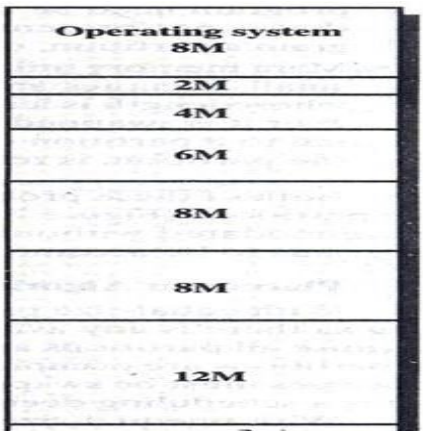
1) Fixed Partitioning
2) Variable partitioning

1) Fixed Partitioning

Main memory is divided into multiple partitions of fixed size at the time of system generation. A process may be loaded into a partition of equal size or greater size. Partitions can be of equal size or unequal size.

Main memory is divided into equal size partitions. Any process with less or equal size can be loaded in any available partition. If all partitions are full and no process is in ready or running state then the operating system can swap a process out of any partition and load in any other process. Main memory can be also divided into multiple partitions of unequal size. Each process can be loaded into the smallest partition within which the process will fit.





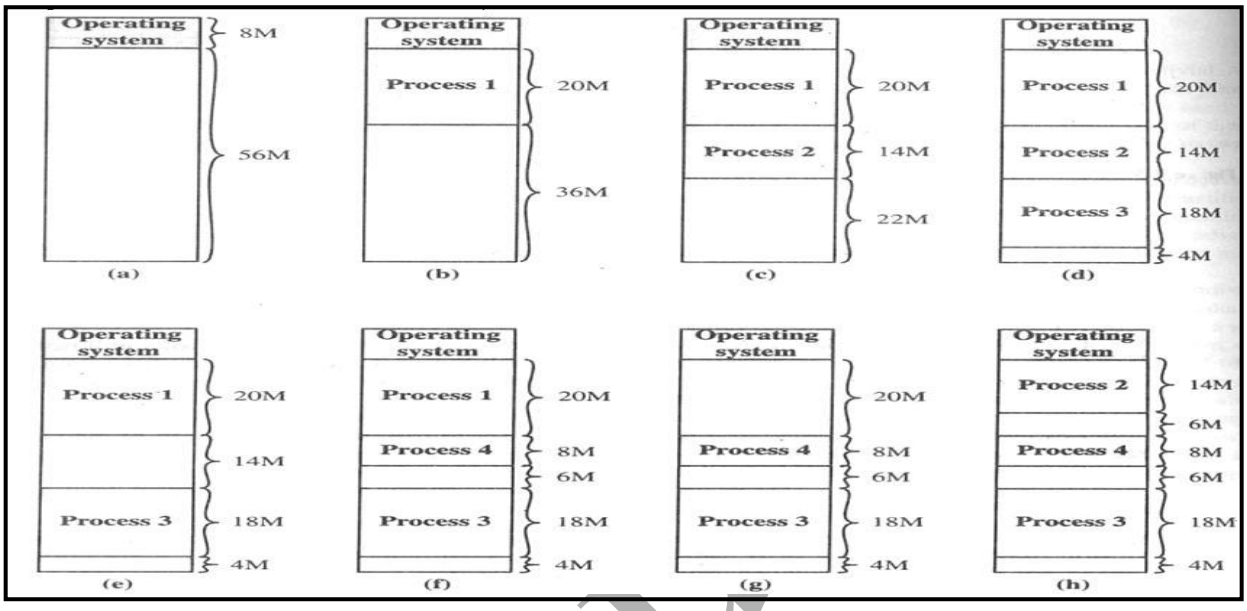
2) Variable partitioning:-

When a process enters in main memory, it is allocated exact size that is required by that process. So in this method, partitions can vary in size depending on memory space required by a process entering in main memory. Operating system maintains a table indicating which parts of memory are available and which are occupied. When new process arrives and it needs space, system searches for available memory space in main memory. If it is available then memory is allocated to the process by creating a partition in memory. Like this depending on size of process and available memory, partitions take place in main memory.

For example:-Consider following table with process and memory space.

Process	Memory space
P1	20 M
P2	14 M
P3	18 M
P4	8 M
P5	10 M

Process of memory allocation:-



c) Compare LINUX and UNIX.

(Any four points – 1 Mark each (Any other valid point shall be considered))

Ans:

	Linux	Unix
What is it?	Linux is an example of Open Source software development and Free Operating System (OS).	Unix is an operating system that is very popular in universities, companies, big enterprises etc.
Cost	Linux can be freely distributed, downloaded freely, distributed through magazines, Books etc. There are priced versions for Linux also, but they are normally cheaper than Windows.	Different flavors of Unix have different cost structures according to vendors
User	Everyone. From home users to developers and computer enthusiasts alike.	Unix operating systems were developed mainly for mainframes, servers and workstations except OSX, Which is designed for everyone. The Unix environment and the client-server program model were essential elements in the development of the Internet

Manufacturer	Linux kernel is developed by the community. Linus Torvalds oversees things.	Three bigest distributions are Solaris (Oracle), AIX (IBM) & HP-UX Hewlett Packard. And Apple Makes OSX, an unix based os..
Usage	Linux can be installed on a wide variety of computer hardware, ranging from mobile phones, tablet computers and video game consoles, to mainframes and supercomputers.	The UNIX operating system is used in internet servers, workstations & PCs. Backbone of the majority of finance infastructure and many 24x365 high availability solutions.
Development and Distribution	Linux is developed by Open Source development i.e. through sharing and collaboration of code and features through forums etc and it is distributed by various vendors.	Unix systems are divided into various other flavors, mostly developed by AT&T as well as various commercial vendors and non-profit organizations.
GUI	Linux typically provides two GUIs, KDE and Gnome. But there are millions of alternatives such as LXDE, Xfce, Unity, Mate, twm, ect.	Initially Unix was a command based OS, but later a GUI was created called Common Desktop Environment. Most distributions now ship with Gnome.
File system support	Ext2, Ext3, Ext4, Jfs, ReiserFS, Xfs, Btrfs, FAT, FAT32, NTFS	jfs, gpfs, hfs, hfs+, ufs, xfs, zfs format
Text mode interface	BASH (Bourne Again SHell) is the Linux default shell. It can support multiple command interpreters.	Originally the Bourne Shell. Now it's compatible with many others including BASH, Korn & C.
Price	Free but support is available for a price.	Some free for development use (Solaris) but support is available for a price.
Security	Linux has had about 60-100 viruses listed till date. None of them actively spreading nowadays.	A rough estimate of UNIX viruses is between 85 -120 viruses reported till date.

Threat detection and solution	In case of Linux, threat detection and solution is very fast, as Linux is mainly community driven and whenever any Linux user posts any kind of threat, several developers start working on it from different parts of the world	Because of the proprietary nature of the original Unix, users have to wait for a while, to get the proper bug fixing patch. But these are not as common.
Processors	Dozens of different kinds.	x86/x64, Sparc, Power, Itanium, PA-RISC, PowerPC and many others.
Examples	Ubuntu, Fedora, Red Hat, Debian, Archlinux, Android etc.	OS X, Solaris, All Linux
Architectures	Originally developed for Intel's x86 hardware, ports available for over two dozen CPU types including ARM	is available on PA-RISC and Itanium machines. Solaris also available for x86/x64 based systems. OSX is PowerPC(10.0-10.5)/x86(10.4)/x64(10.5-10.8)
Inception	Inspired by MINIX (a Unix-like system) and eventually after adding many features of GUI, Drivers etc, Linus Torvalds developed the framework of the OS that became LINUX in 1992. The LINUX kernel was released on 17th September, 1991	In 1969, it was developed by a group of AT&T employees at Bell Labs and Dennis Ritchie. It was written in "C" language and was designed to be a portable, multi-tasking and multi-user system in a time-sharing configuration

d) Describe I/O Burst and CPU burst cycle. Give one example.

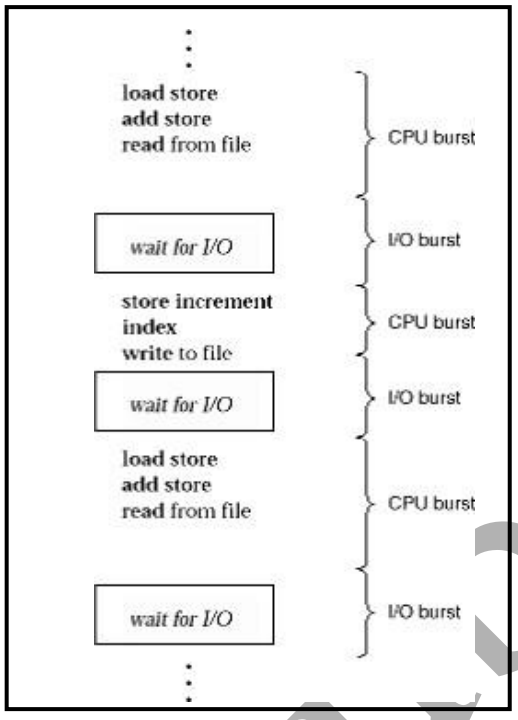
(Explanation - 2 Marks, Example with Diagram -2 Marks)

Ans: Processes require alternate use of processor and I/O in a repetitive fashion. Each cycle consist of a CPU burst followed by an I/O burst .A process terminates on a CPU burst.

CPU-bound processes have longer CPU bursts than I/O-bound processes.

I/O bound process: The process which spends more time in I/O operation than computation (time spends with CPU) is I/O bound process.

CPU bound process: The process which spends more time in computations or with CPU and very rarely with the I/O devices is called as CPU bound process.



e) Describe Shortest Remaining Time (SRTN) algorithm with the help of example.

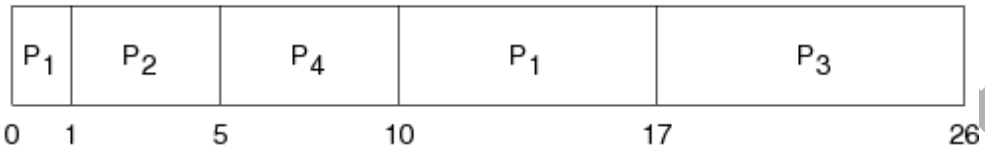
(Description – 2 Marks, Example – 2 Marks)

Ans: Associate with each process the length of its next CPU burst schedule the process with the shortest time preemptive: preempt if a new process arrives with a CPU burst of less length thanthe remaining time of the currently executing process; known as the Shortest Remaining Time First (SRTF) algorithm SJF is provably optimal; it yields a minimum average waiting time for anyset of processes however, we cannot always predict the future (i.e., we do not know the next burst length)we can only estimate its length.

Example:

Process	Burst Time	Arrival
1	8	0
2	4	1
3	9	2
4	5	3

Gantt chart:



average waiting time: (9+0+15+2)/4 = 6.5

average turnaround time: (17+4+24+7)/4 = 13