MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2005 Certified)

SUMMER-16 EXAMINATION
Model Answer

Subject Code: 17512 **Subject Name: Operating System**

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 Equivalentconcept.

Marks

1. a) Attempt any 3 of the following: **4 × 3=12**

1) List any four functions of operating system.
(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 handles 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

2) Describe real time operating system in brief.
(Description of real time operating system - 4 marks; [Types optional])

Ans:

Real time systems are used in environment where a large number of events, mostly external to the computer system, must be accepted and processes in a short time or within certain deadlines. Such applications include real-time simulations, flight control, industrial control, military applications etc.
A primary objective of real-time systems is to provide quick event response time and thus meet the scheduling deadlines. User convenience and resource utilization are of secondary concern to real-time system designers.
In Real time systems, processor is allocated to the highest priority process among those that are ready to execute. Higher priority processes preempt execution of the lower priority processes. This form is called as **'priority –based preemptive scheduling'**.

The primary functions of the real time operating system are to:

1. Manage the processor and other system resources to meet the requirements of an application.
2. Synchronize with and respond to the system events.
3. Move the data efficiently among processes and to perform coordination among these processes.

Types of real time system:

1. Hard real time:-
Hard real time means strict about adherence to each task deadline. When an event occurs, it should be serviced within the predictable time at all times in a given hard real time system.

Example: -video transmission, each picture frame and audio must be transferred at fixed rate.

2. Soft real time:-
Soft real time means that only the precedence and sequence for the task operations are defined, interrupt latencies and context switching latencies are small. There can be few deviations between expected latencies of the tasks and observed time constraints and a few deadline misses are accepted.

Example: - Mobile phone, digital cameras and orchestra playing robots.

3) What is process management? State four functions to be performed by OS for process management.
(Description of Process Management - 2 marks; any four functions - ½ mark each)

Ans:

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.

A processes software that performs some action and can be controlled by a user, by other applications or by the operating system. A process needs various system resources including CPU time, memory, files and I/O devices to complete the job execution. These resources can be given to the process when it is created or allocated to it while it is running.

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

1. Creation and deletion of user and system processes.
2. Suspension and resumption of processes.
3. A mechanism for process synchronization.
4. A mechanism for process communication.
5. A mechanism for deadlock handling.

4) What is file? List and explain attributes of files.
(Definition of file - 1 mark; list and explain any 6 attributes - ½ mark each)

Ans:

Definition of file:

A file is named collection of related information that is recorded on secondary storage. It is a sequence of bits, bytes, lines or records representing data. Data can be stored on secondary storage only when it is place inside the file. File represents programs and data. Data files may be numeric, alphabetic, alphanumeric or binary. Files are mapped by the operating system onto physical devices.

Attributes of file

- **Name**. It is a string of characters which is in human readable form.

- **Identifier**. This unique tag, usually a number, identifies the file within the file system; it is the non-human-readable name for the file.
- **Type**. This is the information used by the system to support different types of the files.
- **Location**. This information is a pointer to a device and to the location of the file on that device.
- **Size**. The current size of the file (in bytes, words, or blocks) and possibly the maximum allowed size are included in this attribute.
- **Protection**. Access-control information determines who can do reading, writing, executing, and so on.
- **Time, date, and user identification**. This information may be kept for creation, last modification, and last use.

Attribute	Meaning
Protection	Who can access the file and in what way
Password	Password needed to access the file
Creator	ID of the person who created the file
Owner	Current owner
Read-only flag	0 for read/write; 1 for read only
Hidden flag	0 for normal; 1 for do not display in listings
System flag	0 for normal files; 1 for system file
Archive flag	0 for has been backed up; 1 for needs to be backed up
ASCII/binary flag	0 for ASCII file; 1 for binary file
Random access flag	0 for sequential access only; 1 for random access
Temporary flag	0 for normal; 1 for delete file on process exit
Lock flags	0 for unlocked; nonzero for locked
Record length	Number of bytes in a record
Key position	Offset of the key within each record
Key length	Number of bytes in the key field
Creation time	Date and time the file was created
Time of last access	Date and time the file was last accessed
Time of last change	Date and time the file has last changed
Current size	Number of bytes in the file
Maximum size	Number of bytes the file may grow to

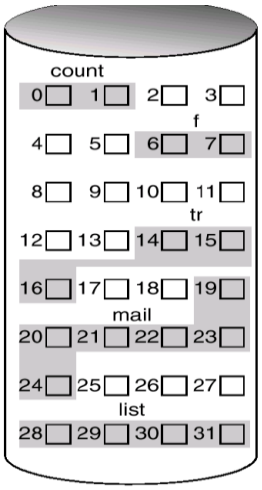
b) Attempt any one of the following. (6 × 1 =6)

1) Describe the contiguous allocation method for file, state any two merits and demerits.
(Description - 4 marks; any two merits - ½ mark each; any two demerits - ½ mark each)

Ans:

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. With this ordering, accessing block b+1 after block b normally requires no head movement. 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.



directory

file	start	length
count	0	2
tr	14	3
mail	19	6
list	28	4
f	6	2

Merits:

1. Easy to implement.
2. Supports both **direct and sequential** access.

Demerits:

1. 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.
2. 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.
3. At the time of file creation, predicting how much space is needed for a file is difficult.
   If we allocate too little space for a file, then we may find that the file cannot be extended.
   If we allocate more space than the current file size for future use, then it may result in memory wastage i.e. internal fragmentation.

2) Describe following operating system structures.

i) Monolithic ii) Microkarnel.

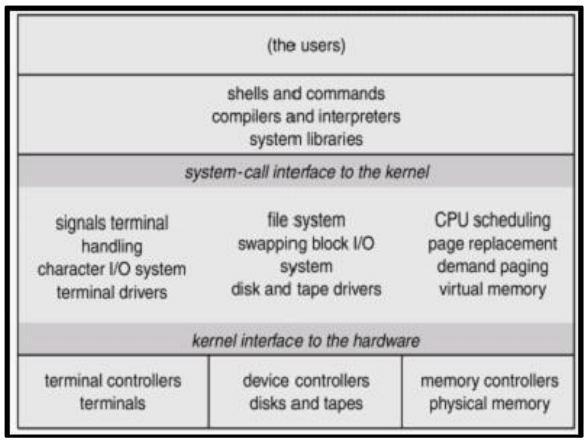
(Description of monolithic - 2 marks; Diagram- 1 mark; description of microkernel - 2 marks; diagram -1 mark)

Ans:

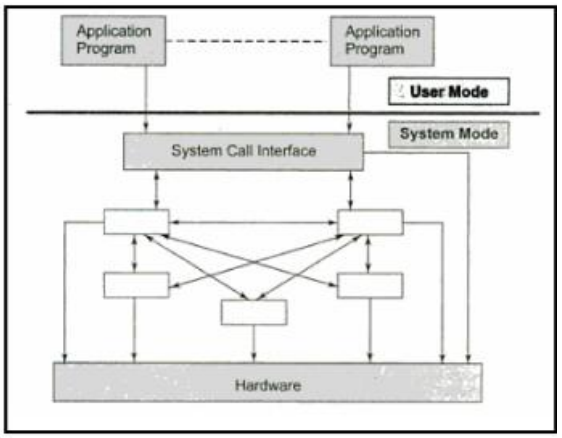
i) Monolithic Systems:

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.

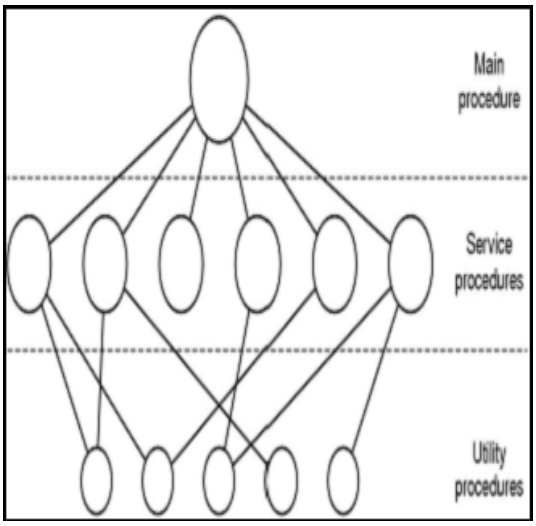
For constructing the actual object program of the operating system when this approach is used, one compiles all the individual procedures, or files containing the procedures, and then binds them all together into a single object file with the linker. In terms of information hiding, there is essentially 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.



OR

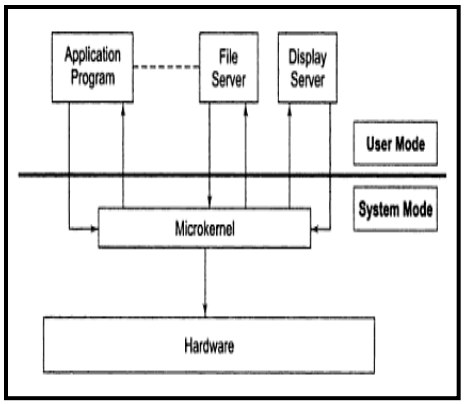


OR

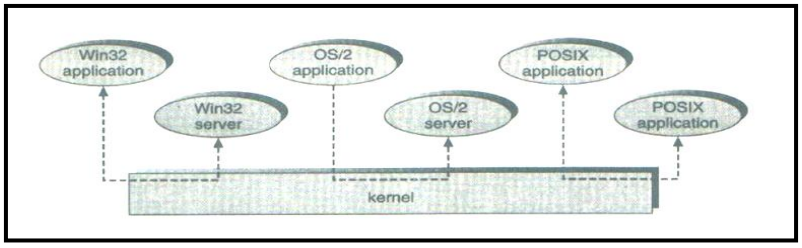


ii) Microkernel

A **microkernel** (also known as μ-kernel) is the near-minimum amount of software that can provide the mechanisms needed to implement an operating system (OS). These mechanisms include low-level address space management, thread management, and inter-process communication (IPC).If the hardware provides multiple rings or CPU modes, the microkernel is the only software executing at the most privileged level (generally referred to as supervisor or kernel mode).Moves as much from the kernel into "user" space. Communication takes place between user modules using message passing.



OR



2. Attempt any 4 of the following: (4 × 4=16)

(a) Compare Unix and Linux operating system w.r.t
1) User Interface 2) Name of provider
3) Processing speed 4) Security
(Comparison - 1 mark to each point; note: any relevant description for the comparison point shall be considered.)

Ans:

CRITERIA	LINUX	UNIX
User interface	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.
Name of Provider	Redhat, Ubantu, Fedora	Osx, Solaris, All LINUX
Processing speed	**Low:** As it is GUI based processing time is more as compare to UNIX.	**High:** As it is command based direct interpretation of commands is done so it takes less time as compare to LINUX.

Security	Linux has had about 60-100 viruses listed till date. None of them actively is spreading nowadays.	A rough estimate of UNIX viruses is between 85 -120 viruses reported till date.

(b) Describe evolution of operating system.
(Explanation of four generations - 1 mark each)

[Note - marks shall be given for generations or types of operating system**]**

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.

OR

Description of batch, Multi programmed, Multitasking, Timesharing, Desktop, Distributed Systems, Clustered System, Real Time system

- **Batch Systems:** Main function of a batch processing system is to automatically keep executing the jobs in a batch.
- **Multiprogramming:** It executes multiple programs simultaneously by a single processor.
- **Multitasking:** Multitasking is a logical extension of multiprogramming. Multiple jobs are executed by the CPU switching between them, but the switches occur so frequently that the users may interact with each program while it is running.
- **Time-Sharing Systems–Interactive Computing:** In time sharing system, the CPU executes multiple jobs by switching among them
- **Desktop Systems:** Personal computers – computer system dedicated to a single user.
- **Distributed system:** Distributed system or distributed data processing is the system in which processors, data and other aspects of a data processing system may be dispersed within on organization.
- **Clustered system**: It is a group of connected computers working together as one unit.
- **Real Time system:**
  A Real Time system is used when there are rigid time requirement on the operation of a processor or the flow of data and thus is often used as a control device in a dedicated application.

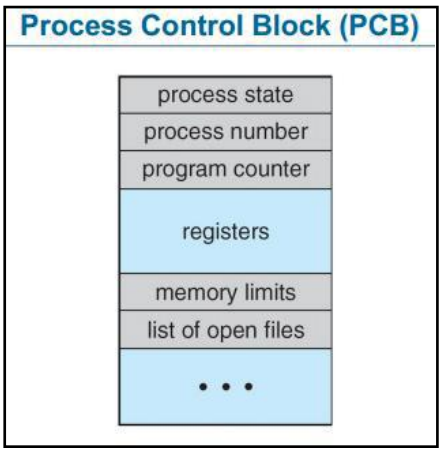
(c) With neat diagram describe use of Process Control Block (PCB)
(Explanation - 2 marks; Diagram - 2 marks)

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 contains 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 the following terms :

i) Preemptive scheduling ii)Nonpreemptive scheduling

(Preemptive Scheduling - 2 marks; Non Preemptive Scheduling - 2 marks)

Ans:

i) Preemptive Scheduling

1. **Even if CPU is allocated to one process, CPU can be preempted to other process if other process is having higher priority or some other fulfilling criteria.**
2. It is suitable for RTS.
3. Only the processes having higher priority are scheduled.
4. It doesn't treat all processes as equal.
5. Circumstances for preemptive
   - process switch from running to ready state
   - process switch from waiting to ready State

ii) Non Preemptive Scheduling

1. **Once the CPU has been allocated to a process the process keeps the CPU until it releases CPU either by terminating or by switching to waiting state.**
2. It is not suitable for RTS.
3. Processes having any priority can get scheduled.
4. It treats all process as equal.
5. Circumstances for Non preemptive
   - Process switches from running to waiting state
   - Process terminates

(e) Describe working of sequential and direct access methods.

(Description of Sequential method - 2 marks; direct access method - 2 marks)

Ans:

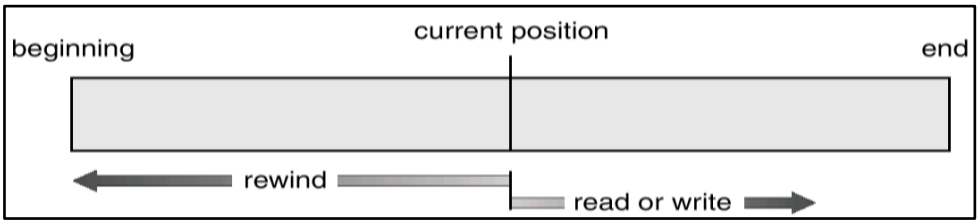
Sequential Access Method:

The simplest access method is sequential access. Information in the file is processed in order, one record after the other.

This mode of access is by far the beginning current position most common; for example, editors and compilers usually access files in this fashion.

Reads and writes make up the bulk of the operations on a file.

- A read operation read next reads the next portion of the file and automatically advances a file pointer, which tracks the I/O location.
- Similarly, the write operation write next appends to the end of the file and advances to the end of the newly written material (the new end of file).



To read a piece of data that is stored at the end of the file, one has to read all of the data that comes before it-you cannot jump directly to the desired data. This is similar to the way cassette tape players work. If one wants to listen to the last song on a cassette tape, he has to either fast-forward over all of the songs that come before it or listen to them. There is no way to jump directly to a specific song.

Direct Access Method:

A file is made up of fixed-length logical records that allow programs to read and write records rapidly in no particular order. Thus, we may read block 14, then read block 53, and then write block 7. There are no restrictions on the order of reading or writing for a direct-access file.

The direct-access method is based on a disk model of a file, since disks allow random access to any file block. Direct-access files are of great use for immediate access to large amounts of information. Databases are often of this type. For the direct-access method, the file operations must be modified to include the block number as a parameter.

The block number provided by the user to the OS is normally a relative block number.

- A relative block number is an index relative to the beginning of the file.
- Thus, the first relative block of the file is 0, the next is 1, and so on, even though the actual absolute disk address of the block may be 14703 for the first block and 3192 for the second.

The use of relative block numbers allows the OS to decide where the file should be placed (called the allocation problem) and helps to prevent the user from accessing portions of the file system that may not be part of her file.

When you work with a direct access file (which is also known as a random access file), you can jump directly to any piece of data in the file without reading the data that comes before it. This is similar to the way a CD player or an MP3 player works. You can jump directly to any song that you want to listen to. Sequential access files are easy to work with, and you can use them to gain an understanding of basic file operations.

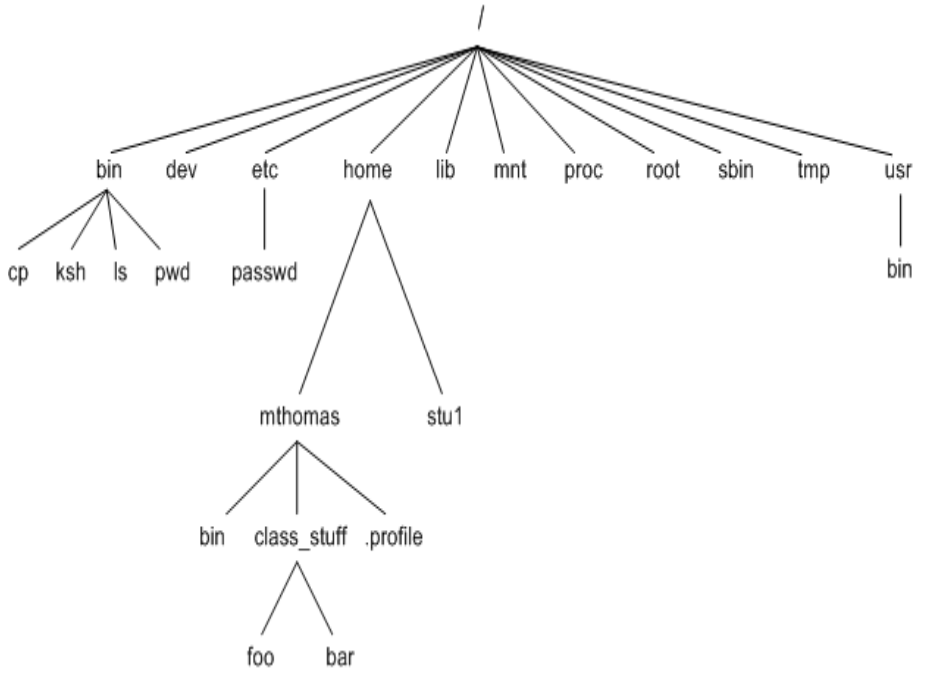
(f) Explain in brief the unix file system.
(Description of unix file system - 4 marks)

Ans:

The Unix file system is a methodology for logically organizing and storing large quantities of data such that the system is easy to manage. A **file** can be informally defined as a collection of (typically related) data, which can be logically viewed as a stream of bytes (i.e. characters). A file is the smallest unit of storage in the Unix file system.

The Unix file system has a hierarchical (or tree-like) structure with its highest level directory called root (denoted by /, pronounced *slash*). Immediately below the root level directory are several subdirectories, most of which contain system files. Below this can exist system files, application files, and/or user data files. Similar to the concept of the process parent-child relationship, all files on a UNIX system are related to one another. That is, files also have a parent-child existence. Thus, all files (except one) share a common parental link, the top-most file (i.e. /) being the exception.

Below is a diagram (slice) of a "typical" Unix file system. The top-most directory is / (slash), with the directories directly beneath being system directories.



The following system files (i.e. directories) are present in most Unix file systems:

- ***bin*** **-** short for binaries, this is the directory where many commonly used executable commands reside
- ***dev*** **-** contains device specific files
- ***etc*** **-** contains system configuration files
- ***home*** **-** contains user directories and files
- ***lib*** **-** contains all library files
- ***mnt*** **-** contains device files related to mounted devices
- ***proc*** **-** contains files related to system processes
- ***root*** **-** the root users' home directory (note this is different than /)
- ***sbin*** **-** system binary files reside here. If there is no sbin directory on your system, these files most likely reside in etc
- ***tmp*** **-** storage for temporary files which are periodically removed from the file system
- ***usr*** **-** also contains executable commands

3. Attempt any 4 of the following: 4 × 4 =16

(a) List any four operating system services and describe in one/two sentences.
(List any four services - 2 marks; Description of each - ½ marks each)

Ans:

User interface

(1) Program execution
(2) I/O operations
(3) File-system manipulation
(4) Communications
(5) Error detection
(6) Accounting
(7) Resource allocation
(8) Protection and security

Description of services of operating system

1. **User interface:** Almost all operating systems have a user interface (UI).Almost all operating systems have a user interface (UI). It varies between Command-Line (CLI), Graphics User Interface (GUI).
2. **Program execution:** The system must be able to load a program into memory and to run that program, end execution, either normally or abnormally (indicating error)

3. **I/O operations: - S**ince user programs cannot execute I/O operations directly, the operating system must provide some means to perform I/O. Each program requires an input and produces output.
4. **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.
5. **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, and reassembled 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.
6. **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.
7. **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 ID for the user, all such in formations we can manage using accounting service provided by many multiuser systems.
8. **Resource allocation:** When there are multiple users or multiple jobs running at the same time. Resources must be allocated to each of them. Many different types of resources are managed by the operating system.
9. **Protection and security:** The owners of information stored in multiuser or networked computer system may want to control use of the information. When several separate processes execute concurrently, it should not be possible for one process to interfere with the others or with the operating system itself, and protection involves ensuring that all access to system resources is controlled. Security of the system from outsiders is important.

(b) Describe concept of virtual memory with suitable example.

(Description of Virtual memory - 2 marks; Example - 2 marks)

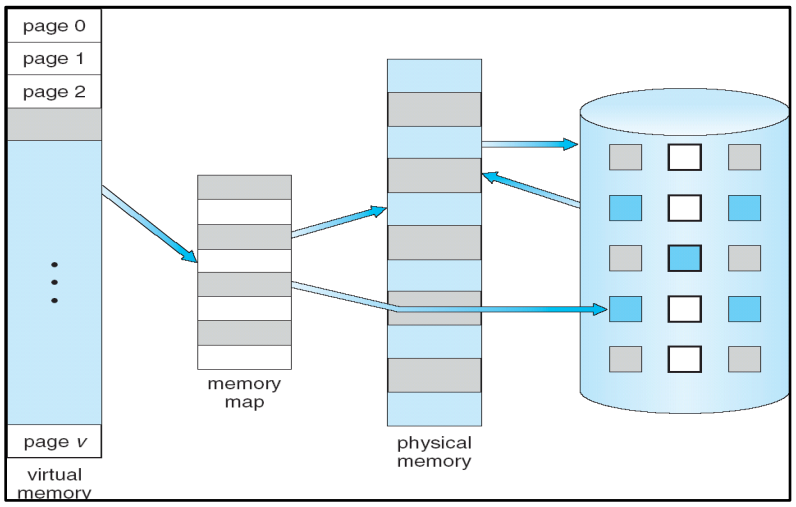
Ans:

Virtual memory is the separation of user logical memory from physical memory.

This separation allows an extremely large virtual memory to be provided for programmers when only a smaller physical memory is available.

Virtual memory makes the task of programming much easier, because the programmer no longer needs to worry about the amount of physical memory available, or about what code can be placed in overlays, but can concentrate instead on the problem to be programmed.

It is the process of increasing the apparent size of a computer's RAM by using a section of the hard disk storage as an extension of RAM.
As computers have RAM of capacity 64 or 128 MB to be used by the CPU resources which is not sufficient to run all applications that are used by most users in their expected way and all at once.

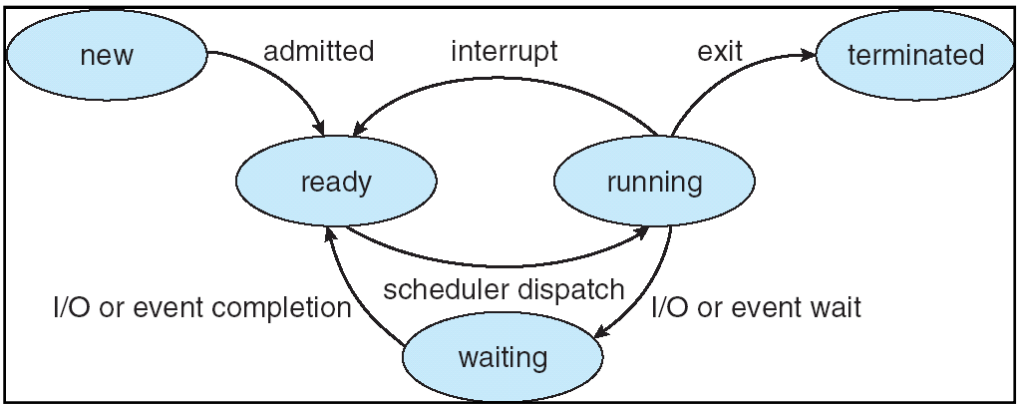


Example:

For example, an e-mail program, a web browser and a word processor is loaded into RAM simultaneously; the 64 MB space is not enough to store all these programs.
Without a virtual memory, a message "You cannot load any more applications. Please close an application to load a new one." would be displayed. By using a virtual memory, a computer can look for empty areas of RAM which is not being used currently and copies them on to the hard disk device. Thus RAM is freed to load new applications.
Actually it is done automatically, the user do not even know that it is happening, and the user feels like RAM has unlimited space even though the RAM capacity is 32 MB. It is a process of increasing computer's RAM by using a section of the hard disk storage as an extension of RAM.

(c) Draw the process state diagram and describe each state in one/two sentences.
(Diagram - 2 marks; Description of states - 2 marks)

Ans:



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 that is currently waiting for 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.

(d) State and explain criteria in CPU scheduling.
(List any four criteria - 2 marks; Explanation - 2 marks)

Ans:

1. **CPU utilization**
2. **Throughput**
3. **Turnaround time**
4. **Waiting time**
5. **Response time**

Explanation of criteria for CPU scheduling -

1. **CPU utilization:** Keep the CPU as busy as possible.
2. **Throughput:** Number of processes that complete their execution per time unit.
3. **Turnaround time:** Amount of time to execute a particular process. The interval from the time of submission of a process to the time of completion is the turnaround time.
4. **Waiting time:** Amount of time a process has been waiting in the ready queue
5. **Response time:** Amount of time it takes from when a request was submitted until the first response is produced, **not** output (for time-sharing environment)

(e) What is FCFS algorithm? Describe with example.
(Explanation - 2 marks; Example - 2 marks)

Ans:

First-Come - First-Served (FCFS) Scheduling

FCFS scheduling is non preemptive algorithm. Once the CPU is allocated to a process, it keeps the CPU until it releases the CPU, either by terminating or by requesting I/O.

In this algorithm, a process, that a request the CPU first, is allocated the CPU first. FCFS scheduling is implemented with a FIFO queue. When a process enters the ready queue, its PCB is linked to the tail of the queue. When the CPU is available, it is allocated to the process at the head of the queue. Once the CPU is allocated to a process, that process is removed from the queue. The process releases the CPU by its own.

Example 01:

Process	Burst Time
P_1	24
P_2	3
P_3	3

Suppose that the processes arrive in the order: P_1, P2, P_3

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

4. a) Attempt any 3 of the following: **Marks 4×3=12**

(a) What is inter process communication? Describe any one technique of it.

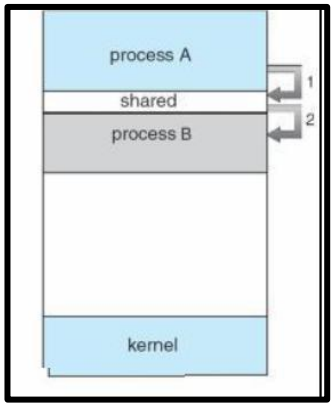
(Define inter process communication -1 mark; diagram of any one model - 1 mark; explanation - 2 marks)

Ans:

Inter-process communication: Cooperating processes require an Inter- process communication (IPC) mechanism that will allow them to exchange data and information.

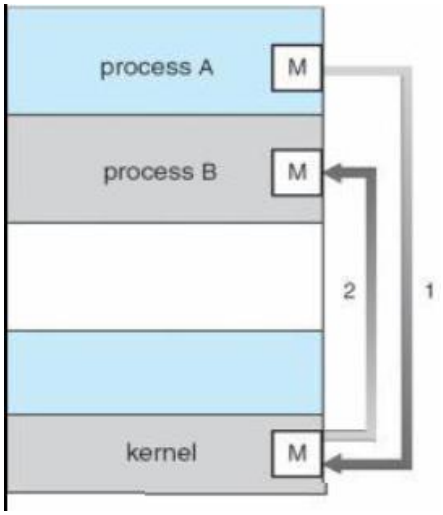
There are two models of IPC

1. **Shared memory**



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 the operating 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.

2. Message Passing



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 s sending and receiving messages through the kernel. The processes that want to communicate with each other must have a communication link between them. Between each pair of processes exactly one communication link.

(b) Differentiate between long term scheduler and short term scheduler on basis of

i) Selection of job

ii) Frequency of execution

iii) Speed

iv) Accessing which part of system

(Difference with respect to four criteria's - 1 mark each)

Ans*:*

Criteria	long term scheduler	short term scheduler
Selection of job	It selects processes from pool and loads them into memory for execution	It selects processes from ready queue which are ready to execute
Frequency of execution	executes much less frequently when ready queue has space to accommodate new process.	frequently select a new process for the CPU, at least once every 100 milliseconds
Speed	Speed is less short term scheduler	Speed is fast
Accessing which part of system	Access job pool and ready queue	Access ready queue and CPU.

(c) What is system call? List types of system call with one example of system call.
(Description of system call - 2 marks; List of any four system calls with one example - ½ marks each)

Ans:

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

Types of system calls:

- **a.** Process or Job control
- **b.** File Management
- **c.** Device Management
- **d.** Information Maintenance

a. 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

b. **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
c. **System calls Related to Device Management:** Request a device, Release a device Read, Write, Reposition Get device attribute, set device attribute
d. **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.

(d) What are the activities involved in secondary storage management?
(Description of three Activities - 4 marks)

Ans:

The three major activities of an operating system in regard to secondary storage management are:

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

b) Attempt any one. (6 × 1=6)

a) Describe how semaphores are useful for solving problems of interprocess communication.
(Description of use of semaphore - 4 marks)

Ans:

Semaphore is a synchronization tool. A semaphore S is an integer variable which is initialized and accessed by only two standard operations: wait () and signal ().All the modifications to the integer value of semaphore in wait () and signal () operations can be done only by one process at a time.
Example for Semaphore used to solve synchronization problem:-
Consider two concurrently running processes P_1 and P_2.P_1 contains statement S_1 and P_2 contains statement S_2.When we want to execute statement S_2 only after execution of statement S_1, then we can implement it by sharing a common semaphore **synch** between two processes. Semaphore **synch** is initialized to 0.to execute the sequence modify code for process P_1 and P_2.

Process P1 contains:

S_1;
signal (synch);

Process P2 contains:-

wait (synch);
S_2;

As synch is initialized to 0, Process P_2 will wait and process P_1 will execute. Once process P_1 completes execution of statement S_1, it performs signal () operation that increments synch value. Then wait () operation checks the incremented value and starts execution of statement S_2 from Process P_2.

b) Write in short on basic memory management.

(Explanation of basic memory management - 4 marks)

[**Note: any relevant description including fixed partitioning, variable partitioning, free space management shall be considered**]

Ans:

Memorymanagementisthefunctionalityofanoperatingsystemwhichhandlesormanagesprimarymemory.Memorymanagementkeepstrackofeachandeverymemorylocationeitheritisallocatedtosomeprocessoritisfree.Itcheckshowmuchmemoryistobeallocatedtoprocesses.Itdecideswhichprocesswillgetmemoryatwhattime.Ittrackswheneversomememorygetsfreedorunallocatedandcorrespondinglyitupdatesthestatus.

Two important terms considered while loading programs into the partition are internal fragmentation and external fragmentation.

Internal fragmentation is memory which is internal to a region but not being used i.e. a memory space wasted in a partition. If block of data is smaller than partition size then the internal fragmentation takes place.

External fragmentation occurs when memory partition is available and unused but too small for any waiting process i.e. process size is greater than the size of available partition.

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.

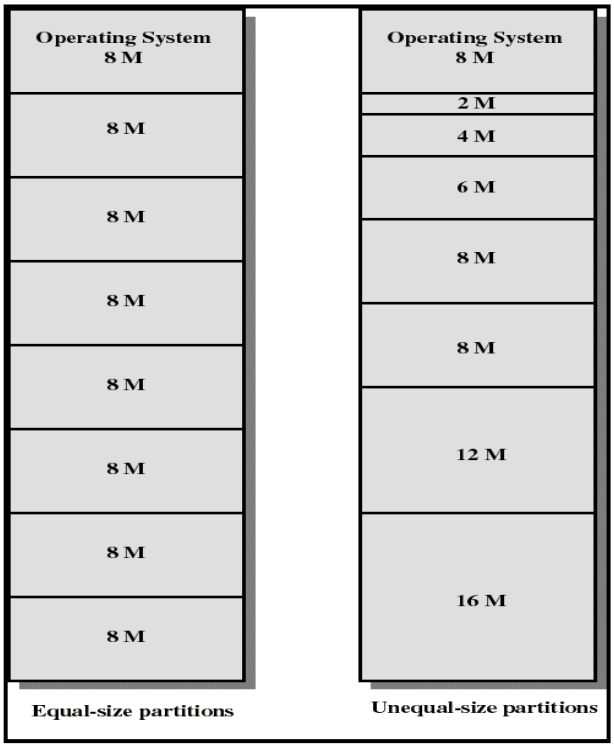
qual size partitioning:-

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.

Disadvantage: -Main memory Utilization is extremely inefficient.

Unequal size partitioning:-

Main memory is divided into multiple partitions of unequal size. Each process can be loaded into the smallest partition within which the process will fit.



5. Attempt any 2 of the following. **8×2=16**

a) Describe following terms

1) Scheduling queues 2) Schedular

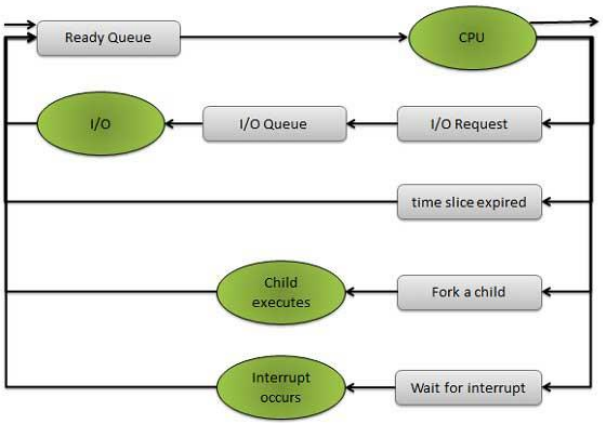
3) Thread 4) Multithreading .

(For each term - 2 marks)

Ans:

1. Scheduling Queues:

Scheduling queues refers to **queues** of processes or devices. When the process enters into the system, then this process is put into a job **queue**. This **queue** consists of all processes in the system. The operating system also maintains other **queues** such as device **queue**



The above figure shows the queuing diagram of process scheduling.

- Queue is represented by rectangular box.
- The circles represent the resources that serve the queues.
- The arrows indicate the process flow in the system.

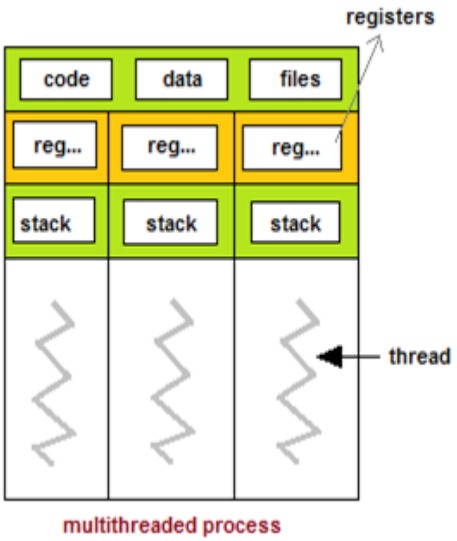
2. Schedular:

The CPU scheduler selects process from the processes in memory that are ready to execute and allocates the CPU to one of themThe long-term scheduler selects process from job pool and loads it in ready queue. A short term scheduler selects a process from ready queue and assigns CPU to it for execution.

3. A **thread** is a basic unit of CPU utilization, consisting of a program counter, a stack, and a set of registers, (and a thread ID). Traditional (heavy weight) processes have a single thread of control - There is one program counter, and one sequence of instructions that can be carried out at any given time.

4. Multithreading:

Multithreading is the ability of a central processing unit (**CPU**) or a single core in a multi-core **processor** to execute multiple processes or threads concurrently, appropriately supported by the **operating system.**



b) Solve the following problem using SJF and Round Robin (RR) scheduling algorithm. Find average waiting time for each algorithm.

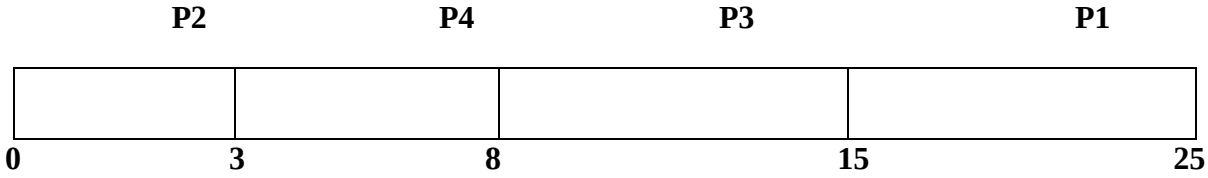
Process	Burst time
P_1	10
P_2	3
P_3	7
P_4	5

(For each scheduling, Gantt chart - 1 mark; waiting time calculation - 2 marks; average Waiting time - 1 mark)
[**Note: For round robin scheduling time quantum is not mentioned. Student can consider any time quantum and calculate average time.**]

Ans:

SJF

Gantt chart

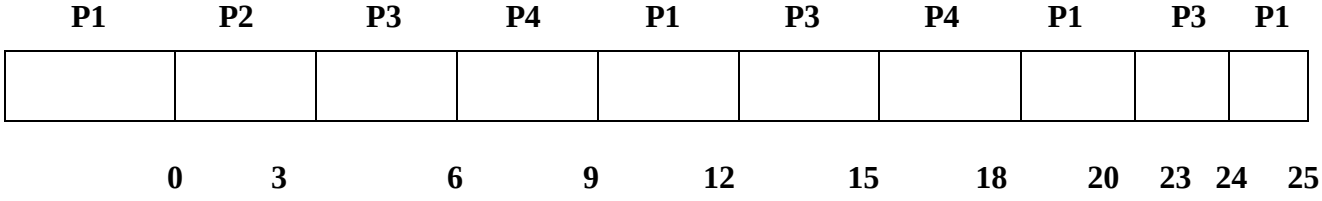


Waiting time
P1 15 mesc
P2 0 mesc
P3 8 msec
P4 3 msec

Average waiting time=waiting time of all processes/number of processes

=waiting time of (P1+P2+P3+P4)/4
=15+0+8+3/4=26/4=6.5 mesc

Round Robin:
[Note: assuming time quantum of 3 msec]**
Gantt chart:



Waiting time
P1 =0+(12-3)+(20-15)+(24-23)=0+9+5+1=15msec
P2 =3msec
P3 =6+(15-9)+(23-18)=6+6+5=17msec
P4 =9+(18-12)=9+6=15 msec
Average waiting time=waiting time of all processes/number of processes

=waiting time of (P1+P2+P3+P4)/4
=15+3+17+15/4=50/4=12.5 msec

c) Explain how priority scheduling algorithm works with suitable explain, also list advantages and disadvantages.

(Description - 3 marks; any relevant example - 3 marks; any one advantage - 1 mark; any one disadvantage - 1 mark)

Ans:

Priority scheduling algorithm:
In priority scheduling algorithm, Number (integer) indicating priority is associated with each process. The CPU is allocated to a process with the highest priority. A priority algorithm will preempt the CPU if the priority of the newly arrived process is higher than the priority of the currently running process.

A major problem with priority scheduling is indefinite blocking or starvation. A solution to the problem of indefinite blockage of the low-priority process is *aging*. Aging is a technique of gradually increasing the priority of processes that wait in the system for a long period of time.

Advantage:
Priority Scheduling-

- Simplicity.
- Reasonable support for priority.
- Suitable for applications with varying time and resource requirements.

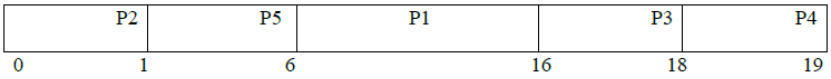
Disadvantages of Priority Scheduling-

- Indefinite blocking or starvation.
- A priority scheduling can leave some low priority processes waiting indefinitely for CPU.

Example:

PROCESS	BURST TIME	PRIORITY
P1	10	3
P2	1	1
P3	2	4
P4	1	5
P5	5	2

Gantt chart:



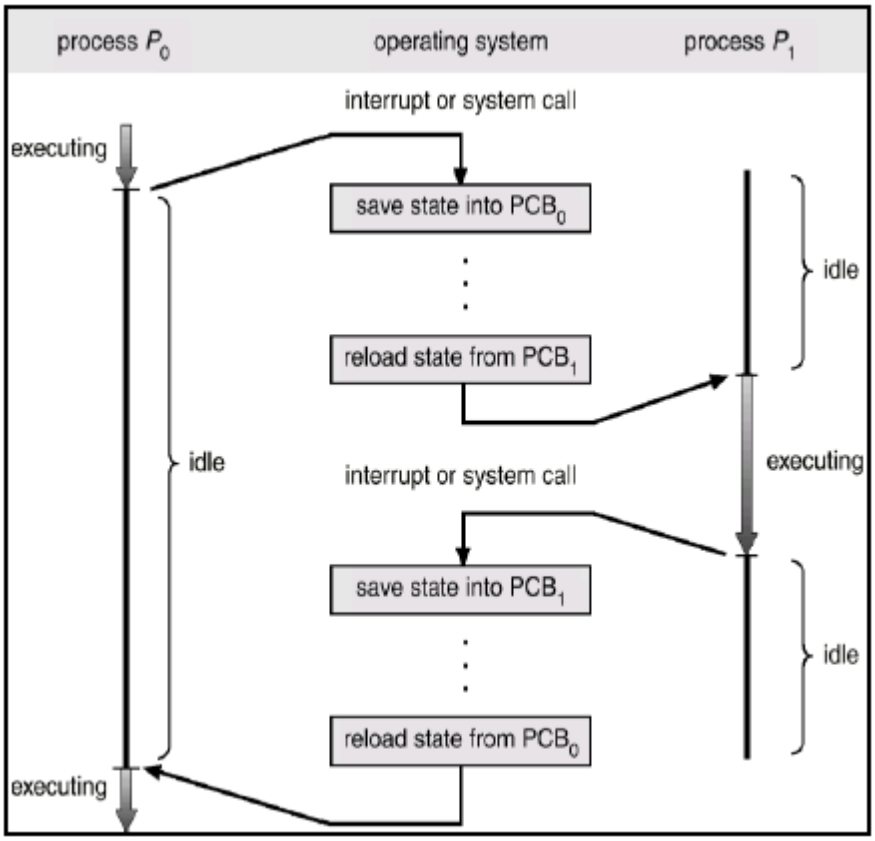
Waiting time for each process: p1 =6 ,p2 =0, p3 =16 ,p4 =18 ,p5 =1
Average waiting time: = (6+0+16+18+1) /5 =41/5 = 8.2 milliseconds

6. Attempt any 4 of the following: **4 × 4 =16**

a) Describe how context switch is executed by operating system.
(Description - 3 marks; Diagram - 1 mark)

Ans:

Switching the CPU to another process requires saving the state of current process and loading the saved state for new process. This process is known as a context switch. The context switch is represented in PCB. Saves context of old process in its PCB and loads context of new process into the memory which is schedule to run next.



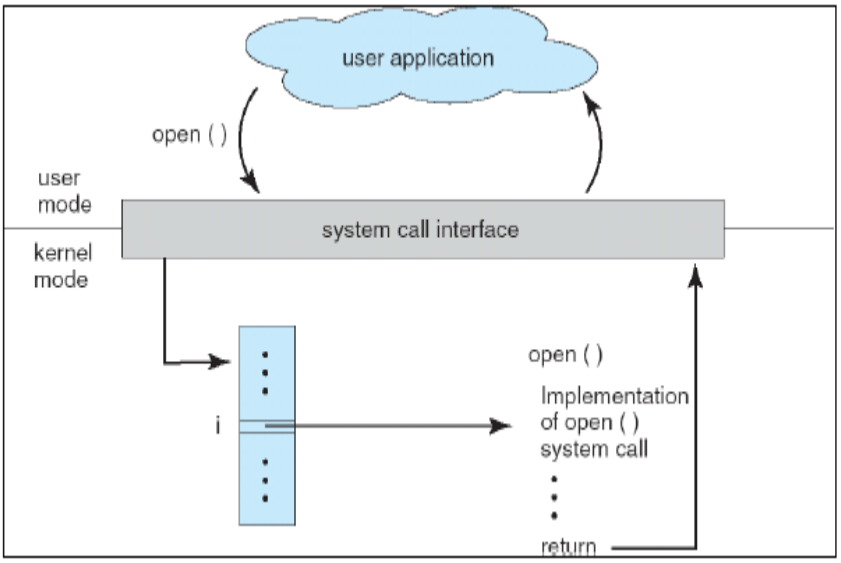
b) Explain how parameter passing is done while implementing system calls.
(Description - 2 marks; diagram - 2 marks)

Ans:

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

- Each system call associated with a particular number.
- System call interface maintains a table indexed according to these numbers.
- The system call interface invokes intended system call in operating system kernel& returns status of the system call and any return values.
- 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.

- Most details of operating system interface hidden from programmers by API. It is managed by run-time support library.



c) What is multiprocessor system? Give two advantages of it.
(Description - 2 marks; any two advantages - 1 mark each)

Ans:

Multiprocessor Systems:
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:
• Less time duration required for the large process.
• Increase throughput.

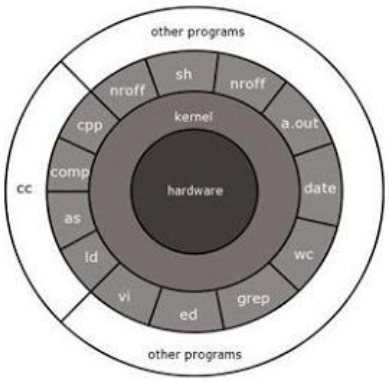
d) Draw and explain structure of unix operating system.
(Explanation - 2 marks; Diagram - 2 marks)

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 the file 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 customize 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)**



e) Describe optimal page replacement algorithm with example.
(Description - 2 marks; any relevant example - 2 marks)

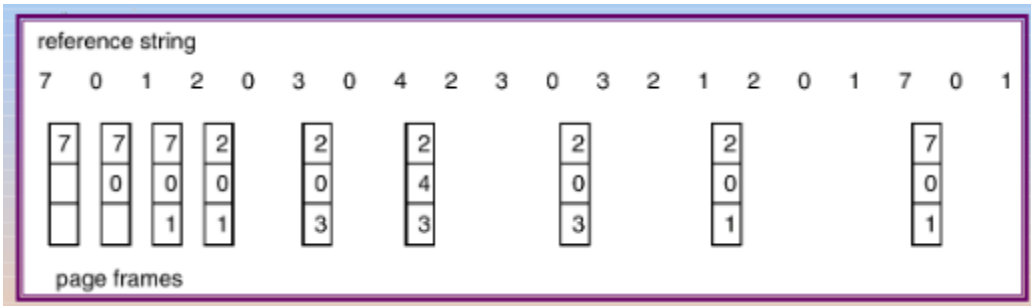
Ans:

Optimal Algorithm

- Replace page that will not be used for longest period of time.
- Has the lowest page-fault rate of all algorithms
- It replaces the page that will not be used for the longest period of time. difficult to implement, because it requires future knowledge used mainly for comparison studies.

Example:-

Reference string:- 7,0,1,2,0,3,0,4,2,3,0,3,2,1,2,0,1,7,0,1
Number of frames: 3



Number of page fault:-09
Number of page hit:11