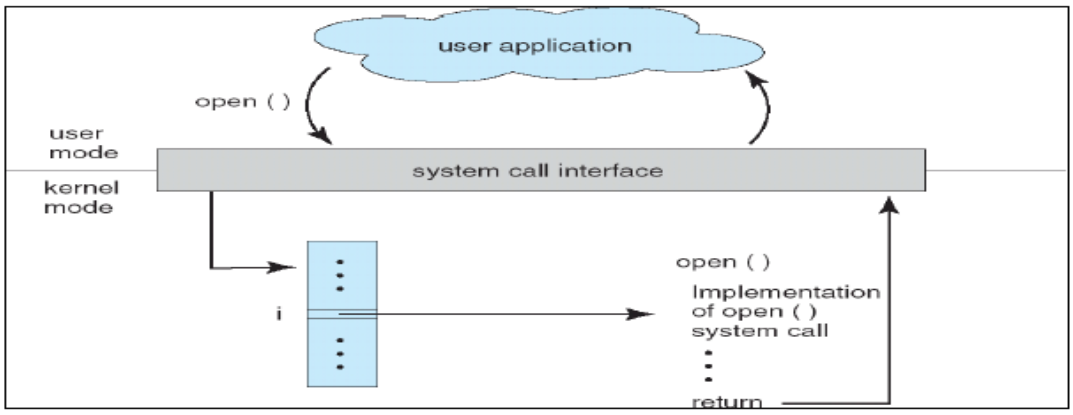
Important Instructions to examiners:

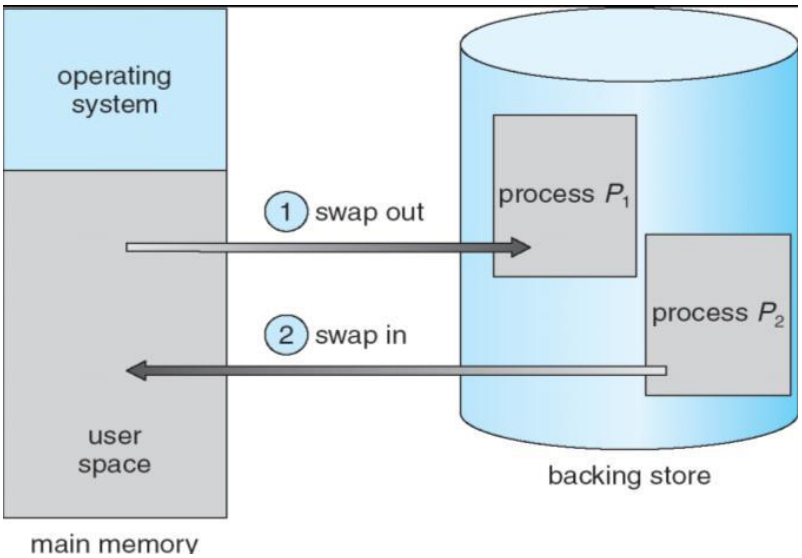
1) The answers should be examined by key words and not as word-to-word as given in the model answer scheme.
2) The model answer and the answer written by candidate may vary but the examiner may try to assess the understanding level of the candidate.
3) The language errors such as grammatical, spelling 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 judgement 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.

Q. No.	Sub Q. N.	Answer	Marking Scheme
1.	**(A)**	**Attempt any THREE :**	**12 Marks**
	(a)	**Define realtime operating system. Explain with the help of example.**	**4 M**
	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'.** 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 compute r 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.	**(Description: 2 marks, any relevant example: 2 marks)**

	(b)	**Describe the purpose of system calls? State two system calls with its functions.**	**4 M**
	Ans:	**System calls** provide an interface between the process and the **operating system**. **System calls** allow user-level processes to request some services from the **operating system** which process itself is not allowed to do. System call provides basic functionality to user to operate operating system. System calls are programming interface to the services provided by the Operating system.  System calls- **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.	**(purpose: 1 marks, any two system calls with their functions: 1½ mark each)**
	(c)	**Define swapping? When it is used?**	**4 M**
	Ans:	**Swapping** Swapping is a simple memory/process management technique used by the operating system (os) to increase the utilization of the processor by moving some blocked process from the main memory to the secondary memory (hard disk); thus forming a queue of temporarily suspended process and the execution continues with the newly arrived process. After performing the swapping process, the operating system has two options in selecting a process for execution. Assume a multiprogramming environment with Round-Robin CPU scheduling algorithm. When a quantum expires memory manager will start to swap out the process that just finished, and swap in another price to the memory space that has been freed. In the meantime, CPU scheduler will allocate a time slice to some other process in memory. When each process finishes its quantum it will be swapped back with another process. Ideally, memory manager can swap process fast enough so that there is always process in memory,	**(Definition/ description: 2 marks, purpose: 2 marks)**

ready to execute, when CPU scheduler wants to reschedule the CPU. The quantum must also be sufficiently large that reasonable amounts of computing are done between swaps.

Swapping can be implemented in various ways. For example, swapping can be priority based.



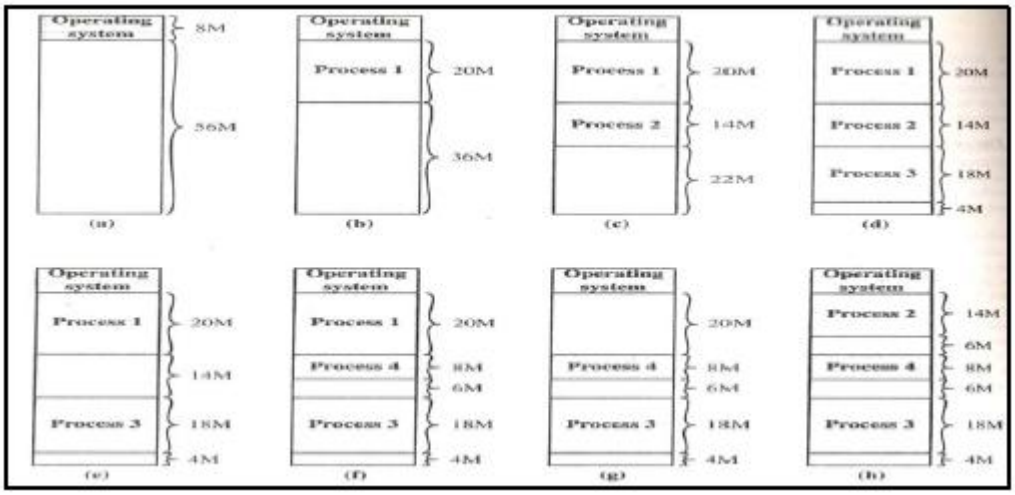
Swapping of two processes using a disk as a backing store.

	(d)	**List and state any four services provided by an operating system.**	**4 M**
	Ans:	**OS services provided to the user:-** **1. User Interface** **2. Program execution** **3. I/O operations** **4. File system manipulation** **5. Communication** **6. Error detection** **1. User Interface**: - All operating systems have a user interface that allows users to Communicate with the system. There are three types of user interfaces are available:- **1.** Command line interface (CLI): - It uses text commands and a method for entering Them. For example working on DOS prompt. **2.** Batch interface: - Commands and directives to control that commands, are entered into A file and the file is executed. For example, Combining set of C programming Statements into a file to perform a specific task and executing that file in TURBO C. **3.** Graphical user interface (GUI): - This interface is a window system with a pointing device to direct I/O, select from menus and make selections and keyboard to enter the Text. For example, Windows system provides icons for selecting an application. Double Clicking on that icon will open that application. Some system provides to or all three of these variations. **2. Program execution**: - The operating system provides an environment where the user	**(List- 1 mark, Explanation of any four services: 1/2 mark each)**

		can. Conveniently run programs. To run a program, the program is loaded into the main Memory and then CPU is assigned to that process for its execution. Operating system Performs this function for the convenience of the user. It also performs other important Tasks like allocation and de-allocation of memory, CPU scheduling etc. It also provides Service to end process execution either normally or abnormally by indicating error. **3. I/O operations**: - When a program is running, it may require input/output resources such As a file or devices such as printer. For specific devices, special functions may be required. Such as recording to a CD drive. For efficiency and protection users usually cannot control I/O devices directly. So the operating system provides a service to do I/O. **4. File system manipulation: -** Programs may need to read and write data from and to the Files and directories. Operating system manages the secondary storage. User gives a Command for reading or writing to a file. Operating system makes it easier for user programs to accomplish their task such as opening a file, saving a file and deleting a file From the storage disk. It also provides services for file permission management to allow or Deny access to files or directories based on file ownership. **5. Communication: -** In the system, one process may need to exchange information with Another process. Such communication may occur between processes that are executing on different computer systems tied together by a computer network. Communication can be implemented via shared memory or through message passing, in which packets of Information are moved between processes by the operating system. **6. Error detection: -**The operating system needs to be constantly aware of possible errors. Errors can occur in: **1.** CPU and memory hardware such as a memory error or power failure **2.** I/O devices such as parity error on tape, a connection failure on a network or lack of paper in the printer **3.** The user program such as an arithmetic overflow, an attempt to access an illegal Memory location or a too-great use of CPU time. For each type of error, the operating system should take the appropriate action to Ensure correct and consistent computing. Debugging facilities can greatly enhance The user's and programmers abilities to use the system efficiently.	
	(B)	**Attempt any ONE:**	**6 Marks**
	(a)	**Explain the concept of variable memory partitioning with example.**	**6 M**
	Ans;	**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 is 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.	**(Concept of variable partitioning: 3 marks, Example with Diagram: 3 marks)**

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. 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.

(b)	**List and explain components of operating system.**	**6 M**

Ans:

List of System Components:
1. Process management
2. Main memory management
3. File management
4. I/O system management
5. Secondary storage management

(List : 2mark, Description of any four : 1 mark each)

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 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:
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 and data transfer rate and access methods. A file system 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

- The creation and deletion of files.
- The creation and deletion of directions.
- The support of primitives for manipulating files and directions.
- The mapping of files onto secondary storage.
- The backup of files on stable storage media.

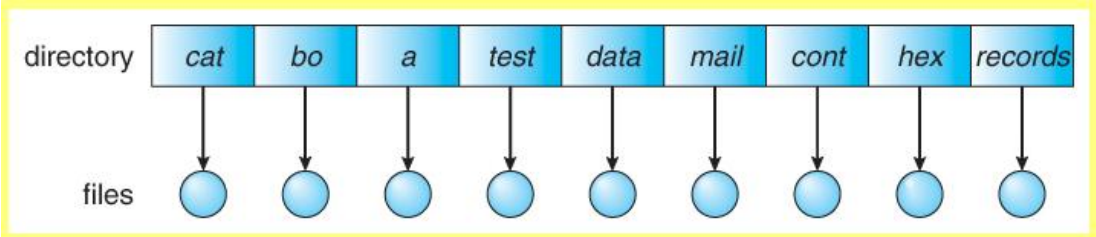
I/O System Management I/O subsystem hides the peculiarities of specific hardware devices from the user. Only the device driver knows the peculiarities of the specific device to which it is assigned.

Secondary-Storage Management:
Systems have several levels of storage, including primary storage, secondary storage and

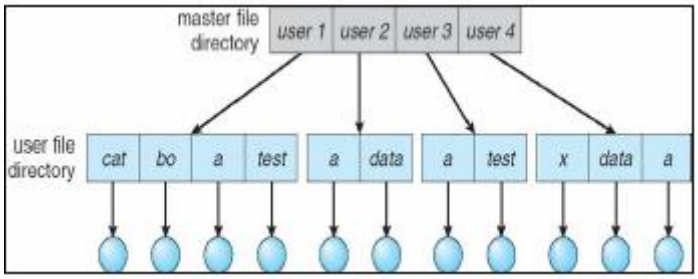
		cache storage. Instructions and data must be placed in primary storage or cache to be referenced by a running program. Because main memory is too small to accommodate all data and programs, and its data are lost when power is lost, the computer system must provide secondary storage to back up main memory. 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: • Managing the free space available on the secondary-storage device • Allocation of storage space when new files have to be written. • Scheduling the requests for memory access.	
2.		**Attempt any FOUR:**	**16**
	(a)	**List and explain major features of unix.**	**4 M**
	Ans:	**MULTI USER CAPABILITY:** In multi user system all users share various computer resources (hard disk, printer, and memory). **MULTI TASKING CAPABILITY:** It is capable of carrying out more than one job at the same time. It allows user to type in a program in its editor while it simultaneously executes some other commands like sort, copy a huge file. The later job is performed in the background while in the foreground uses the editor or takes a directory listing or anything else. This is managed by dividing the CPU time intelligently between all process being carried out .Depending on the priority of the task, the OS approximately allots small time slots (of the order of milliseconds or microseconds) to each foreground and background task. **COMMUNICATION:** The communication may be within the network of a single main computer or between two or more such computer networks. The users can easily exchange mail, data programs through such networks. Distance; possess no barrier to passing information or messages to and fro. **SECURITY:** Unix has 3 inherent provisions for protecting data. i) Assign passwords and login name to user's area. ii) Provide read, write and execute permission to each file. iii) Encrypt files into an unreadable format, and decrypting the file is also possible.	**(Any four features: 1 mark each)**

		PORTABILITY: Unix is a highly portable OS. It can be ported to a variety of hardware platforms. It is the only system which has been ported across various hardware platforms of major vendors such as IBM, DEC, HP and SUN. Most OS are written for one specific machine or platform only. **Machine-independence**: The System hides the machine architecture from the user, making it easier to write applications. **UNIX shell**: UNIX has a simple user interface called the shell that has the power to provide the services that the user wants. **Pipes and Filters**: UNIX has facilities called Pipes and Filters which permit the user to create complex programs from simple programs. **Utilities:** UNIX has over 200 utility programs for various functions. **OR** Built-in networking (TCP/IP) is standard 1. It supports high level language. 2. Easy to use. 3. Security provider. 4. Simple user interface. 5. Make complex program 6. Hierarchy files structure (Directories or files). 7. Multi user and multiprocessor system. 8. Hide machine architecture from the user	
	(b)	**Describe evolution of operating system.** **{**Note - marks shall be given for generations or types of operating system**}**	**4 M**
	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.	**(Explanation of four generations: 1 mark each)**

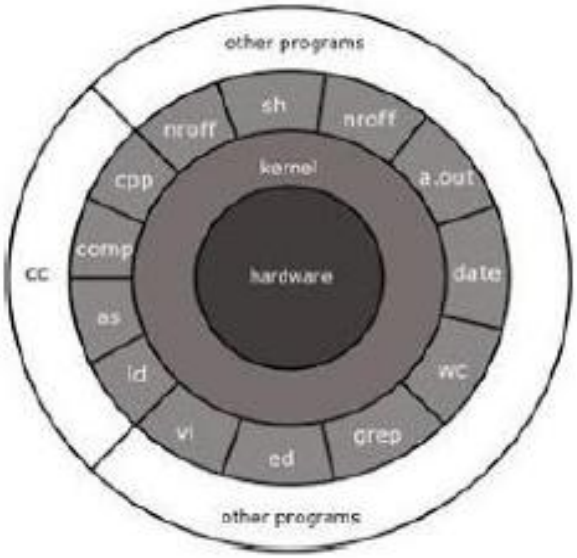
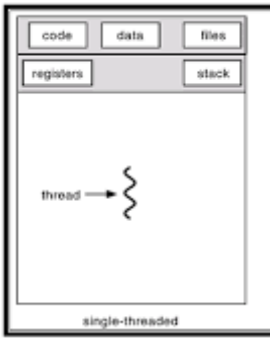
		The 1960's - Third Generation: Third generation 1965 – 1980 - ICs and multiprogramming. The systems of the 1960's were also batch processing systems, but they were able to take better advantage of the computer's resources by running several jobs at once. So operating systems designers developed the concept 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)	**Describe single level and two level directory structures.**	**4 M**
	Ans:	Single level directory is simple to implement but each file must have a unique name. In this all file are stored in the same directory. A single-level directory has significant limitations, however, when the number of files increases or when there is more than one user. Since all files are in the same directory, they must have unique names. If there are two users who call their data file "test", then the unique-name rule is violated. Even with a single-user, as the number of files increases, it becomes difficult to remember the names of all the files in order to create only files with unique names	**(Explanation with diagram of single and two level directory structure:2 marks each)**



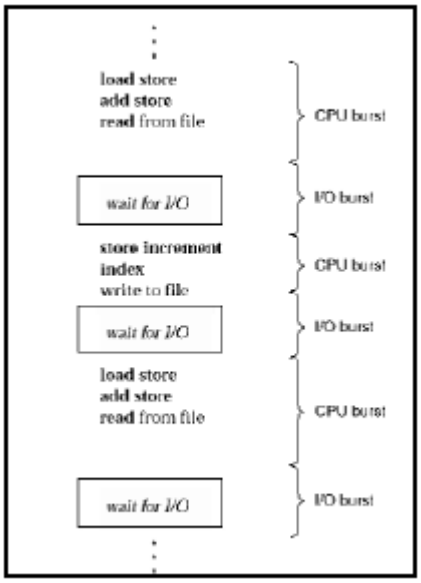
In two -level directory it create a separate directory for each user. In the two-level directory structure, each user has his own **user file directory** (UFD). The UFDs have similar structures, but each lists only the files of a single user. When a user job starts or a user logs in, the system's **master file directory** (MFD) is searched. The MFD is indexed by user name or account number, and each entry points to the UFD for that user.

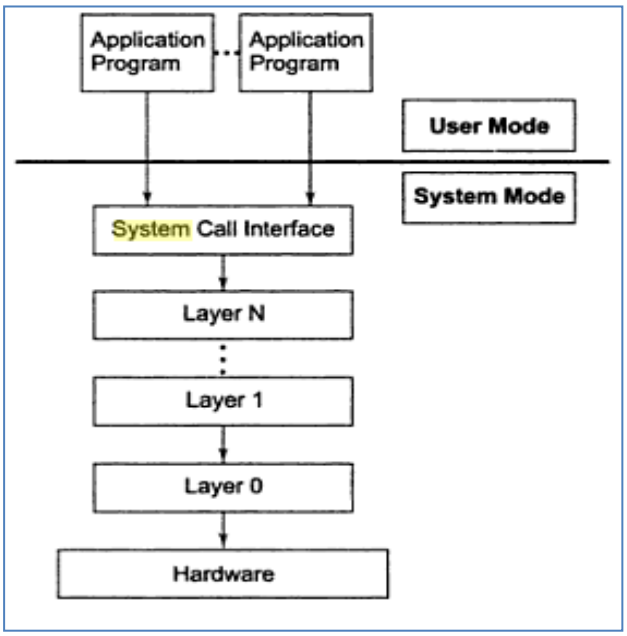


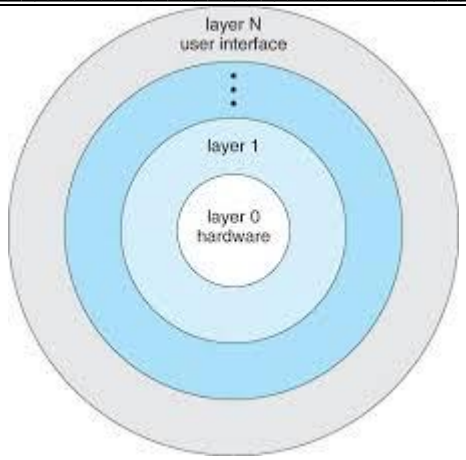
	(d)	**Explain structure of unix operating system with the help of diagram.**	**4 M**
	Ans:	The kernel of UNIX is the hub of the operating system: it allocates time and memory to Programs and handles the file store and communications in response to system calls. As an illustration of the way that the shell and the kernel work together, suppose a user types 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	**(Explanation : 2 marks, Diagram:2 marks)**

		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)	**Define thread. State any three benefits of thread.**	**4 M**
	Ans:	A thread, sometimes called a lightweight process, is a basic unit of CPU utilization. A traditional (or heavyweight) process has a single thread of control. If a process has multiple threads of control, it can do more than one task at a time. This is because there are situations in which it is desirable to have multiple threads of control in the same address space, running as though they were separate processes.  	(**Definition: 1 mark, Any three benefits: 1 mark each)**

		Threads benefits The benefits of multithreaded programming can be broken down into four major categories. **1. Responsiveness:** Multithreading an interactive application may allow a program to continue running even if part of it is blocked or is performing a lengthy operation, thereby increasing responsiveness to the user. **For example:** A multithreaded web browser could still allow user interaction in one thread while an image is being loaded in another thread. A multithreaded Web server with a front-end and (thread) processing modules. **2. Resource sharing:** By default, 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. A word processor with three threads. **For example:** A multithreaded word processor allows all threads to have access to the document being edited. **3. Economy**: Because threads share resources of the process to which they belong, it is more economical to create and switch threads, than create and context switch processes (it is much more time consuming). For example: in Sun OS Solaris 2 creating a process is about 30 times slower than is creating a thread (context switching is about five times slower than threads switching). **4. Utilization of multiprocessor architectures**: The benefits of multithreading can be greatly increased in a multiprocessor architecture (or even in a single-CPU architecture), where each thread may be running in parallel on a different processor.	
✓	**(f)**	**Describe CPU and I/O burst cycle with the help of diagram.**	**4 M**
	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.	**(Explanation : 2 marks, Example with Diagram:2 marks)**

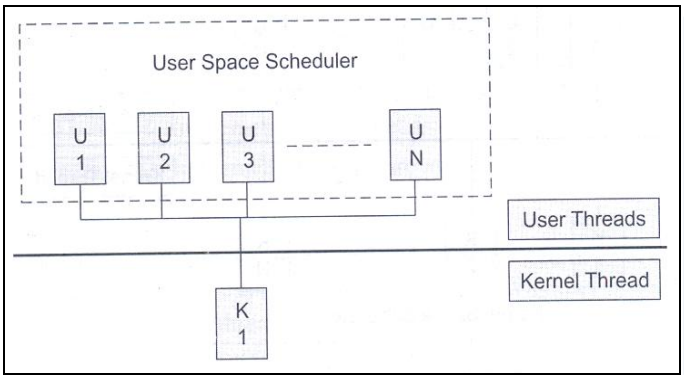
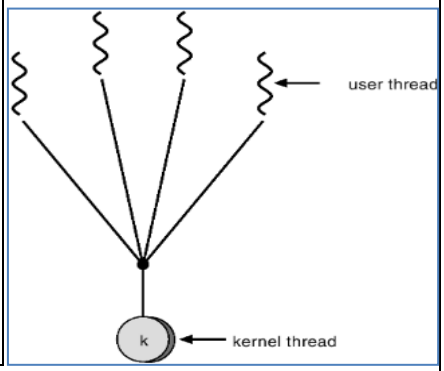
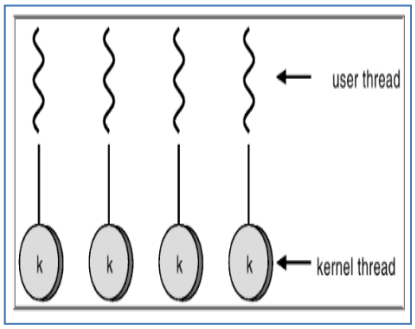
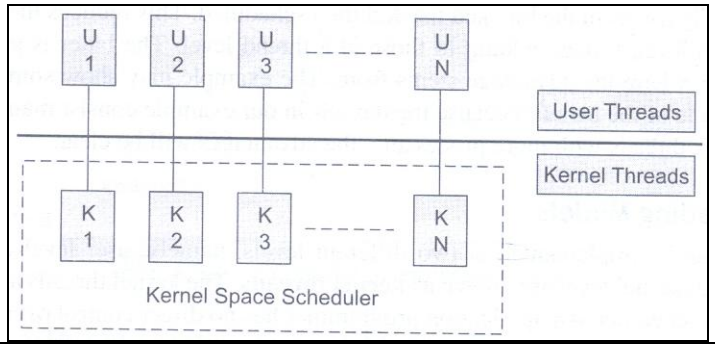


3.		**Attempt any FOUR :**	**16**
	(a)	**Explain layered operating system structure.**	**4 M**
	Ans:	**Structure:**  **OR**	**(Explanatio: 2 marks , diagram 2 marks)**

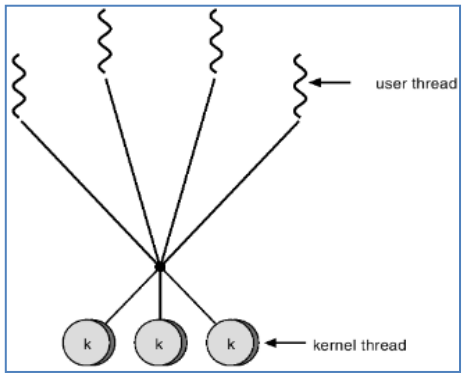


The modules of the operating system are divided into several layers stacked one above the other, thus forming a hierarchical structure. The lowest layer (Layer 0) interacts with the underlying hardware and the topmost layer (Layer N) provides an interface to the application programs/ users. Only adjacent layers can communicate with each other. A layer N can request for services only from a layer immediately below it (layer N-1). A layer N can provide services only to the layer immediately above it (layer N + 1). A Layer only needs to know what services are offered by the layer below it. In this structure any request that requires access to hardware has to go through all layers. Bypassing of layers is not allowed.

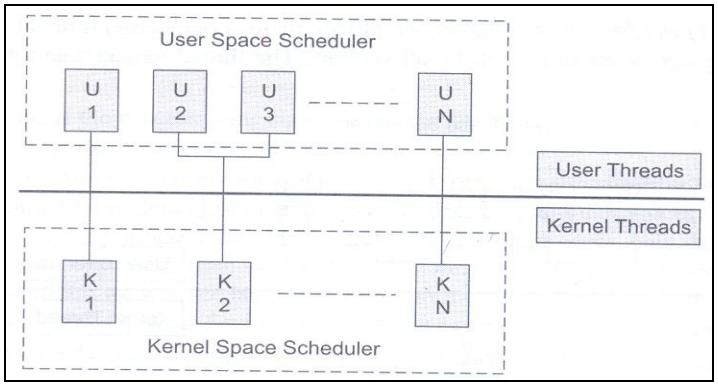
	(b)	**List & explain any four file attributes.**	**4 M**
	Ans:	A file's attributes vary from one operating system to another but typically consists of these: • Name: The symbolic file name is the only information kept in human readable form. Name of the file is usually a string of characters such as abc.c. When a file is named it becomes independent of the process, the user, and even the system that created it. • Identifier: This unique tag, usually a number, identifies the file system; it is the non-human-readable name for the file. • Type: This information is needed for systems that support different types of files. File types include text file, source file, object file, executable file, etc. • 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. It indicates protection permissions such as (r, w, and x) for each file or directory. • Time, date and user identification: This information may be kept for creation, last modification, and last use. These data can be useful for protection, security, and usage monitoring.	**(Any four attributes:1mark each)**

	(c)	**List & explain various types of multi-threading models.**	**4 M**
	Ans:	Multithreading models:- **1.** Many-to-One **2.** One-to-Many **3.** Many-to-Many **1. Many-to-One**: - This model maps many user level threads to one kernel level thread. Thread management is done by thread library in user space.   OR 1. **One-to-One**: It maps each user level thread to a kernel level thread. Even one thread makes a blocking call; other thread can run with the kernel thread.  OR  	**(List 1 mark, Each Type: 1 mark)**

2. **Many-to-many**: - This model maps many user level threads to a smaller or equal number of kernel threads. Number of kernel threads may be specific to either a particular application or particular machine.

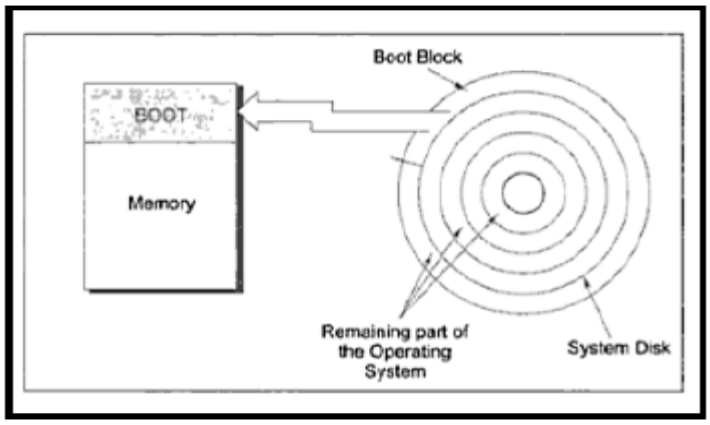


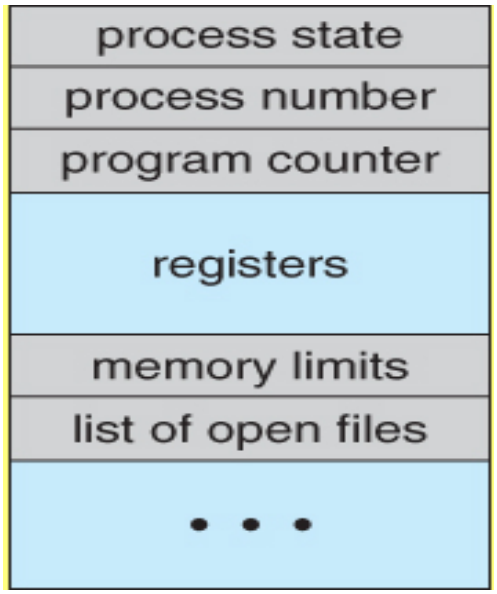
OR

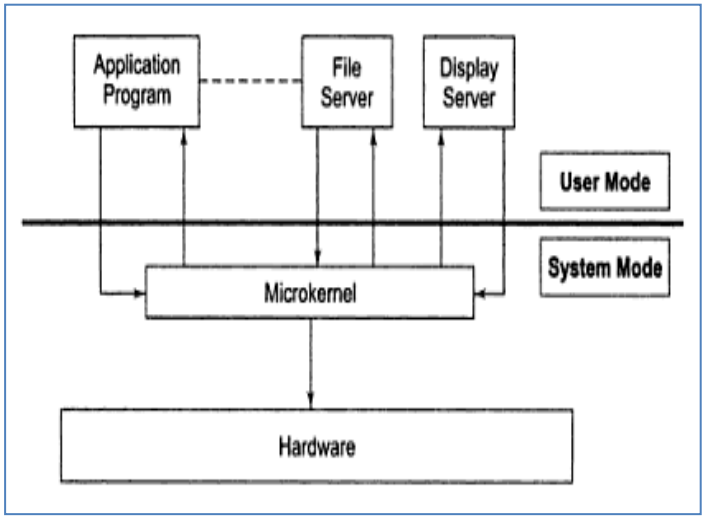
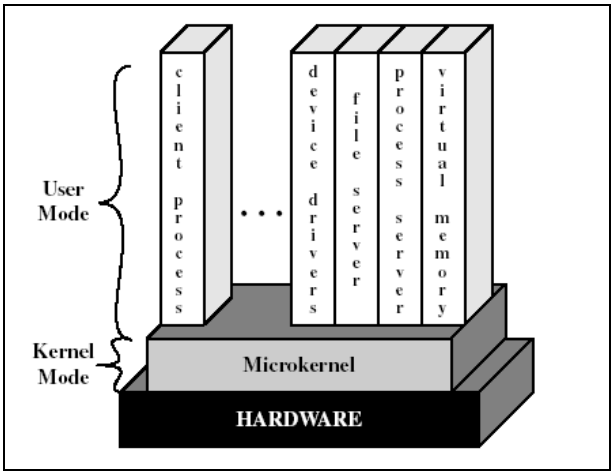


	(d)	**Explain in detail how deadlock can be handled.** {**Note Relevant explanation of deadlock avoidance or prevention shall be considered**}	**4M**
	Ans:	**Method for Handling Deadlocks** There are three different methods for dealing with the deadlock problem: • We can use a protocol to ensure that the system will never enter a deadlock state • We can allow the system to enter a deadlock state and then recover. • We can ignore the problem all together, and pretend that deadlocks never occur in system. This solution is the one used by most operating systems, including UNIX. 1. To ensure that deadlocks never occur, the system can be use either a deadlock –	**(Explanation 4 marks)**

		prevention or a deadlock-avoidance scheme. Deadlock prevention is a set of methods for ensuring that at least one of the necessary conditions cannot hold. These methods prevent deadlocks by constraining how requests for resources can be made. 2. Deadlock avoidance, on the other hand, requires that the operating system be given in advance additional information concerning which resources a process will request and use during its lifetime .with this additional knowledge, we can decide for each request whether or not the process, and the future requests and releases of each process, to decide whether the current request can be satisfied or must be delayed. 3. If the system does not employ either a deadlock-prevention or deadlock – avoidance algorithm, then a deadlock situation may occur .in this environment, the system can provide an algorithm that examines the state of the system to determine whether a deadlock has occurred, and an algorithm to recover from the deadlock. 4. If a system does not ensure that a deadlock will never occur, and also does not provide a mechanism for deadlock detection and recovery, then we may arrive at a situation where the system is in a deadlock state yet has no way of recognizing what has happened	
	(e)	**State and explain criteria used in differentiating CPU scheduling.**	**4M**
	Ans:	• **CPU utilization: -** In multiprogramming the main objective is to keep CPU as busy as possible. CPU utilization can range from 0 to 100 percent. • **Throughput: -** It is the number of processes that are completed per unit time. It is a measure of work done in the system. When CPU is busy in executing processes, then work is being done in the system. Throughput depends on the execution time required for any process. For long processes, throughput can be one process per unit time whereas for short processes it may be 10 processes per unit time. • **Turnaround time:-**The time interval from the time of submission of a process to the time of completion of that process is called as turnaround time. It is the sum of time period spent waiting to get into the memory, waiting in the ready queue, executing with the CPU, and doing I/O operations. It indicates the time period for which a process exists in the system. • **Waiting time: -** It is the sum of time periods spent in the ready queue by a process. When a process is selected from job pool, it is loaded into the main memory (ready queue).A process waits in ready queue till CPU is allocated to it. Once the CPU is allocated to the process, it starts its execution and if required request for resources. When the resources are not available that process goes into waiting state and when I/O request completes, it goes back to ready queue. In ready queue again it waits for CPU allocation. • **Response time:-**The time period from the submission of a request until the first response is produced is called as response time. It is the time when system responds to the process request not the completion of a process. In the system, a process can	**(Any four criteria- 1 mark each)**

		produce some output fairly early and can continue computing new results while previous results are being output to the user.	
4.	**(A)**	**Attempt any THREE :**	**12**
	(a)	**Describe the steps involved in booting process.**	**4M**
	Ans:	The loading of the operating system is done with the help of a special program called BOOT. This program is stored in one or two sectors on the disk with a pre-determined address. This portion is called as BOOT Block. The ROM contains minimum program called as bootstrap program. When the computer is turn ON, the control is transferred to this program automatically by the hardware itself. This program in ROM locates the BOOT program and loads it into predetermined memory locations. This BOOT program loads the operating System into the memory.  	**(Description of steps 3 marks, Diagram: 1 mark)**
	(b)	**Explain process control block with suitable diagram.**	**4M**

	Ans:	**Diagram:**  **Explanation:** Each process is represented as a process control block (PCB) in the operating system. It contains information associated with specific process. **Process State**: It indicates current states of a process. Process state can be new, ready, running, waiting and terminated. **Process number**: Each process is associated with a unique number. **Process Counter**: It indicates the address of the next instruction to be executed for the process. **CPU Registers**: The registers vary in number and type depending on the computer architecture. Register includes accumulators, index registers, stack pointers and general purpose registers plus any condition code information. **Memory Management Information**: It includes information such as value of base and limit registers, page tables, segment tables, depending on the memory system used by OS. **Accounting Information**: This information includes the amount of CPU used, time limits, account holders, job or process number and so on. It also includes information about listed I/O devices allocated to the process such as list of open files. Each PCB gives information about a particular process for which it is designed.	**(Explanation : 2 marks, correct diagram: 2 marks)**

	(c)	**Explain microkernel operating system structure.**	**4M**
	Ans:	**Structure**:-  **OR**  **Working:-** In this system, kernel provides only the most essential operating system functions like process management, communication primitives and low level memory management. System programs and user level programs implemented outside the kernel, provides the remaining operating system services. These programs are known as servers. Due to separation of functionality of kernel, size of the kernel is reduced. This reduced kernel is called as microkernel. The application programs and various servers communicate with each other using messages that passed through microkernel. The microkernel validates the messages and passes them between the various modules of the operating system and permits access to the hardware.	**(Explanation : 2 marks, diagram: 2 marks)**

(d) Differentiate between short term, medium term and long term scheduling **4M**

Ans: **(Any other relevant point shall be considered.1 mark each)**

Sr. No.	long term scheduler	Medium term scheduler	short term scheduler
1	It is a job scheduler	It is a process swapping scheduler	It is a CPU scheduler
2	It selects processes from job pool and loads them into memory for execution.	It selects a process from swapped-out process.	It selects processes from ready queue which are ready to execute and allocates CPU to one of them.
3	Access job pool and ready queue	Access process from swapped out process queue.	Access ready queue and CPU.
4	It executes muchless frequently when ready queue has space to accommodate new process.	It executed whenever swapped queue contains a swapped out process.	frequently select a new process for the CPU, at least once every100 milliseconds
5	Speed is less than short term scheduler	Speed is in between both short and long term scheduling	Speed is fast
6	It is almost absent or minimal in time sharing system	It is a part of time sharing system	It is also minimal in time sharing system
7	It controls the degree of multiprogramming	It reduces the degree of multiprogramming	It provides lesser control over degree of multiprogramming

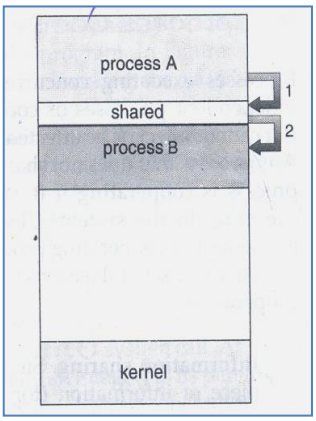
(B) Attempt any ONE : **6**

(a) Draw and explain inter-process communication model. **6M**

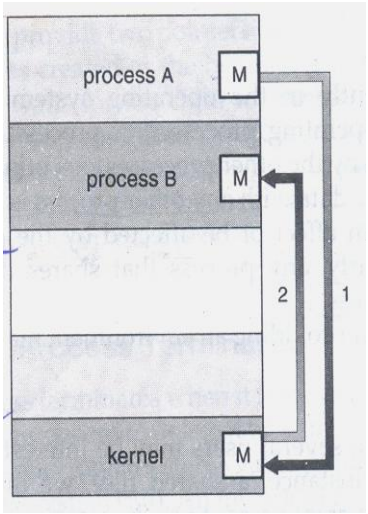
Ans: **(Two models: explanation 2 marks each, Diagram 1 mark each)**

With inter process communication, cooperative processes exchange data and information with each other.

1. Shared memory model: - In this model, a region of memory that can be shared by cooperating processes is established. A shared memory area resides in the address space of the process creating the shared memory segment. The processes that wish to communicate using the shared memory segment must attach it to their address space. All processes sharing the region can exchange the information by reading and writing data in the shared areas.

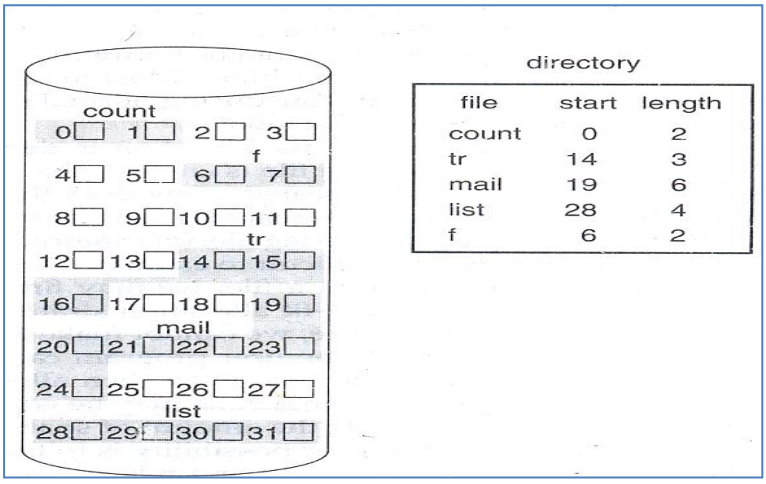


2. Message passing model: - In message passing model, a cooperating process communicates by exchanging messages with other processes. It allows processes to communicate and to synchronize their actions without sharing the same address space. It is particularly useful in a distributed environment, where the communicating processes may reside on different computers connected by a network. For example, chat program that allows exchange of information among processes residing on different computer.



	(b)	**List different file allocation methods. Explain any one with suitable diagram and example.**	**6M**
	Ans:	**Different file allocation methods:** 1. Contiguous file allocation 2. Linked file allocation 3. Indexed file allocation	**(List of file allocation methods: 2**

1. Contiguous Allocation:-



In this method, each file occupies contiguous blocks of memory space on the disk. Disk addresses specify linear ordering on the disk. When a file has to be stored on a disk, system search for contiguous set of blocks as required by the file size i.e. system waits till it finds required number of memory blocks in sequence. When space is available system stores the file in the disk and makes an entry in the directory. Directory entry contains name of the file, starting address of the block allocated to the file and length of the file (number of blocks allocated to the file).if the file is n blocks long and starts at location b ,then it occupies blocks b,b+1,b+2,…,b+n-1.Both sequential access and direct access supports contiguous allocation. For sequential access, the file system remembers the disk address of the last block referenced and when required reads the next block. For direct access to block i of a file that starts at block b, we can access with block b+i.

2. Linked Allocation:-

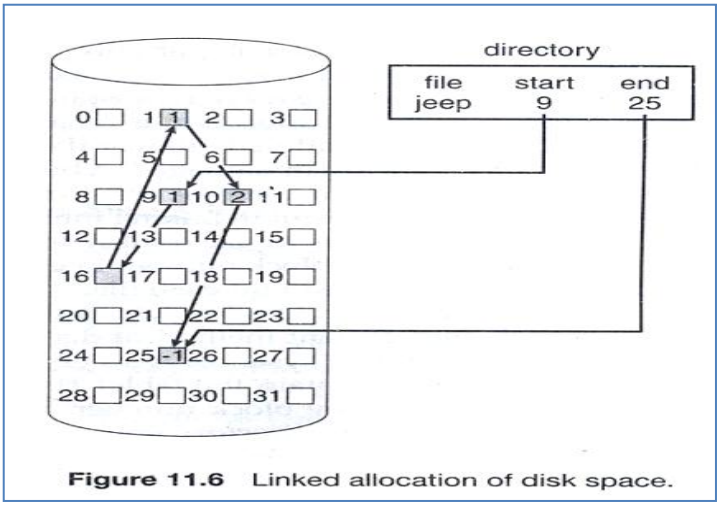


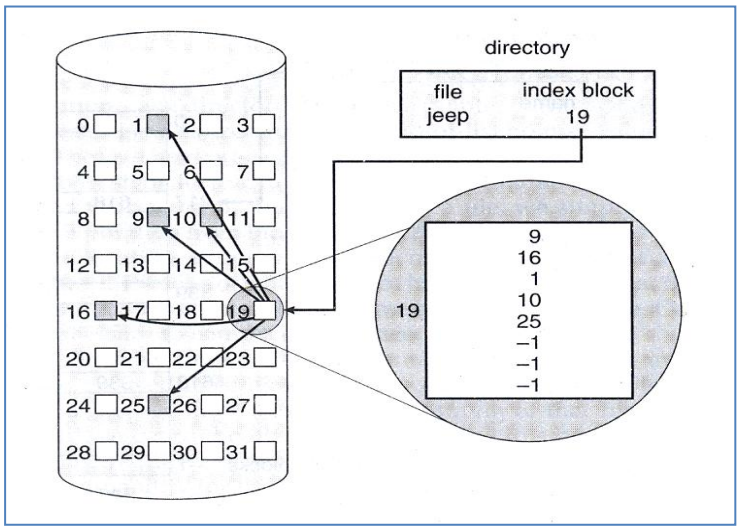
Figure 11.6 Linked allocation of disk space.

marks, explanation of any one: 2 marks, diagram/ example: 2 marks)

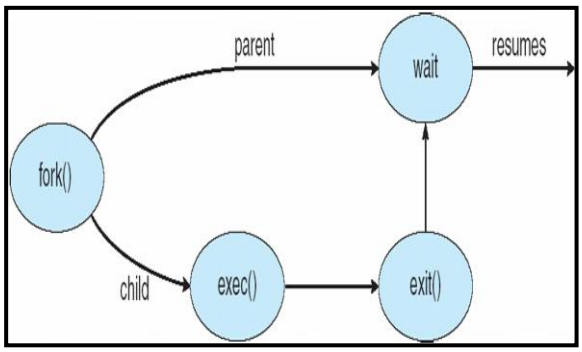
In this method, each file occupies disk blocks scattered anywhere on the disk. It is a linked list of allocated blocks. When space has to be allocated to the file, any free block can be used from the disk and system makes an entry in directory. Directory entry for allocated file contains file name, a pointer to the first allocated block and last allocated block of the file. The file pointer is initialized to nil value to indicate empty file. A write to a file, causes search of free block. After getting free block data is written to the file and that block is linked to the end of the file. To read the file, read blocks by following the pointers from block to block starting with block address specified in the directory entry.

For example, a file of five blocks starting with block 9 and continue with block 16,then k 1,then block 10 an finally block 25.each allocated block contains a pointer to the next k.If each block is 512 bytes in size and a disk address pointer requires 4 bytes, then the can see block of 508 bytes.

3. Indexed Allocation:-



In this method, each file has its own index block. This index block is an array of disk block addresses. When a file is created, an index block and other disk blocks according to the file size are allocated to that file. Pointer to each allocated block is stored in the index block of that file. Directory entry contains file name and address of index block. When any block is allocated to the file, its address is updated in the index block. Any free disk block can be allocated to the file. Each i^{th} entry in the index block points to the itch block of the file. To find and read the i^{th} block, we use the pointer in the i^{th} index block entry. It supports direct access without suffering from external fragmentation.

5.		Attempt any TWO:	16
	(a)	**Define process. Describe process creation and termination.**	**8M**
	Ans:	Process is 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. A process in execution needs resources like CPU, memory; I/O. Current machines allow several processes to share resources. **Process creation:** Create Process Operating system creates a new process with the specified or default attributes and identifier. A process may create several new sub processes. 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 sub process. 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.  **2.Terminate a Process:** Process executes last statement and asks the operating system to delete it (**exit**) a. Output data from child to parent (via **wait**) b. Process' resources are de allocated 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. Parent may terminate execution of children processes (**abort**) 1. The task given to the child is no longer required. 2. Child has exceeded its usage of some of the resources that it has been allocated. 3. Operating system does not allow a child to continue if its parent terminates, all children terminated - **cascading termination**	**(Definition of Process:2 marks, Process creation: 3 marks, Process termination: 3 marks)**

	(b)	**Explain the pre-emptive and non- pre-emptive type of scheduling .State when pre-emptive and non-pre-emptive type scheduling is used.**	**8M**
	Ans:	**a. Pre-emptive scheduling** Even if CPU is allocated to one process, CPU can be pre-empted to other process if other process is having higher priority or some other fulfilling criteria. Whereas the pre-emptive scheduling is based on priority where a scheduler may pre-empt a low priority running process anytime when a high priority process enters into a ready state. Circumstances for pre-emptive: • process switch from running to ready state • process switch from waiting to ready state e.g. SJF, Priority, Round Robin **b. Non pre-emptive scheduling** Once the CPU has been allocated to a process the process keeps the CPU until releases CPU either by terminating or by switching to waiting state Non-pre-emptive algorithms are designed so that once a process enters the running state, it cannot be pre-empted until it completes its allotted time, Circumstances for Non pre-emptive: • When process switches from running to waiting state • When process terminates E.g. FCFS. **When pre-emptive and Non-pre-emptive type Scheduling is used :** **Non-pre-emptive:** Circumstances for Non pre-emptive: • When process switches from running to waiting state • When process terminates **Pre-emptive Scheduling:** Circumstances for pre-emptive: • process switch from running to ready state • process switch from waiting to ready state	**(Explanation of pre-emptive & non-pre-emptive: 2 marks, each, when it is used: 2 marks each)**
	(c)	**Explain priority scheduling algorithm with example. List its advantages and disadvantages.**	**8M**
	Ans:	**Priority scheduling algorithm:** The SJF algorithm is a special case of the general priority scheduling algorithm Number (integer) is associated with each process. The CPU is allocated A priority to the process with the highest priority (smallest integer = highest priority) Priority scheduling can be either pre-emptive or non-pre-emptive • A pre-emptive priority algorithm will pre-emptive the CPU if the priority of the newly arrival process is higher than the priority of the currently running process. • A non-pre-emptive priority algorithm will simply put the new process at the head of the ready queue.	**(Description priority:2 marks, example of priority: 4 marks, advantages & Disadvantage : 2 marks)**

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. SJF is a priority scheduling where priority is the predicted next CPU burst time

Advantage Priority Scheduling-

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

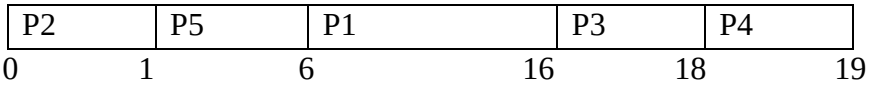
Disadvantages of Priority Scheduling-

- Indefinite blocking or starvation.
- A priority scheduling can leave some low priority waiting processes indefinitely for CPU.
- If the system eventually crashes then all unfinished low priority processes gets lost.

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 FOUR:**	**16**
	(a)	**What are the different responsibilities of memory management? Explain.**	**4M**
	Ans:	The services provided under memory management are directed to keeping track of memory and allocating/de allocating it to various processes. The OS keeps a list of free Memory locations. (Initially, after the booting but before any process starts, the full memory, excepting the part occupied by the OS itself is free.)Before a program is loaded in the memory from the disk, this module (MM) consults this free list, allocates the memory to the process, depending upon the program size and updates the list of free memory.	**(Explanation :4 marks)**

Main memory management:

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

Functions of memory management:

1) Keeping track of which part of memory are currently being used and by whom

2) Deciding which processes are to be loaded in to memory when memory space becomes available.

3) Allocating& De allocating space as needed

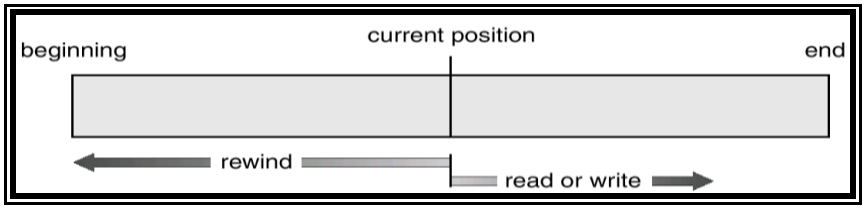
(b) Differentiate Between linux and unix. **4M**

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, etc..	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.
Number of shells	Sh, bash, csh and tsh,ksh	B, C, K, Bash, tcsh, zsh

(Any four points 1 mark each)

			Architecture	Small scale (is available on PARIS and Itanium machines. Solaris also available for x86/x64 based systems.)	High configuration (Originally developed for Intel's x86 hardware, ports available for over two dozen CPU types including ARM)	
			Applications	Command base (The UNIX operating system is used in internet servers, workstations & PCs. Backbone of the majority of finance 32 infrastructure and many 24x365 high availability solution. GUI: Initially Unix was a command based OS, but later a GUI was created called Common Desktop Environment. Most distributions now ship with Gnome.)	GUI based(Linux can be installed on a wide variety of computer hardware, ranging from mobile phones, tablet computers and video game consoles, to mainframes and super computers. GUI: Linux typically provides two GUIs, KDE and Gnome. But Linux GUI is optional.)	
			Case Of Operation	It requires Less no. of Kernel for operation.	It requires Multiple kernel for operation	
			system requirement	Hardware on UNIX Random Access Memory (RAM) 256MB recommended 128MB minimum 250MB available hard drive space CD-ROM drive TCP/IP network interface A persistent Internet Connection	Red Hat Enterprise Linux OS CPU Types –Pentium4 or higher; GHz or higher Memory/RAM – 1GB minimum, up to the system limit Hard Disk – 4GB minimum Other - to read the Directory Server using port number lass then 1024, such as the default you can port 389, you can set up and start the directory server as root, but it is not necessary to run the Directory Server root.	
	(c)	**Describe Distributed Operating system.**				**4M**
	Ans:	Distributed systems use multiple central processors to serve multiple real-time applications and multiple users. Data processing jobs are distributed among the				**(Description of distributed**

		processors accordingly. The processors communicate with one another through various communication lines (such as high-speed buses or telephone lines). These are referred as **loosely coupled systems** or distributed systems. Processors in a distributed system may vary in size and function. These processors are referred as sites, nodes, computers, and so on. The advantages of distributed systems are as follows − • With resource sharing facility, a user at one site may be able to use the resources available at another. • Speedup the exchange of data with one another via electronic mail. • If one site fails in a distributed system, the remaining sites can potentially continue operating. • Better service to the customers. • Reduction of the load on the host computer. • Reduction of delays in data processing.	**OS : 4 marks)**
	(d)	**Describe sequential file access method.**	**4M**
	Ans:	Information in the file is processed in order, one record after the other. This is by far the most common mode of access of files. For example, computer editors usually access files in this fashion. A read operation reads the next portion of the file and automatically advances the file pointer. Similarly, a write appends to the end of the file and the file pointer. Similarly, a write appends to the end of the file and the file pointer. Similarly, a write appends to the end of the end of the file and advances to the end of the newly written material (the new end of file). Such a file can be reset to the beginning, and, on some systems, a program may be able to skip forward or backward n records, for some integer n. This scheme is known as sequential access to a file. Sequential access is based on a tape model of a file.  	**(Description: 2 marks, Diagram : 2 marks)**
	(e)	**Explain context switch with suitable example.**	**4M**

Ans:

Context Switch

A context switch is the mechanism to store and restore the state or context of a CPU in Process Control block so that a process execution can be resumed from the same point at a later time. Using this technique a context switcher enables multiple processes to share a single CPU. Context switching is an essential part of a multitasking operating system features.

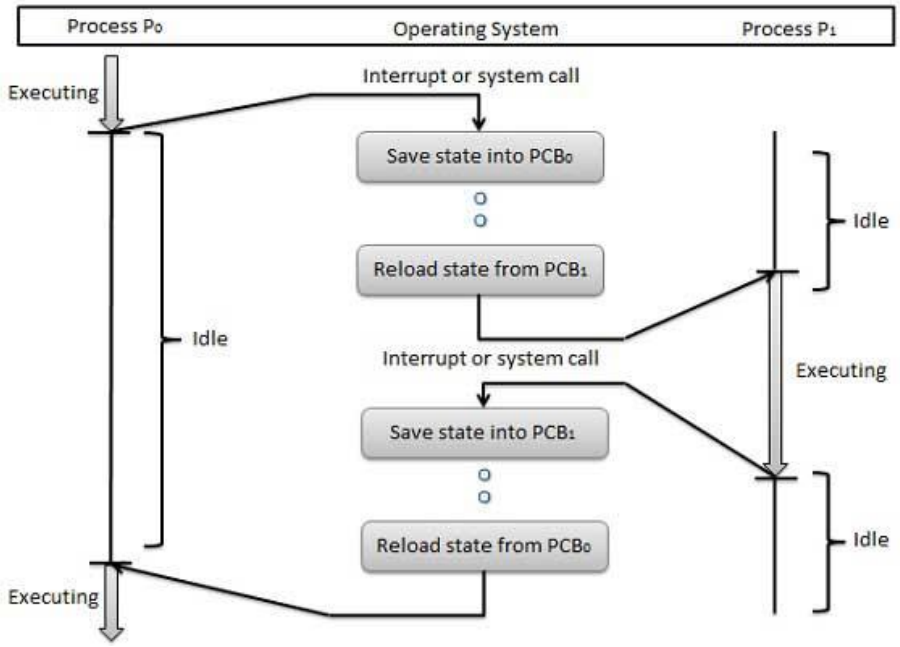
When the scheduler switches the CPU from executing one process to execute another, the context switcher saves the content of all processor registers for the process being removed from the CPU, in its process descriptor. The context of a process is represented in the process control block of a process.

Context switch time is pure overhead. Context switching can significantly affect performance as modern computers have a lot of general and status registers to be saved. Content switching times are highly dependent on hardware support.

Example:

We can run two processes at the same time. When process1 (Po) waits for an I/O, process2 (P1) executes and when process2 waits for I/O, process1 executes. Some time is required for turning CPU's attention from process1 to process2 called context switching.

After the context switch the old program will remain in the main memory. The status of CPU registers and the pointers to the memory allocated to this process must be stored. A specific memory area is used by OS which maintained for each process. This area is called as register save area which is a part of PCB.



Some hardware systems employ two or more sets of processor registers to reduce the amo of context switching time. When the process is switched, the following information is store

- Program Counter
- Scheduling Information

(Explanation of context switch:2 marks, example : 2 marks)

		• Base and limit register value • Currently used register • Changed State • I/O State • Accounting	