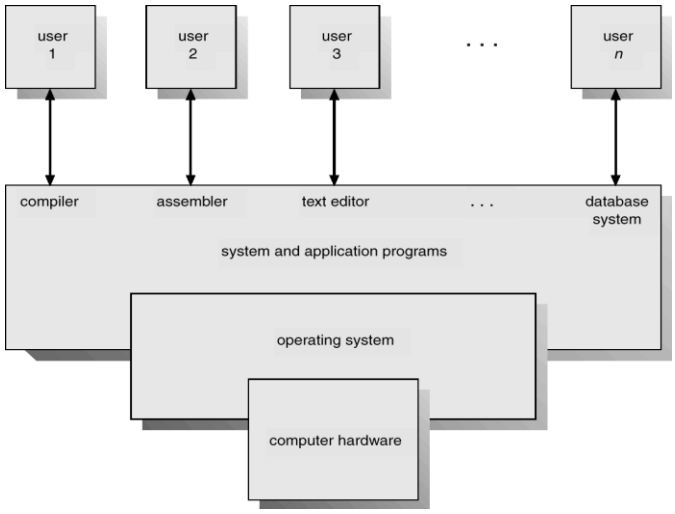
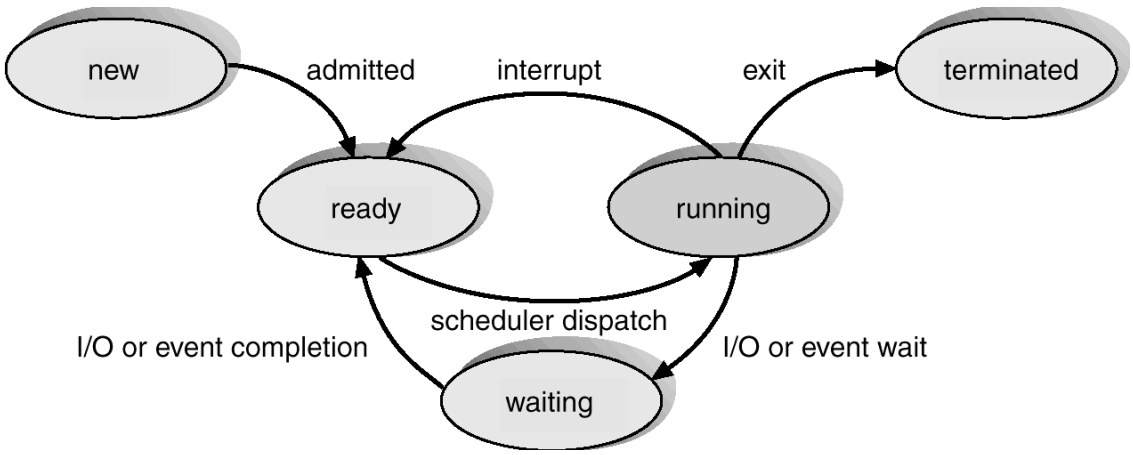
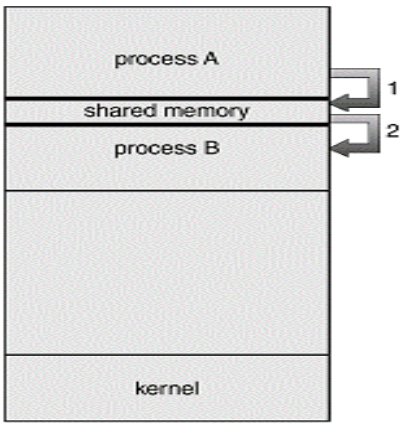
Subject Name: Operating System **Model Answer** Subject Code: 17512

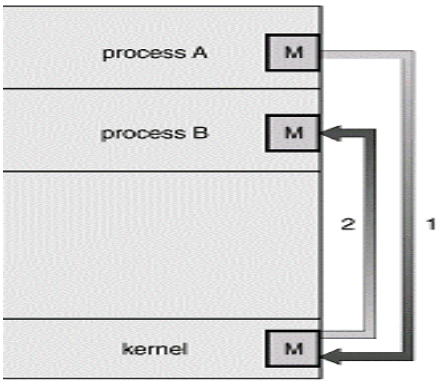
Important Instructions to examiners:

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.	Answers	Marking Scheme
1.	**a)**	**Attempt any THREE of the following:**	**12 Marks**
	(i)	**List and draw a neat labelled diagram of four components of a computer system.**	**4M**
	Ans:	A computer system can be divided into four components: 1) The hardware. 2) The operating system 3) Application programs 4) The users **Diagram:**  	**(List: 2marks, Correct diagram: 2 marks)**

	(ii)	**List three main levels of data storage and explain cache storage.**	**4M**
	Ans:	**{{**Note: - Any other relevant explanation shall be considered. **}}** **Three levels of data storage:** 1. Primary Storage 2. Secondary Storage 3. Tertiary Storage **Cache Storage:** A Cache (Pronounced as "cash") is a small and very fast temporary storage memory. It is designed to speed up the transfer of data and instructions. It is located inside or close to the CPU chip. It is faster than RAM and the data/instructions that are most recently or most frequently used by CPU are stored in cache. As CPU has to fetch instruction from main memory speed of CPU depending on fetching speed from main memory. CPU contains register which has fastest access but they are limited in number as well as costly. Cache is cheaper so we can access cache. Cache memory is a very high speed memory that is placed between the CPU and main memory, to operate at the speed of the CPU. It is used to reduce the average time to access data from the main memory. The cache is a smaller and faster memory which stores copies of the data from frequently used main memory locations. Most CPUs have different independent caches, including instruction and data.	**(List:1 mark, Relevant explanation of cache storage: 3 marks)**
	(iii)	**List and draw a neat labelled diagram of process state.**	**4M**
	Ans:	**List:** Each process may be in one of the following states. • New State: The process is being created. • Ready State: The process is waiting to be assigned the processor. • Running State: Instructions from the process are executing. • Waiting or Blocked: The process is waiting for some event to occur. • Terminated State: The process has finished execution. **Process state diagram:**  	**(List:2 marks, Diagram: 2 marks)**

	(iv)	**List merits of I/O scheduling (Four points) and Demerits of I/O scheduling.**	**4M**
	Ans:	**{{**Any other relevant Merits and Demerits shall be considered*}}** **Merits of I/O scheduling:** • It improves overall performance of the system. • It can share device access fairly among processes. • It helps in reducing the average waiting time for I/O to complete. • It increases throughput of the system. • It helps to prioritize process's I/O requests. **Demerits of I/O scheduling:** • There is a risk of starvation for longer processes. • may lead to poor overlap of I/O and CPU since CPU-bound processes will force I/O bound processes to wait for the CPU, leaving the I/O devices idle • Difficult to know the length of the next CPU request • An I/O bound process on a heavily loaded system will run slower • May involve a large context switch overhead	**(Merits: 2 marks, Demerits: 2 marks)**
	b)	**Attempt any <u>ONE</u> of the following:**	**6 Marks**
	(i)	**Explain the working of Inter-process communication considering.** **1) Shared memory** **2) Message passing**	**6M**
	Ans:	**1)Shared memory:** In this model, 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.  (a) Shared Memory	**(Diagram: 1mark each, Explanation :2 marks each)**

		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 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 exist.  (b)Message Passing	
	(ii)	**List four Deadlock prevention condition and explain the following terms.** **1) Removal of "No preemption" condition.** **2) Elimination of "Circular wait" related to deadlock prevention condition.**	**6M**
	Ans:	**Deadlock prevention conditions:-** 1. Preventing Mutual exclusion condition 2. Preventing Hold and wait condition 3. Preventing No preemption condition 4. Preventing Circular wait condition **1) Removal of "No Preemption" Condition** This necessary condition specifies that there is no pre-emption of resources that have already been allocated. To ensure that this condition does not hold, we can use the following protocol. If a process is holding some resources and requests another resource that cannot be immediately allocated to it (that is, the process must wait), then all resources the process is currently holding are preempted. In other words, these resources are implicitly released. The pre-empted resources are added to the list of resources for which the process is waiting. The process will only be restarted when it can regain its old resources, as well as the new ones that it is requesting. For example: If a process requests some resources, we first check if they are available. If so we allocate them. If they are not available, we check whether they are allocated to some other process that is waiting for additional resources. If so, we pre-empt the desired resources from the waiting or held by a waiting process, the requesting process must wait. While it is waiting, some of its resources may be pre-empted, but only if another process requests them. A process can only be restarted when it is allocated the new resources it is requesting and recovers any resources that we pre-empted while it was waiting. **2) Elimination of "Circular wait" related to deadlock prevention condition** If a circular wait condition is prevented, the problem of the deadlock can be prevented too.	**(List: 2 marks, Explanation of each term:2 marks)**

Consider all resources are numbered as shown in figure:

Number	Resource Name
0	Tape Drive
1	Printer
2	Plotter
3	Card Reader
4	Card Punch

Any process has to request for all the required resources in a numerically ascending order during its execution. This would prevent a deadlock. Let us assume that two processes P1 and P2 are holding a tape drive and a plotter respectively. A deadlock can take place only if P1 holds the tape drive and wants the plotter, whereas P2 holds the plotter and requests for the tape drive, i.e. if the order in which the resources are requested by the two processes is exactly apposite. And this contradicts our assumption. Because 0<2, a tape drive has to be requested for before a plotter, by each process, whether it is P1 or P2.

Each process can request resources only in an increasing order of enumeration. That is, a process can initially request any number of instances of a resource type -say, *R;*. After that, the process can request instances of resource type *Rj* if and only if F(Rj) > F(R;). We can demonstrate this fact by assuming that a circular wait exists. Let the set of processes involved in the circular wait be { *P0* , P1, ... , P11}, where *Pi* is waiting for a resource *R;,* which is held by process *Pi+l·* (Modulo arithmetic is used on the indexes, so that *P11* is waiting for a resource R11 held by P0 .) Then, since process *Pi+l* is holding resource *Ri* while requesting resource *Ri+l'* we must have F(Ri) < F(R;H) for all *i.* But this condition means that F(Ro) < F(R1) < ... < F(R11) < *F* (Ro). By transitivity, F(Ro) < F(Ro), which is impossible. Therefore, there can be no circular wait.

2. **Attempt any FOUR of the following:** **16Marks**

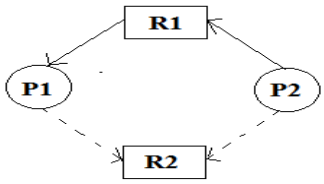
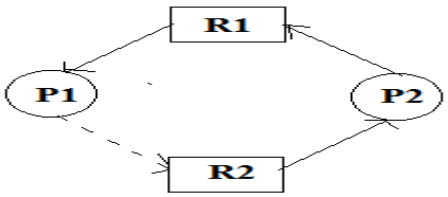
a) **Define clustered systems? List four characteristics of clustered systems.** **4M**

Ans: A clustered system is a collection of connected computers working together as one unit. In this system, any member of the cluster is capable of supporting the processing functions of any other member. Each computer in a cluster is referred to as a node. **(Defination :2 marks, any four relevant characteristics:2 marks)**

Characteristics:-

- Clustering allows two or more system to share storage closely linked via a local area network.
- A cluster has a redundant n+k configuration, where *n* processing nodes are actively processing the application and *k* processing nodes are in a standby state, serving as peers. In the event of a failure of an active node, the application that was running on the failed node is moved to one of the standby nodes. The simplest redundant configuration is active/standby, in which one node is actively processing the application and the other node is in a standby state.
- Other common cluster configurations include ***simplex*** (one active node, no spare), *n+1* ***active*** nodes (n active nodes, 1 spare), and *n* ***active*** nodes. In a configuration with n active nodes, the applications from the failed node are redistributed among the other active nodes using a pre-specified algorithm.
- It provides excellent fault tolerance.

	b)	**Explain following two services of operating systems.** **(i) File system manipulation** **(ii) Resource Allocation**	**4M**
	Ans:	• **File system manipulation:** While working on the computer, generally a user performs various types of operations on files like creating a file, opening a file, saving a file, deleting a file, search for a file with file name from the storage disk, etc. Programs need to read a file or write a files and directories. The operating system gives the permission to program for performing operations on file. Maintain details of files or directories with their respective details. Some programs include permissions management that allow or deny access to files or directories based on file ownership. • **Resource allocation:** When multiple users or multiple jobs are running at the same time, it is the responsibility of an operating system to allocate the required resources to each process. Operating system manages many different types of resources such as CPU, main memory, tape drive or secondary storage etc. For this purpose, various types of algorithms are implemented such as process scheduling, CPU scheduling, disk scheduling etc. There are some routines that allocate printers, modems, USB storage drives and other peripheral devices.	**(Explanation of each :2 marks)**
	c)	**Define synchronization** **Explain** **(i) Blocking** **(ii) Non Blocking in message passing**	**4M**
	Ans:	**Process Synchronization** means sharing system resources by processes in such a way that, concurrent access to shared data is handled thereby minimizing the chance of inconsistent data. Maintaining data consistency demands mechanisms to ensure synchronized execution of cooperating processes. Process Synchronization was introduced to handle problems that arose while multiple process executions. Message passing may be blocking or non-blocking, also known as synchronous and asynchronous. • **Blocking send:** The sending process is blocked until the message is received by the receiving process or by the mailbox. • **Blocking receives:** The receiver blocks until a message is available. • **Non-blocking send:** The sending process sends the message and resumes operation. • **Non-blocking receive:** The receiver retrieves either a valid message or a null.	**(Any relevant Definition of synchronization: 2marks, Each term Explanation:1mark)**
	d)	**List four process scheduling criteria and explain the term Turnaround in detail.**	**4M**
	Ans:	**Process scheduling criteria:** • CPU Utilization • Throughput • Turn-Around Time • Waiting Time • Response Time	**(List:2 marks, Explanation of turnaround :2 marks)**

		Turn-Around 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.	
	e)	**Explain Deadlock Avoidance with example.**	**4M**
	Ans:	Most prevention algorithms have poor resource utilization, and hence result in reduced throughputs. Instead, we can try to avoid deadlocks by making use prior knowledge about the usage of resources by processes including resources available, resources allocated, future requests and future releases by processes. Most deadlock avoidance algorithms need every process to tell in advance the maximum number of resources of each type that it may need. Based on all this info we may decide if a process should wait for a resource or not and thus avoid chances for circular wait. Deadlock can be avoided by following algorithms: **Safe State:** If a system is already in a safe state, we can try to stay away from an unsafe state and avoid deadlock. Deadlocks cannot be avoided in an unsafe state. A system can be considered to be in safe state if it is not in a state of deadlock and can allocate resources up to the maximum available. A safe sequence of processes and allocation of resources ensures a safe state. Deadlock avoidance algorithms try not to allocate resources to a process if it will make the system in an unsafe state. Since resource allocation is not done right away in some cases, deadlock avoidance algorithms also suffer from low resource utilization problem. **Resource Allocation Graph:** A resource allocation graph is generally used to avoid deadlocks. If there are no cycles in the resource allocation graph, then there are no deadlocks. If there are cycles, there may be a deadlock. If there is only one instance of every resource, then a cycle implies a deadlock. Vertices of the resource allocation graph are resources and processes. The resource allocation graph has request edges and assignment edges. An edge from a process to resource is a request edge and an edge from a resource to process is an allocation edge. A calm edge denotes that a request may be made in future and is represented as a dashed line. Based on calm edges we can see if there is a chance for a cycle and then grant requests if the system will again be in a safe state. Example:   If R2 is allocated to p2 and if P1 request for R2, there will be a deadlock. **Bankers Algorithm:** The resource allocation graph is not much useful if there are multiple instances for a resource. In such a case, we can use Banker's algorithm. In this algorithm, every process must tell upfront the maximum resource of each type it need, subject to the maximum available instances for each type. Allocation of resources is made only, if the allocation ensures a safe state; else the processes need to wait. The Banker's algorithm can be divided into two parts: Safety algorithm if a system is in a safe state or not. The resource request algorithm make an assumption of allocation and see if the system will be in a safe	**(Explanation of any one method: 2 marks, Example: 2 marks)**

state. If the new state is unsafe, the resources are not allocated and the data structures are restored to their previous state; in this case the processes must wait for the resource.

Example:

5 processes P_0 through P_4;
3 resource types:
A (10 instances), B (5instances), and C (7 instances)

Snapshot at time T_0:

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

	A B C
P_0	7 4 3
P_1	1 2 2
P_2	6 0 0
P_3	0 1 1
P_4	4 3 1

The system is in a safe state since the sequence < P_1, P_3, P_4, P_2, P_0> satisfies safety criteria

f) Explain "Bitmap" method in free space management technique. **4M**

Ans:

The free space management techniques Bitmap is also referred as Bit vector.

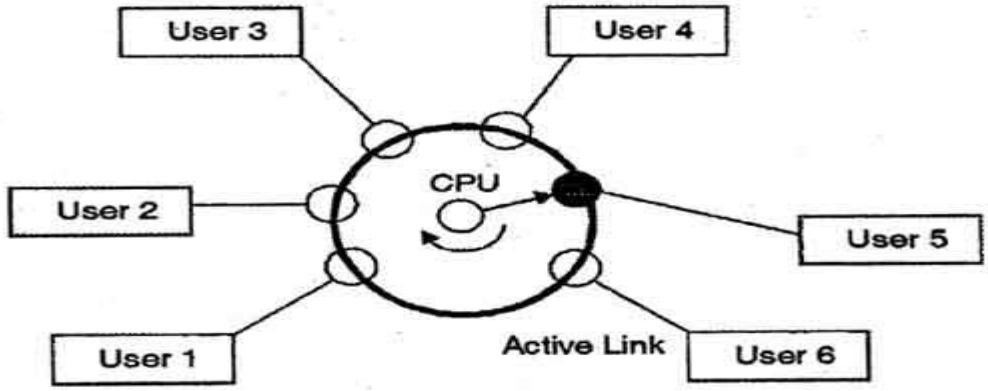
Bit vector:

The free space list is implemented as a Bit Map or Bit Vector. Each block is represented by one bit. If the block is free, the bit is '1'; if the block is allocated the bit is '0'.

Example considers a disk where blocks 2,3,4,5,8,9,10,11,12,13,17,18,25,26 and 27 are free the remaining blocks are allocated then the free space bit map would be: 00111100111111000110000001 1…

The main advantage of this approach is that it is relatively simple and efficient to find the first free blocks or n consecutive free blocks on the disk.

(Relevant Explanation of bitmap method:4 marks)

3.		**Attempt any FOUR of the following:**	**16Marks**
	a)	**Explain Time sharing OS in detail.**	**4M**
	Ans:	In time sharing system, the CPU executes multiple jobs by switching among them. The switches occur so frequently that the users can interact with each program while it is running. It includes an interactive computer system which provides direct communication between the user and the system. A time sharing system allows many users to share the computer resources simultaneously. The time sharing system provides the direct access to a large number of users where CPU time is divided among all the users on scheduled basis. The operating system allocates a time slice to each user. When this time is expired, it passes control to the next user on the system. The time allowed is extremely small and the users are given the impression that each of them has their own CPU and they are the sole owner of the CPU. In this time slice each user gets attention of the CPU. The objective of time sharing system is to minimize response time of process. Example:- The concept of time sharing system is shown in figure:  In above figure, the user 5 is active but user 1, user 2, user 3, and user 4 are in waiting state whereas user 6 is in ready status.	**(Relevant Explanation: 4 marks)**
	b)	**List types of system call and explain the system call – "Information Maintenance".**	**4M**
	Ans:	Types of System calls are as follows: 1. Process or Job control 2. File Management 3. Device Management 4. Information Maintenance 5. Communications **Information Maintenance:** The operating system keeps information about all its processes and provides system calls to access this information. Some system calls exist purely for transferring information between the user program and the operating system. Transferring information between the user program and the operating system requires system call. System information includes	**(List:1 mark, Explanation of information maintenance: 3 marks)**

displaying current date and time, the number of current user, the version number of the operating system, the amount of free memory or disk space and so on.

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.

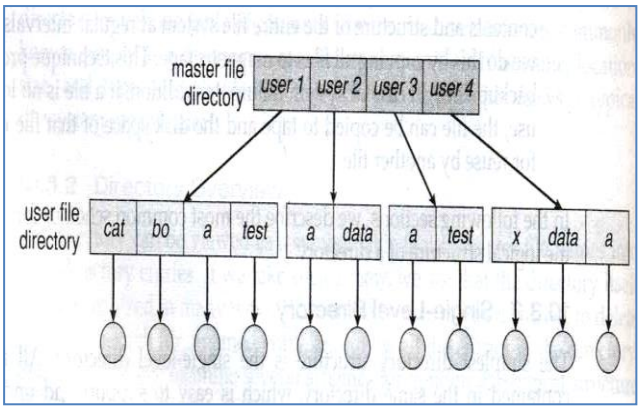
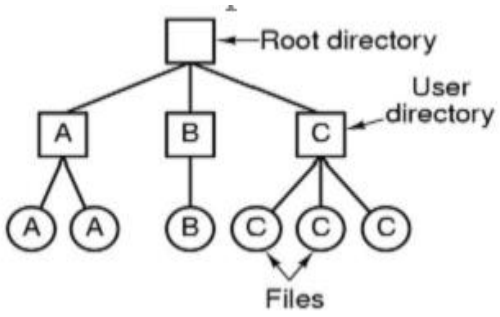
c) Differentiate between long term scheduling and medium term scheduling. **4M**

Ans: **(Any four points: 1 mark each)**

Sr. No.	Long Term scheduling	Medium term scheduling
1	In this scheduling, long term scheduler selects a process from job pool and loads into the memory.	In this scheduling, medium term scheduler, selects a process from swapped queue and loads into the memory.
2	It works with job pool and memory.	It works with swapped process queue and memory
3	It is a job scheduler.	It is a swapped process scheduler.
4	Processes switch its state from new to ready. A process selected from job pool enters to the memory for the first time.	Processes switch its state from blocked to ready. A process selected from swapped queue re-enters into the memory.
5	It controls the degree of multiprogramming	It does not the degree of multiprogramming.

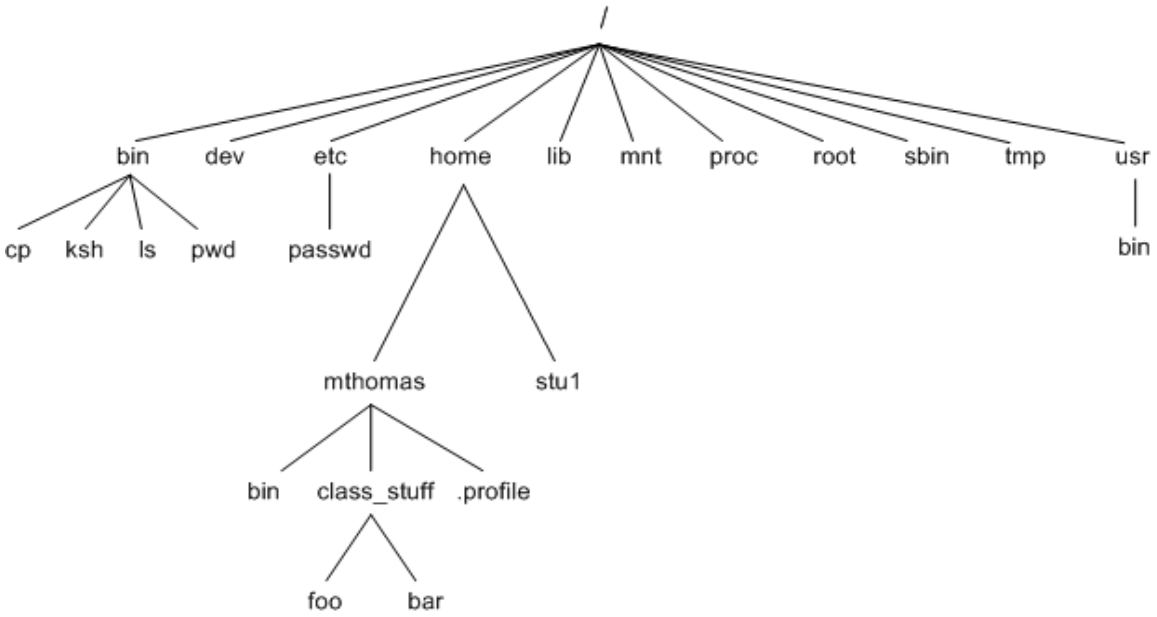
d) The Jobs are scheduled for execution as follows- solve the problem by using preemptive SJF (Shortest Job First). Find average waiting time using Gantt chart. **4M**

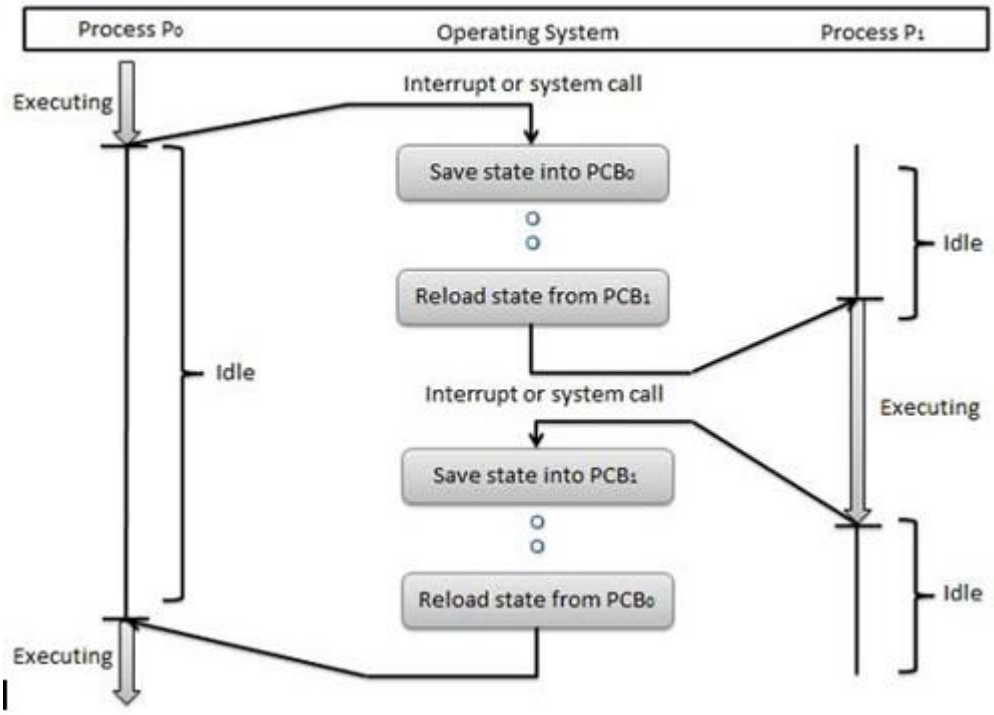
Process	Arrival Time	Burst Time
P1	**0**	**10**
P2	**1**	**04**
P3	**2**	**14**
P4	**3**	**08**

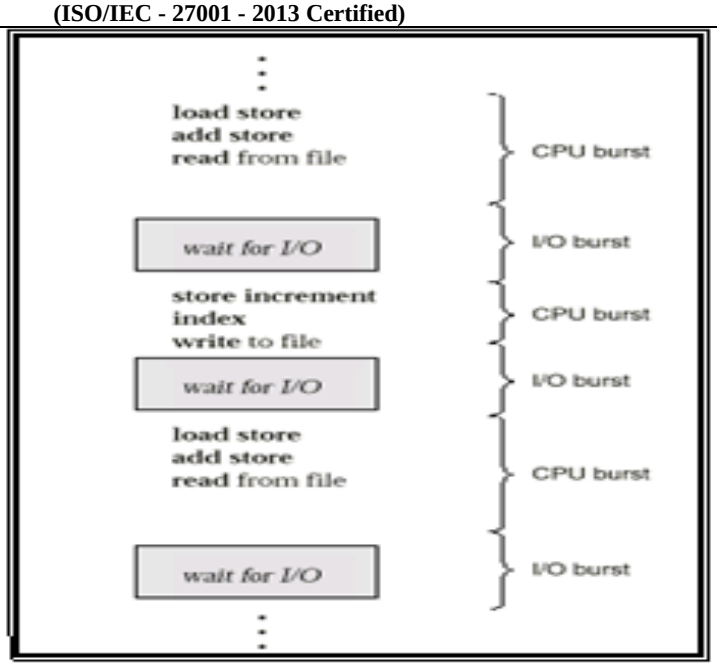
	Ans:	 Waiting Time P1=13-1=12 P2=1-1=0 P3=22-2=20 P4=5-3=2 Average waiting time= (12+0+20+2)/4=8.5 ms	**(Gantt chart:2 marks, Average Waiting Time: 2 marks)**
	e)	**Explain the working of Two-level directory structure with neat labelled diagram.**	**4M**
	Ans:	The two level directory structures, each user has its own user file directory (UFD). The UFD lists only files of a single user. System contains a master file directory (MFD) which is indexed by user name or account number. Each entry in MFD points to the UFD for that user.  **OR** 	**(Explanation: 2 marks, any correct Diagram : 2 marks)**

		When a user refers to a particular file, only his own UFD is searched. Different users can have files with the same name, as long as all the file names within each UFD are unique. When we create a file for a user, operating system searches only that user's UFD same name file already present in the directory. For deleting a file again operating system checks the file name in the user' UFD only.	
4.	**a)**	**Attempt any <u>THREE</u> of the following:**	**12Marks**
	(i)	**List Advantages and Disadvantages of Batch Monitoring functions. (Four points)**	**4M**
	Ans:	**Advantages of batch systems:-** ● Grouping the jobs in batch reduce the time required for loading system setup for execution of each job. ● No need of special hardware and system support to input data in batch systems. ● Best for large organizations but small organizations can also benefit from it. ● Batch systems can work offline so it makes less stress on processor. ● Processor consumes good time while processing that mean it knows which job to process next. In real time systems we don't have expectation time of how long the job is and what is estimated time to complete it. But in batch systems the processor knows how long the job is as it is queued. ● Sharing of batch system for multiple users. ● The batch systems can manage large repeated work easily. **<u>Disadvantages of batch processing systems</u>** ● Computer operators must be trained for using batch systems. ● It is difficult to debug batch systems. ● Batch systems are sometime costly. ● If some job takes too much time i.e. if error occurs in job then other jobs will wait for unknown time.	**(Any four relevant advantages and disadvantages: 2 marks each)**
	(ii)	**Explain major activities of memory management component of an operating system.**	**4M**
	Ans:	Main memory is a large array of words or bytes, ranging in size from hundreds of thousands to billions. Each word or byte has its own address. Main memory is a repository of quickly accessible data shared by the CPU and I/O devices. The central processor reads instructions from main memory during the instruction fetch cycle and both reads and writes data from main memory during the data fetch cycle. The main memory is generally the only large storage device that the CPU is able to address and access directly. For a program to be executed, it must be mapped to absolute addresses and loaded into memory. As the program executes, it accesses program instructions and data from memory by generating these absolute addresses. To improve both the utilization of the CPU and the speed of the computer's response to its users, operating system keeps several programs in memory. To handle memory with multiple programs memory management is necessary. Activities of memory management. 1) Keeping track of which part of memory are currently being used and by whom. 2) Deciding which processes and data to move into and out of memory. 3) Allocating & Deallocating space as needed.	**(Relevant Explanation: 4 marks)**

	(iii)	**Define the following with respect to resources.** **1) A preemptable resource** **2) A non-preemptable resource**	**4M**
	Ans:	1. **Preemptable resource:** These are the resources that can be taken away from its currently allocated process (owner) and be allocated to another process. An example is memory space. 2. **Non-preemptable resource:** These are the resources that cannot be taken away from its allocated process. An example is a non sharable printer. Memory is an example of a Preemptable resource. Consider, for example, a system with 32 MB of user memory, one printer, and two 32-MB processes that each want to print something. Process A request and gets the printer, then start to compute the values to print. Before it has finished with the computation, it exceeds its time quantum and is swapped out. Process B now runs and tires, unsuccessfully, to acquire the printer. Potentially, we now have a deadlock situation, because A has the printer and B has the memory, and neither can proceed without the resource held by the other. Fortunately, it is possible to preempt (take away) the memory from B by swapping it out and swapping A in. Now A can run, do its printing, and then release the printer. No deadlock occurs. A Non-preemptable resource, in contrast is one that cannot be taken away from its current owner without causing the computation to fail. If a process has begun to burn a CD-ROM, suddenly taking the CD recorder away from it and giving it to another process will result in a garbled CD. CD recorders are not Preemptable at an arbitrary moment.	**(Define preemptable and non-preemptable resource: 2 marks each)**
	(iv)	**List four types of UNIX files and draw Unix file system.**	
	Ans:	Types of Unix files: • Ordinary files • Directory files • Special or Device files • Fifo (Pipe) files	**(List: 2 marks, Any relevant Diagram: 2 marks)**

		Unix file system:  	
	b)	**Attempt any <u>ONE</u> of the following:**	**6 Marks**
	(i)	**Explain working of CPU switch from process to process with diagram.**	**6M**
	Ans:	A CPU switch from process to process is referred as context switch. A context switch is a mechanism that 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. When the scheduler switches the CPU from one process to another process, the context switch saves the contents of all process registers for the process being removed from the CPU, in its process control block. Context switch includes two operations such as state save and state restore. State save operation stores the current information of running process into its PCB. State restore operation restores the information of process to be executed from its PCB. Switching the CPU from one process to another process requires performing state save operation for the currently executing process (blocked) and a state restore operation for the process ready for execution. This task is known as context switch.	**(Explanation: 4 marks, Diagram: 2 marks)**

			
	(ii)	**Explain CPU and I/O burst cycle with the help of diagram.**	**6M**
	Ans:	CPU burst cycle: - It is a time period when process is busy with CPU. I/O burst cycle: - It is a time period when process is busy in working with I/O resources. A process execution consists of a cycle of CPU execution and I/O wait. A process starts its execution when CPU is assigned to it, so process execution begins with a CPU burst cycle. This is followed by an I/O burst cycle when a process is busy doing I/O operations. A process switch frequently from CPU burst cycle to I/O burst cycle and vice versa. The complete execution of a process starts with CPU burst cycle, followed by I/O burst cycle, then followed by another CPU burst cycle, then followed by another I/O burst cycle and so on. The final CPU burst cycle ends with a system request to terminate execution.	**(Explanation: 4 marks, Diagram:2 marks)**



5.		**Attempt any <u>TWO</u> of the following:**	**16Marks**
	a)	**Describe concept of file, its types and operations on file attributes in detail.**	**8M**
	Ans:	**Concept of file:** A file is a named collection of related information that is recorded on secondary storage such as magnetic disks, magnetic tapes and optical disks. The operating system abstracts from the physical properties of its storage devices to define a logical storage unit called as file. Files are mapped by the operating system onto the physical devices. Files represent programs and data. Data files may be numeric, alphabetic, alphanumeric or binary. A file is a sequence of bits, bytes, lines or records, the meaning of which is defined by its creator and user. Many different types of information may be stored in a file: source programs, object programs, executable programs, numeric data, text payroll records, graphic images, sound recording and so on. A file has a certain defined structure according to its type. **Different types of files**	**(Relevant description of file:2 marks, any 6 types:3 marks, any relevant 6 file operation: 3 marks)**

file type	usual extension	function
executable	exe,com,bin Or none	ready-to-run machine-language program
object	Obj,o	compiled, machine language, not linked
source code	c,cc,java,pas	Source code in various languages

	,asm,a	
batch	bat, sh	commands to the command interpreter
text	txt ,doc	textual data, documents
word processor	wp, tex, rtf, doc	various word-processor Formats
library	lib, a, so, dll	libraries of routines for programmers
print or view	ps, pdf,jpg	ASCII or binary file in a format for printing or viewing
archive	arc, zip, tar	related files grouped into one file, sometimes compressed, for archiving or storage
multimedia	mpeg,mov,rm,mp3,avi	binary file containing audio or A\V information

File Operations

Basic file operations are

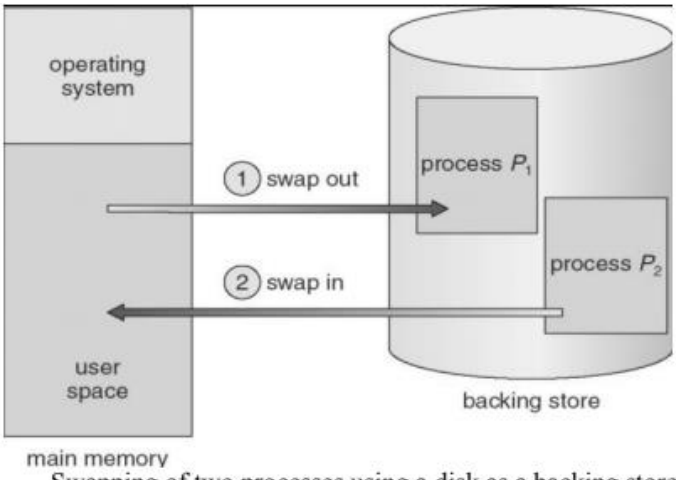
1. Creating a file. Two steps are necessary to create a file.
1. Space in the file system must be found for the file.
2. An entry for the new file must be made in the directory.

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

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

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

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

		6. Truncating a file. The user may want to erase the contents of a file but keep its attributes. Rather than forcing the user to delete the file and then recreate it, this function allows all attributes to remain unchanged (except for file length) but lets the file be reset to length zero and its file space released.	
	b)	**Explain swapping in operating system with diagram and example.**	**8M**
	Ans:	**Swapping**:A process must be in the main memory so that it can execute. Swapping is a memory/process management technique used by the operating system to increase the utilization of the processor. A process in execution may go into blocked state due to expiry of time quantum,occurance of interrupt,etc. when a process is in blocked state and next process is waiting for execution then operating system performs swapping. Swapping is a process of moving blocked process from the main memory to the backing store and new process from backing store to main memory. Swapping forms a queue of temporarily suspended process and the execution continues with the newly arrived process. **Diagram:**  Swapping of two processes using a disk as a backing store. In the above diagram, two processes P1 and P2 are shown. A process P1 is in main memory and in blocked state. Process P2 is in backing store waiting for its turn to execute.As P1 is blocked, operating system swap out this process by moving it from main memory to backing store and swap in process P2 by loading it from backing store to main memory. This process of swap out and swap in is called as swapping of processes. **Example:**Consider three processes P1,P2 and P3 are in memory. A Round-Robin CPU scheduling algorithm is in use. A process P1 starts its execution. When a time quantum expires memory manager swaps out the process P1 and swaps in process P2. If P1 requires more time than the time quantum then it is added to blocked queue and waits for its turn for execution. Once a time quantum of P2 expires ,manager swaps out it and swaps in P3. when time quantum of P3 expires, manager swaps out it and swaps in P1. This cycle of swap out and swap in continues till all the processes completes their execution.	**(Explanation of swapping: 3 marks; Diagram : 2 marks; Any Example: 3 marks)**

c) Comparison between Linux and UNIX. (Four points) — **8M**

(i) User interface
(ii) Name of provider
(iii) Processing speed
(iv) Security

Ans:

Parameter	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.	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, SolariAll 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.

(Each difference: 2 marks)

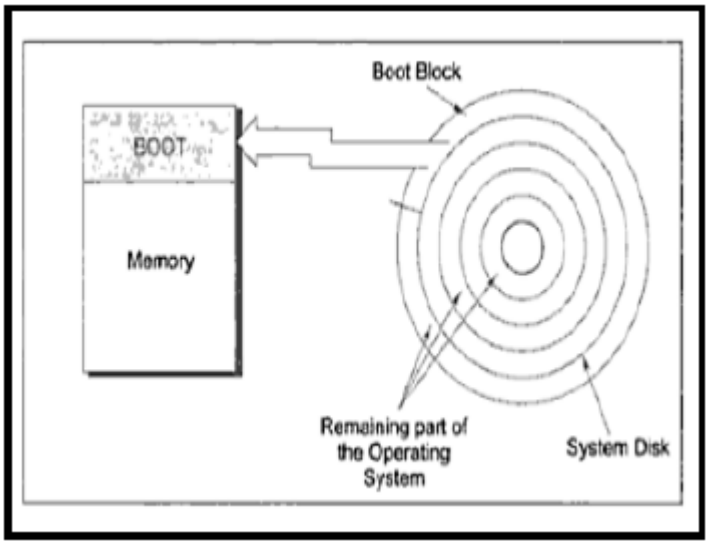
6. Attempt any <u>FOUR</u> of the following : — **16Marks**

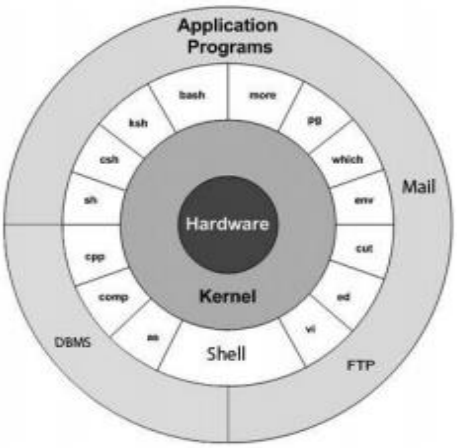
a) List characteristics of operating system for smooth functioning of a computer – system. (Eight points) — **4M**

Ans:

1. It provides user interface in the form of command line interface (CLI), batch interface and graphical user interface (GUI).
2. It supports program execution by loading the contents of program file into memory.
3. System provides I/O resources that may include files and I/O devices required by a running program.
4. It supports file-system manipulation. There are many operations that are performed for file creation, deletion, allocation, and naming.
5. It performs communications by message passing between systems that require 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.

(Explanation of any 8 characteristics: 4 marks (1/2 mark each))

		6. It supports 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. At the software level, media must be checked for data consistency. **7.** Computer system supports accounting of computers that keep track at which users use how much and what kind of computer resources. **8.** System allocates and deallocates resources to the process. When there are multiple users or multiple jobs running at the same time, resources must be allocated to each of them code. **9.** It provides protection and security for owners of information stored in multiuser or networked computer system may want to control use of the information in the form of password or access permissions.	
	b)	**With neat labelled diagram explain the working of Booting process.**	**4M**
	Ans:	The loading of the operating system is achieved by a special program called BOOT. Generally this program is stored in one (or two) sectors on the disk with a pre-determined address. This portion is normally called "BOOT Block" as shown in fig. The ROM normally contains a minimum program. When one turns the computer „ON", the control is transferred to this program automatically by the hardware itself. This program in ROM loads the BOOT program in pre-determined memory locations. The beauty is to keep BOOT program as small as possible, so that the hardware can manage to load it easily and in a very few instructions. This BOOT program in turn contains to read the rest of the Operating System into the memory. This is depicted in figures. The mechanism gives an impression of pulling oneself up. Therefore, the nomenclature bootstrapping or its short form booting.  	**(Explanation:2 marks, Diagram: 2 marks)**
	c)	**With neat labelled diagram explain Unix layered structure.**	**4M**
	Ans:	• Kernel: The kernel is the heart of the operating system. It interacts with the hardware and performs most of the tasks like memory management, task scheduling and file management. • Shell: The shell is the utility that processes your requests. When you type in a command at	**(Explanation:2 marks, Diagram: 2 marks)**

		your terminal, the shell interprets the command and calls the program that you want. The shell uses standard syntax for all commands. C Shell, Bourne Shell and Korn Shell are the most famous shells which are available with most of the Unix variants. • Commands and Utilities: There are various commands and utilities which you can make use of in your day to day activities. cp, mv, cat and grep, etc. are few examples of commands and utilities. There are over 250 standard commands plus numerous others provided through 3rd party software. All the commands come along with various options. Files and Directories − All the data of Unix is organized into files. All files are then organized into directories. These directories are further organized into a tree-like structure called the file system. • Application programs: These are the programs that provide interface to the users through which they can interact with the system.  	
	d)	**Explain the working of semaphores.**	**4M**
	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. Working of semaphore to solve synchronization problem:- Consider two concurrently running processes P1 and P2.P1 contains statement S1 and P2 contains statement S2.When we want to execute statement S2 only after execution of statement S1, 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 P1 and P2.	**(Explanation of working of semaphores : 4 marks)**

Process P1 contains:

S1;

signal (synch);

Process P2 contains:-

wait (synch);

S2;

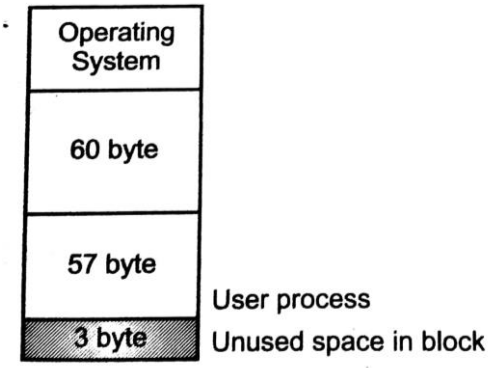
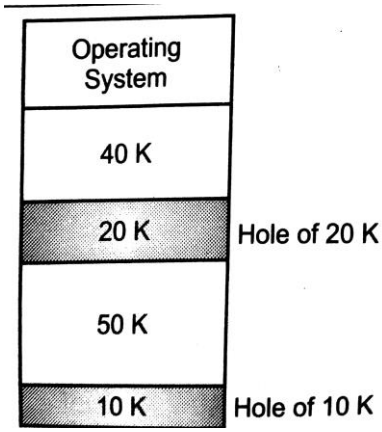
As synch is initialized to 0, Process P2 will wait and process P1 will execute. Once process P1 completes execution of statement S1, it performs signal () operation that increments synch value. Then wait () operation checks the incremented value and starts execution of statement S2 from Process P2.

e) Give difference between External fragmentation and Internal fragmentation (four points) — **4M**

Ans:

Internal fragmentation	External fragmentation
1. Internal fragmentation refers to unused space from space allocated to any process.	1. External fragmentation refers to unused space from the memory that is not allocated to any process.
2. Internal fragmentation occurs when memory block allocated to a process is bigger in size than the required size.	2. External fragmentation occurs when free memory block is available but is less than the size required by the processes waiting to be loaded in memory.
3. Example: In fixed memory partitioning technique all partitions are of same size. So the chances of occurrence of internal fragmentation are more. Consider all partitions are of 8K and the process loaded inside that partition is only 4K then internal fragmentation occurs.	3. Example: In variable partitioning technique each block is of variable size so some tomes even though free partition is available it cannot be allocated to the process that requires more space. So chances of occurrence of external fragmentation are more. Consider 20K and 10K partitions are available but space required by the waiting process is 30K. then external fragmentation occurs.

(Each Difference: 01 mark)

			4. Consider this is fixed partitioning.  60 bytes are allocated to each process but second process is of 57 byte so 3 byte internal fragmentation is shown.	4. Consider this is variable partitioning.  Two processes are already loaded in 40K and 50K partitions. Assume that one more process is waiting to get inside the memory with space requirement of 25K.20K and 10K is external fragmentation i.e. these two partitions are free but cannot be allocated to another process.		