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

SUMMER-15 EXAMINATION

Model Answer

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

Important Instructions to examiners:

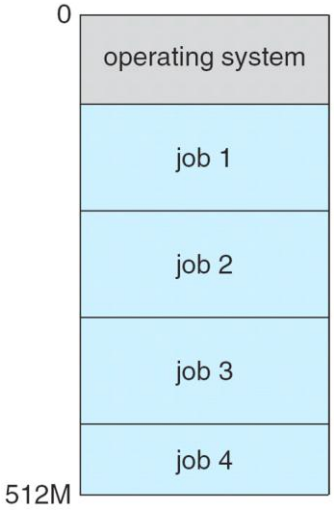
1) The answers should be examined by key words and not as word-to-word as given in The model answer scheme.
2) The model answer and the answer written by candidate may vary but the examiner may try to assess the understanding level of the candidate.
3) The language errors such as grammatical, spelling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

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

a) Explain time sharing operating system.

(Explanation- 4Marks)

Ans: The CPU is multiplexed among several jobs that are kept in memory and on disk i.e. the CPU is allocated to a job only if the job is in memory for specific time period. A job swapped in and out of memory to the disk. On-line communication between the user and the system is provided; when the operating system finishes the execution of one command, it seeks the next "control statement" from the user's keyboard. On-line system must be available for users to access data and code.



b) List system component. Explain any two.

(List - 1Mark, Description of any two – 1 ½ Marks each)

Ans: List of System Components:

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

Process Management

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

The 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 deallocate memory space as needed.

File Management

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

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

- 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.

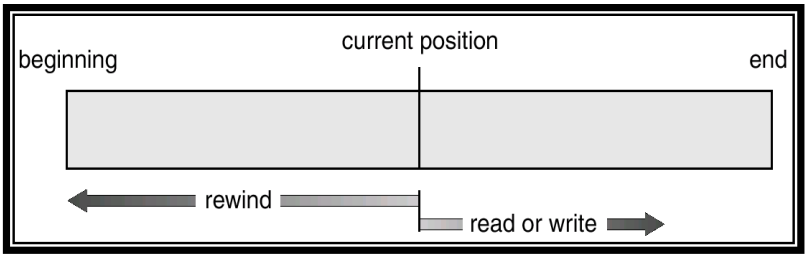
c) Describe the sequential file access method.
(Explanation-2 Marks, diagram-2 Marks)

Ans:

Most programming languages provide two different ways to access data stored in a file i.e. information. When it is used, this information must be accessed and read into computer memory. The information in the file can be accessed in several ways

Sequential Access Method:

- The simplest access method is sequential access. Information in the file is processed in order, one record after the other.
- This mode of access is by far the beginning current position most common; for example, editors and compilers usually access files in this fashion.
- Reads and writes make up the bulk of the operations on a file.
- A read operation read next reads the next portion of the file and automatically advances a file pointer, which tracks the I/O location.
- Similarly, the write operation write next appends to the end of the file and advances to the end of the newly written material (the new end of file).



If you want to read a piece of data that is stored at the very end of the file, you have to read all of the data that comes before it-you cannot jump directly to the desired data. This is similar to the way cassette tape players work. If you want to listen to the last song on a

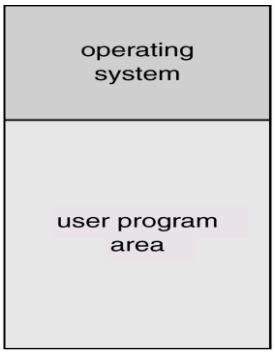
cassette tape, you have to either fast-forward over all of the songs that come before it or listen to them.

d) Explain batch operating system.
(Explanation – 3 Marks, diagram -1 Mark)

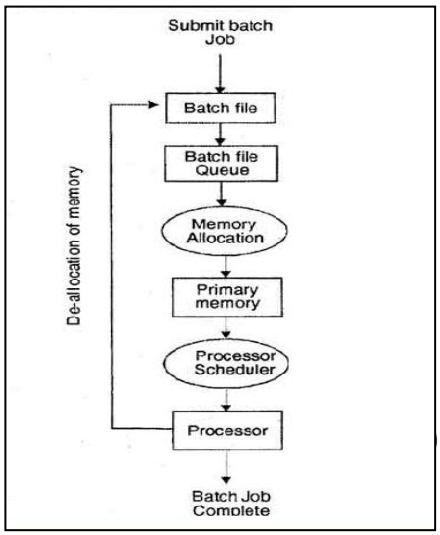
Ans: A batch operating system normally reads a stream of separate jobs (from a card reader. For example), each with its own control cards that predefine to prevent errors and improper use of the computer. It is concerned with the operation and control if I/O devices.

- A batch system is one in which jobs are bundled together with the instruction necessary to allow them to be processed without intervention. Often jobs of a similar nature can be bundled together to further increase economy.
- Common input devices were card readers and tape drives. The basic physical layout of the memory of batch job computer is shown in fig.
- The OS was simple, its major task was to transfer control from one job to the next. The job was submitted to the computer operator in form of punch cards. At some later time the output appeared.
- The OS was always resident in memory. Often magnetic tapes and drums were used to store intermediate data and compiled programs.

Example: Payroll system, stock control and billing systems.



OR



b) Attempt **any one** of the following**:** **Marks 6**

a) Explain LRU page replacement algorithm by taking suitable example.
(Explanation - 3Marks, Example - 3Marks)

Ans:

Least Recently Used (LRU) Algorithm:

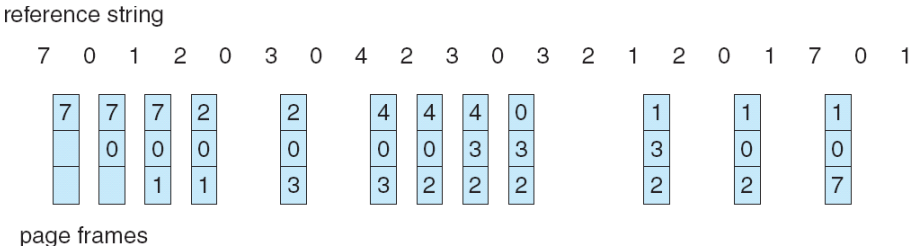
Counter implementation:-Every page entry has a counter; every time page is referenced through this entry; copy the clock into the counter.

When a page needs to be changed, look at the counters to determine which are to change.

Stack implementation – keep a stack of page numbers in a double link form:

LRU replacement associates with each page the time of that page's last use. When a page must be replaced, LRU chooses the page that has not been used for the longest period of time.

Example: - total available frames:-3



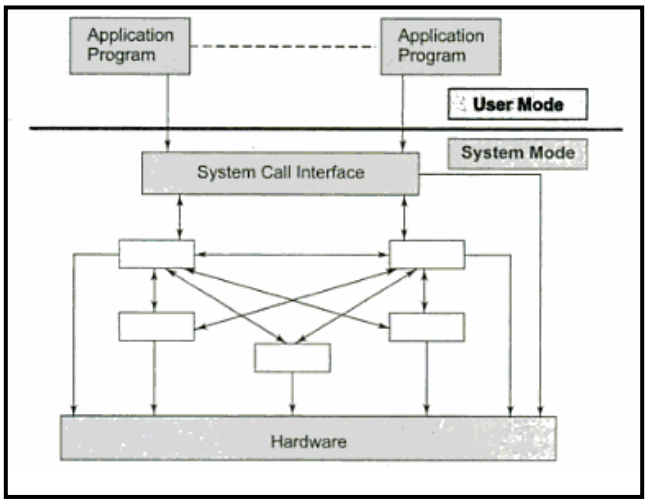
Initially, reference 7, 0, 1 are store inside the three free frames. When the reference to page 4 occurs, LRU replacement checks for existing pages inside the frames for their last access. From the three frames in the memory, page 7 was used least recently. Thus, the LRU algorithm replaces page 7, not knowing that page 7 is about to be used.

Advantage: Less number of faults as compared to FIFO

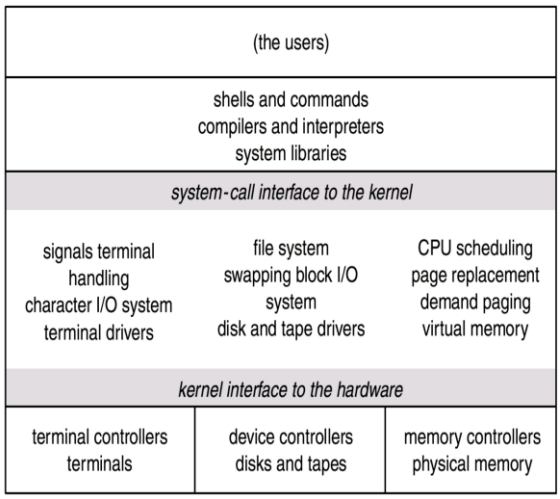
b) Draw and explain monolithic structure of operating system.
(Explanation- 3Marks, Diagram-3Marks)

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

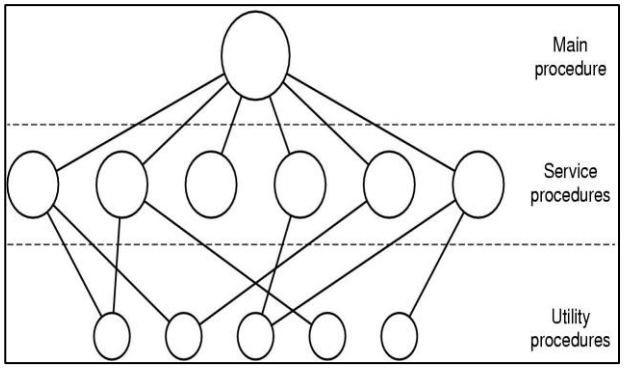
Simple structuring model for a monolithic system. (Any one diagram from the Following)



OR


(the users)
shells and commands
compilers and interpreters
system libraries
system-call interface to the kernel
signals terminal handling
character I/O system
terminal drivers
file system
swapping block I/O system
disk and tape drivers
CPU scheduling
page replacement
demand paging
virtual memory
kernel interface to the hardware
terminal controllers
terminals
device controllers
disks and tapes
memory controllers
physical memory

OR


Main procedure
Service procedures
Utility procedures

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

a) Compare: UNIX and LINUX w.r.t. following point: user interface, processing speed. ***(Each point carries-2Marks)***

Ans:

Point	LINUX	UNIX
User Interface	Linux typically provides two GUIs, KDE and Gnome. But there are millions of alternatives such as LXDE, Xfce, Unity, Mate, twm, ect.	Unix is a command based OS.
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 command is done so it takes less time as compared to Linux

b) Explain the working of clustered operating system. ***(Description- 4 Marks)***

Ans:

Clustered System: Cluster is a group of interconnected, whole computers working together as a unified computing source that can create the illusion of being one machine. Each computer in a cluster is typically referred to as a node. Clustering (means gather together) allows two or more system to share storage closely linked via a local area network. Asymmetric Cluster (at least two servers: One is on a standby mode while the other is monitoring the other one. If one stops other will work).Symmetric Cluster (all work at the same level: They work together and monitor each other).

Cluster – is collection of computer in which any member of the cluster is capable of supporting the processing function of any other member. A cluster has n+k configuration where n processing nodes are actively processing the application & k processing nodes are in a standby state, serving as a spares. 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.

Other common cluster configuration include Simplex (one active node, no spare), n+1 active node (n active nodes, 1 spare)

Clustered system can be implemented using LAN.This system is a subsystem of a telecommunication switching system, running certain centralized application function. A LAN is interconnecting hub that provides connectivity with each other in the switching system.

c) Explain static and dynamic memory partioning with advantages and drawback.
(Explanation- static & dynamic 1Mark each, any one Advantage and disadvantage 1 Mark each for each type)

Ans: Static Memory Partitioning

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

Operating System 8 M
8 M
8 M
8 M
8 M
8 M
8 M
8 M

Advantages:

- Simple to implement
- It requires minimal operating system software and processing overhead as partitions are fixed at the time of system generation.

Disadvantages: Memory wastage

- Inefficient use of memory due to internal fragmentation.
- Maximum number of active processes is fixed.

Dynamic Memory partitioning

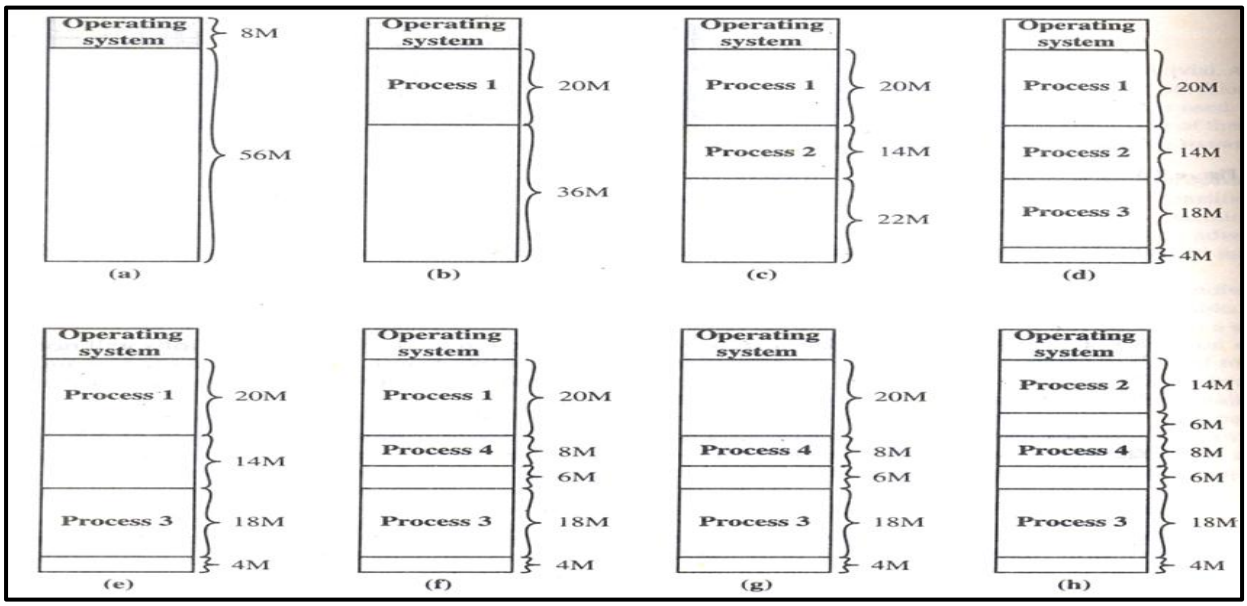
When a process enters in main memory, it is allocated exact size that is required by that process. So in this method, partitions can vary in size depending on memory space required by a process entering in main memory. Operating system maintains a table indicating which parts of memory are available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

partition in memory. Like this depending on size of process and available memory, partitions take place in main memory.

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

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

Process of memory allocation:-



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

Advantages:
No internal fragmentation,

More efficient use of main memory.

Disadvantages:
It suffers from external fragmentation,

It needs compaction.

d) Explain structure of UNIX operating system.
(Explanation – 2 Marks, Diagram -2 Marks)

Ans:

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

Amongst the functions performed by the kernel are:

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

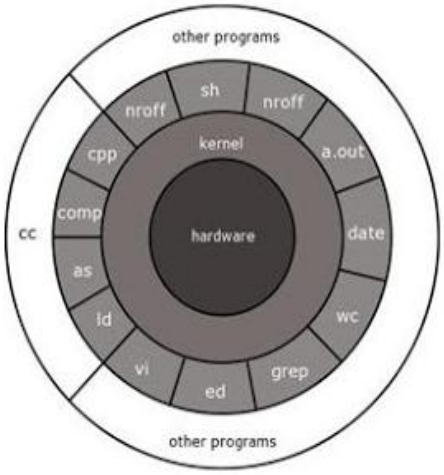
The shell:

The shell acts as an interface between the user and the kernel. When a user logs in, the login program checks the username and password, and then starts another program called the shell. The shell is a command line interpreter (CLI). It interprets the commands the user types in and arranges for them to be carried out. The commands are hemselvesprograms: when they terminate, the shell gives the user another prompt (% on our systems). The user can customise his/her own shell, and users can use different shells onthesamemachine.

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 previouscommands.

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.

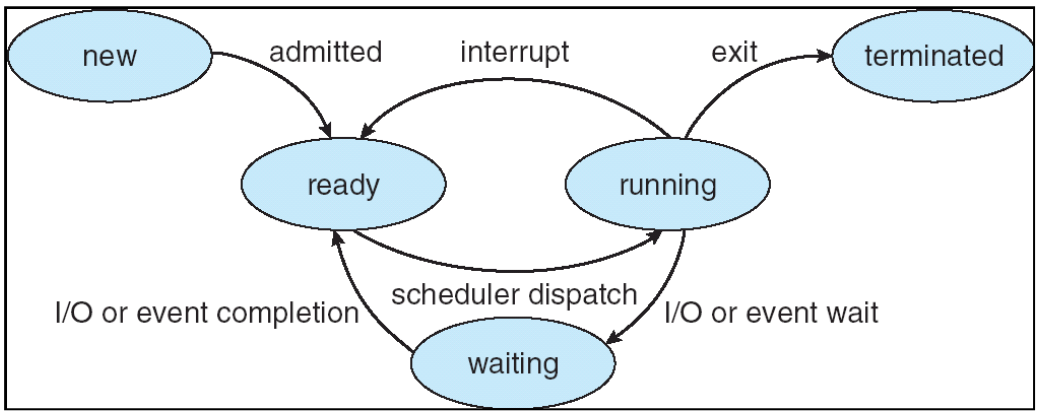
- **Bourne shell (sh)**
- **C shell (csh)**
- **TC shell (tcsh)**
- **Korn shell (ksh)**



(e) Define process. Explain process state in detail with the help of state diagram.
(Process definition-1Mark, Process state diagram- 2Marks, Description -1Mark)

Ans: **Definition:** Process is a program in execution.

Diagram:



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

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

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

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

New: The process being created is available in the new state. It is the new state because the system is not permitted it to enter the ready state due to limited memory

available in the ready queue. If some memory becomes available, then the process from the new state will go to ready state.

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

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

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

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

(f) Explain pre-emptive and non pre-emptive scheduling.
(Description of each - 2Marks)

Ans:

Preemptive Scheduling:-Even if CPU is allocated to one process, CPU can be preempted to other process if other process is having higher priority or some other fulfilling criteria.

- Throughput is less
- It is suitable for RTS.

- Only the processes having higher priority are scheduled.
- It doesn't treat all processes as equal.
- Algorithm design is complex.
  - Circumstances for preemptive
  - Process switch from running to ready state
  - Process switch from waiting to ready state

    For e.g.: Round Robin, Priority algorithms

Non Preemptive Scheduling

Once the CPU has been allocated to a process the process keeps the CPU until it releases CPU either by terminating or by switching to waiting state.

Throughput is high.

It is not suitable for RTS.

Processes having any priority can get scheduled.

It treats all process as equal.

Algorithm design is simple.

Circumstances for Non preemptive

- Process switches from running to waiting state
- Process terminates

For e.g.: FCFS algorithm

3. Attempt **any four** of the following: **Marks 16**

a) Describe any four secondary storage management activities.
(Any three activities-4Marks)

Ans: 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.

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

1.Managing the free space available on the secondary-storage device.

2.Allocation of storage space when new files have to be written.

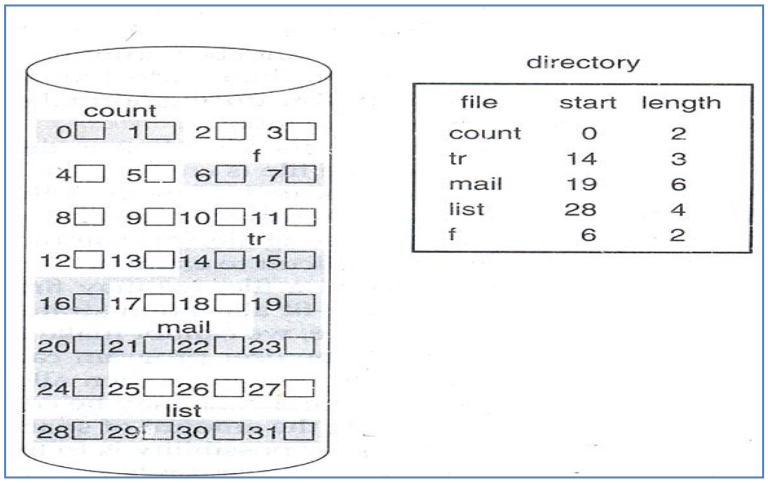
3.Scheduling the requests for memory access.

b) With suitable diagram explain contiguous allocation method.
(Explanation -2 Marks, Diagram-2 Marks)

Ans:

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.



c) Describe any four operations performed on process.
(Any two operations shall be considered, explanation of each operations- 2 Marks)

Ans:

Process creation

Create Process Operating system creates a new process with the specified or default attributes and identifier. A process may create several new 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 subprocess. So it forms a tree of processes. When operating system issues a CREATE system call, it obtains a new process control block from the pool of free memory, fills the fields with provided and default parameters, and insert the PCB into the ready list. Thus it makes the specified process eligible to run the process.

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

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

1. The parent continues to execute concurrently with its children.
2. The parent waits until some or all of its children have terminated.

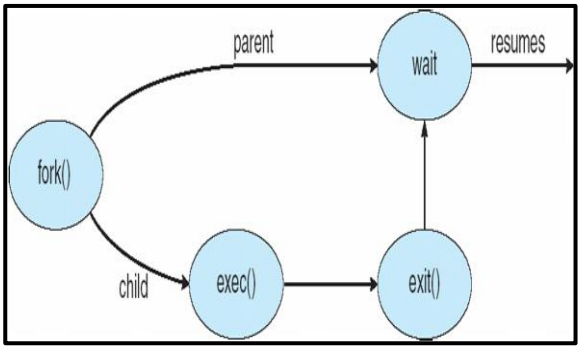
For address space, two possibilities occur:

1. The child process is a duplicate of the parent process.
2. The child process has a program loaded into it.

For example: UNIX examples

i) fork system call creates new process

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



Process Termination:

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

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

ii) Process' resources are deallocated by operating system

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

A parent may terminate the execution of one of its children for a variety of reason, such as these:

- The child has exceeded its usage of some of the resources that it has been allocated. (To determine whether this has occurred, the parent must have a mechanism to inpect the state of its children.)
- The task assigned to the child is no longer required.
- The parent is exiting, and the operating system does not allow a child to continue if tis parent terminates.

d) Define deadlock. What are necessary conditions for deadlock.
(Definition of deadlock-2Marks, four conditions-2Marks)

Ans:

Deadlock

A deadlock consists of a set of blocked processes, each holding a resource and waiting to acquire a resource held by another process in the set.

Deadlock can arise if four conditions hold simultaneously.

Deadlock Characterization

1.Mutual exclusion: only one process at a time can use non-sharable resource.

2. Hold and wait: a process is holding at least one resource and is waiting to acquire additional resources held by other processes.

3. No pre-emption: a resource can be released only voluntarily by the process holding it after that process completes its task

4. Circular wait: there exists a set $\{P_0, P_1, \ldots, P_0\}$ of waiting processes such that P_0 is waiting for a resource that is held by P_1, P_1 is waiting for a resource that is held by

P_2, …, P_{n-1} is waiting for a resource that is held by P_n, and P_n is waiting for a resource that is held by P_0

e) Write steps for Banker's algorithm to avoid deadlock. Also give one example. ***(steps for algorithm-2Marks,Any relevant example-2Marks)***

Ans:

Banker's Algorithm

This algorithm calculates resources allocated, required and available before allocating resources to any process to avoid deadlock. It contains two matrices on a dynamic basis. Matrix A contains resources allocated to different processes at a given time. Matrix B maintains the resources which are still required by different processes at the same time.

Algorithm F: Free resources

Step 1: When a process requests for a resource, the OS allocates it on a trial basis.

Step 2: After trial allocation, the OS updates all the matrices and vectors. This updation can be done by the OS in a separate work area in the memory.

Step 3: It compares F vector with each row of matrix B on a vector to vector basis.

Step 4: If F is smaller than each of the row in Matrix B i.e. even if all free resources are allocated o any process in Matrix B and not a single process can completes its task then OS concludes that the system is in unstable state.

Step 5: If F is greater than any row for a process in Matrix B the OS allocates all required resources for that process on a trial basis. It assumes that after completion of process, it will release all the recourses allocated to it. These resources can be added to the free vector.

Step 6: After execution of a process, it removes the row indicating executed process from both matrices.

Step 7: This algorithm will repeat the procedure step 3 for each process from the matrices and finds that all processes can complete execution without entering unsafe

state. For each request for any resource by a process OS goes through all these trials of imaginary allocation and updation. After this if the system remains in the safe state, and then changes can be made in actual matrices.

Example of Banker's Algorithm

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	

The content of the matrix *Need* is defined to be *Max – Allocation*

	Need
	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

Example: P_1 Request (1,0,2)

Check that Request ≤ Available (that is, (1,0,2) ≤ (3,3,2) ⇒ true

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

Executing safety algorithm shows that sequence < P_1, P_3, P_4, P_0, P_2> satisfies safety requirement

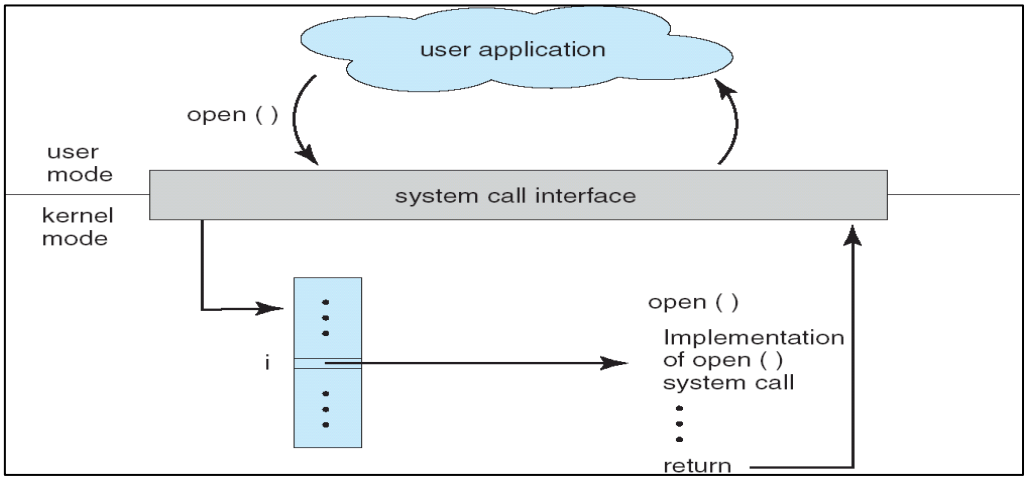
4. a) Attempt **any three** of the following: Marks 12

a) What is system call? Enlist any four system call.
(Define - 2 Marks, listing of any four System calls - 2 Marks)

Ans:

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

1. Each system call associated with a particular number.
2. System call interface maintains a table indexed according to these numbers.
3. The system call interface invokes intended system call in operating system kernel & returns status of the system call and any return values.
4. The caller needs to know nothing about how the system call is implemented. Just needs to obey API and understand what OS will do as a result call.
5. Most details of operating system interface hidden from programmers by API. It is managed by run-time support library.



System calls-

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

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

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

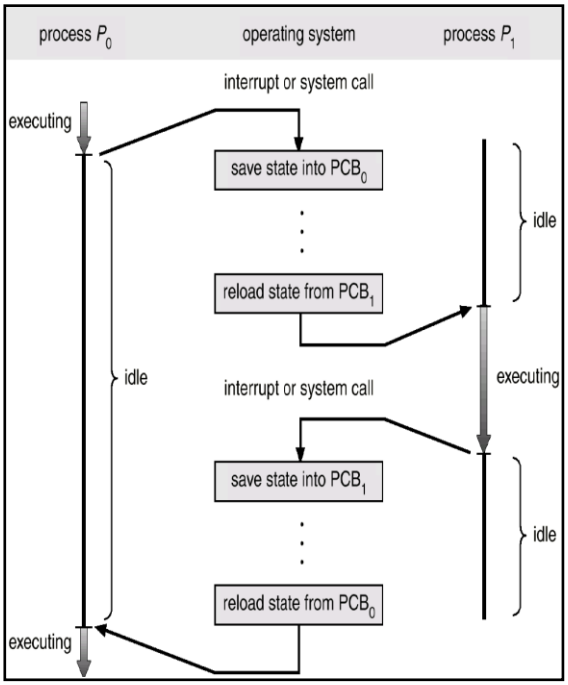
System calls Related to Information Maintenance: Get Time or Date, Set Time or Date Get System data, Set system data Get process, file or device attributes Set process, file or Device attributes.

b) Explain context switch with suitable example.
(Explanation-2Marks, diagram (example)-2Marks)

Ans:

Context switch

- Switch the CPU to another process requires saving the state of old process and loading the saved state for new process. This task is known as a context switch. The context switch represented with PCB.
- Saves context of old process in its PCB and loads context of new process schedule to run.



c) Describe any four services provided by an operating system.
(Explanation of any four services - 1 Mark each)

Ans:

OS services provided to the user:-

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 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 programmer's abilities to use the system efficiently.

OS services provided to the system:-

1. **Resource allocation**: - When there are multiple users or multiple processes running at the same time, resources must be allocated to each of them. Operating system manages resource allocation to the processes. These resources are CPU, main memory, file storage and I/O devices. For maximizing use of CPU, operating system does CPU scheduling. Operating system contains routines to allocate printers, modems, USB storage drives and other peripheral devices.

2. **Accounting**: - Operating system keeps track of usages of various computer resources allocated to users. This accounting is used for reconfiguration of system to improve computing services.

3. **Protection & security**:-Owners of information stored in a multiuser or networked computer system want to control use of that information. When several separate processes execute concurrently, one process should not interfere with the other processes or operating system itself. Protection provides controlled access to system resources. Security is provided by user authentication such as password for accessing information.

d) State and describe types of scheduler.
(State types - 1 Mark, Description of three types - 1 Mark each)

Ans:

Schedulers are of three types:-

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

Long Term Scheduler

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

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

Short Term Scheduler

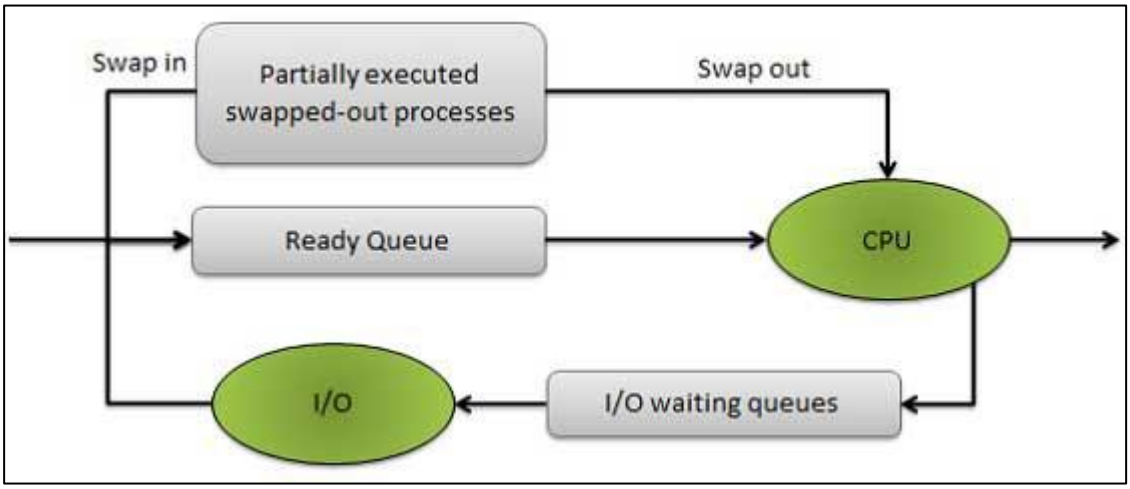
It is also called CPU scheduler. Main objective is increasing system performance in accordance with the chosen set of criteria. It is the change of ready state to running state of

the process. CPU scheduler selects process among the processes that are ready to execute and allocates CPU to one of them.

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

Medium Term Scheduler

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



Running process may become suspended if it makes an I/O request. Suspended processes cannot make any progress towards completion. In this condition, to remove the process from memory and make space for other process, the suspended process is moved to the secondary storage. This process is called swapping, and the process is said to be swapped out or rolled out. Swapping may be necessary to improve the process mix.

b) Attempt **any one** of the following: Marks 6

a) Describe many to one and one to one multithreading model with diagram.
***(Explanation & diagram-3Marks* for each model)**

Ans: 1) Many to One model

This model maps many user level threads to one kernel level thread. Thread management is done by thread library in user space.

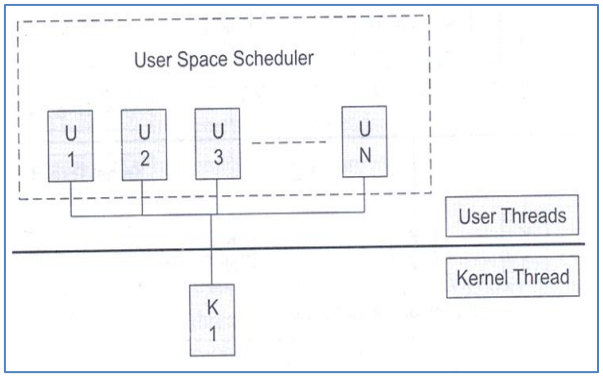
Advantages:-

- It is an efficient model as threads are managed by thread library in user space.

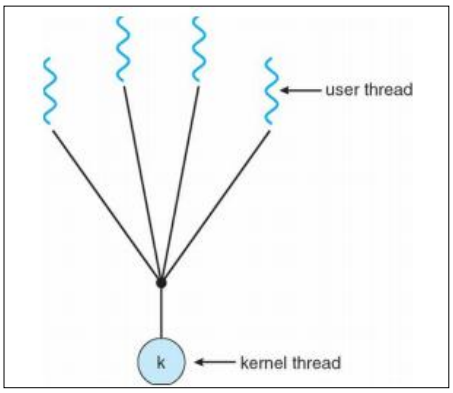
Disadvantages:-

- Only one thread can access the kernel at a time, so multiple threads are unable to run in parallel on microprocessor.
- If a thread makes a blocking system call then the entire process will be block.

Example: - Green threads – a thread library available for Solaris use many-to-one model.



OR



2) One to One model

It maps each user thread to a kernel thread. Even a thread makes a blocking call; other thread can run with the kernel thread.

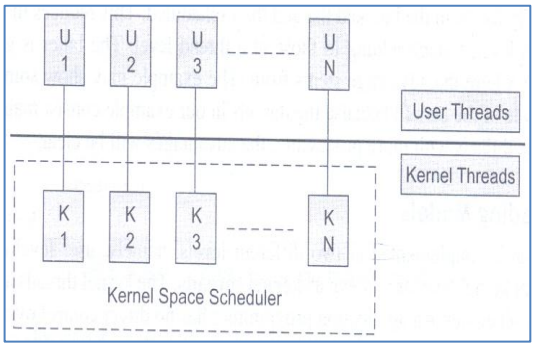
Advantages:-

- It allows multiple threads to run in parallel on multiprocessors.

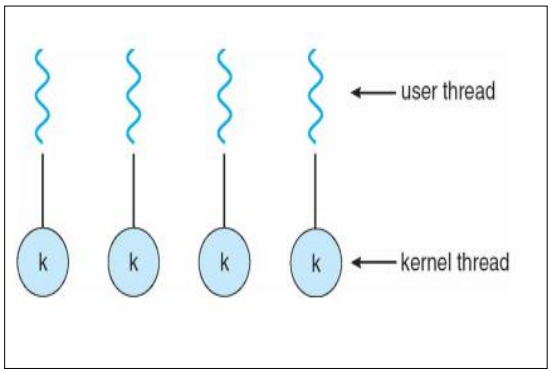
Disadvantages:-

- Creating a user thread requires creating the corresponding kernel thread. Creating kernel thread may affect the performance of an application.

Example: - Linux, Windows OS including Win 95, 98, NT 2000, and XP implement the one-to-one model.



OR



b) Explain concept of page replacement with suitable diagram.
(Explanation-3Marks, Diagram- 3Marks)

[Note:-explanation with any algorithm such as FIFO,Optimal,LRU shall be considered]

Ans:

Basic Page Replacement:

Page replacement takes the following approach. If no frame is free, we find one that is not currently being used and free it. We can free a frame by writing its contents to swap space and changing the page table (and all other tables) to indicate that the page is no longer in memory. We can now use the free frame to hold the page for which the process faulted. We modify the page-fault service routine to include page replacement:

1. Find the location of the desired page on the disk.
2. Find a free frame:
   a. If there is free frame, use it.
   b. If there is no free frame, use a page-replacement algorithm to select a victim frame.
   c. Write the victim frame to the dis; change the page and frame tables accordingly.
3. Read the desired page into the newly freed frame; change the page and frame tables.
4. Restart the user process.

Notice that, if no frame are free, two page transfers (one out and one in) are required. This situation effectively doubles the page-fault service time and increases the effective access time accordingly.

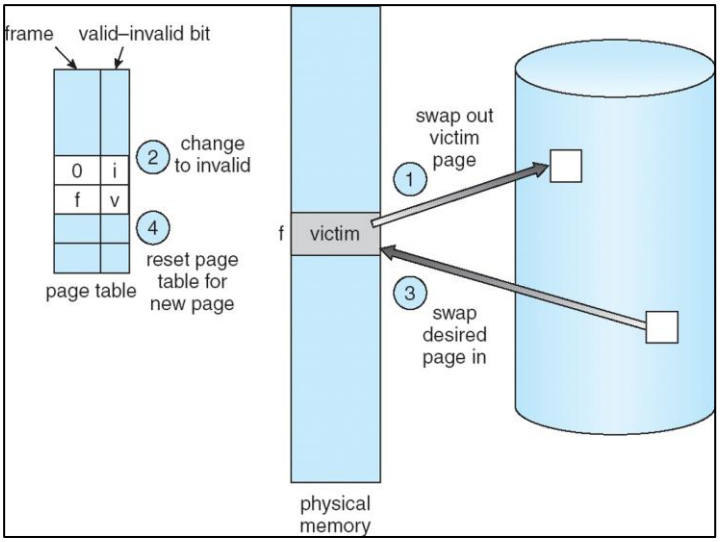


Figure Page replacement

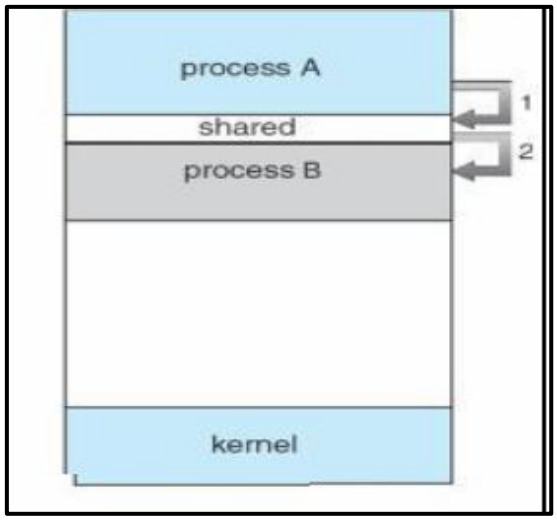
5. Attempt **any two** of the following**:** **Marks 16**

a) Explain Interprocess communication models with diagram.
(Shared memory Diagram- 2 Marks, Explanation- 2 Marks, Message Passing Diagram - 2 Marks, Explanation- 2 Marks)

Ans:

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

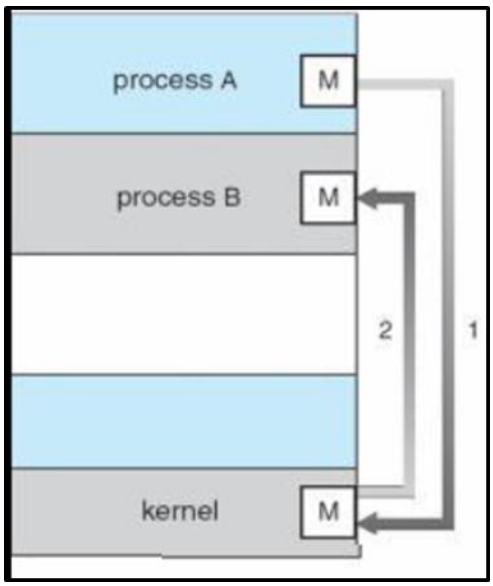
a. Shared memory:



In this a region of the memory residing in an address space of a process creating a shared memory segment can be accessed by all processes who want to communicate with other processes. All the processes using the shared memory segment should attach to the address space of the shared memory. All the processes can exchange information by reading and/or writing data in shared memory segment. The form of data and location are determined by these processes who want to communicate with each other. These processes are not under the control of the operating system. The processes are also responsible for ensuring that they are not writing to the same location simultaneously. After establishing shared memory segment, all accesses to the shared memory segment are treated as routine memory access and without assistance of kernel.

b. 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.



b) Explain SJF algorithm with example. Also calculate average waiting time.
(Explanation- 2 Marks, example- 4 Marks, Average waiting time- 2 Marks)
[Note- Any relevant example shall be considered]

Ans:

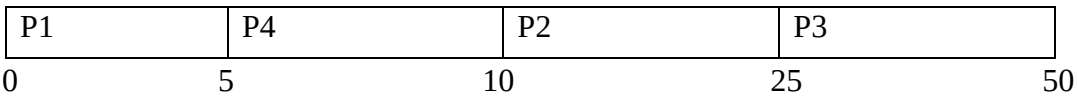
Explanation:-

In this algorithm, each process is associated with the length of the process's next CPU burst. When the CPU is available, it is assigned to the process that has smallest next CPU burst. When the next CPU bursts of two processes are same, then FCFS scheduling is used to select one process.

Example:-

Process	Burst time
P1	**5**
P 2	**15**
P 3	**25**
P 4	**5**

Gantt chart:



Waiting time for each process:

P1 =0;

P2 =10;

P3 =25 ;

P4 =05

Average waiting time = (0+10+25+05)/4

= 40/4 = 10 ms

c) Explain multilevel queue scheduling with example.
(Explanation - 4 Marks, Diagram or Example – 4 Marks)

Ans:

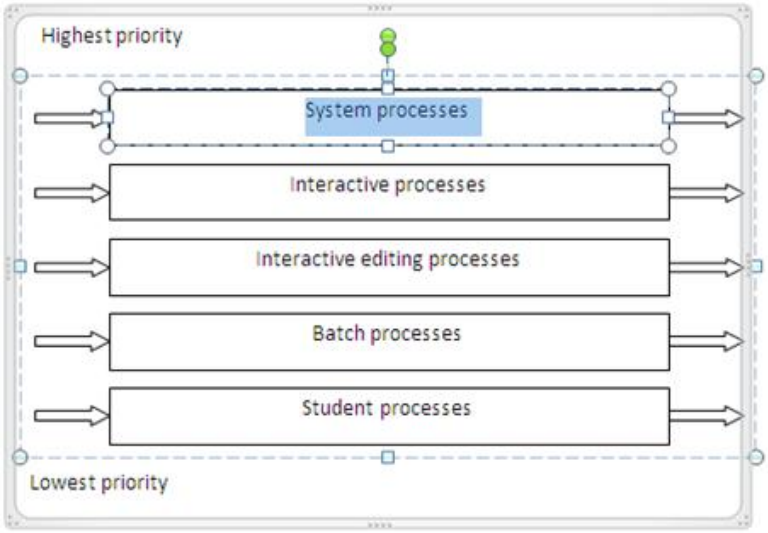
Multilevel queue scheduling

This algorithm partitions the ready queue into several separate queues. The processes are permanently assigned to one queue based on some property of the process. Each queue has its own scheduling algorithm. In addition, there must be scheduling between the queues which is mostly implemented as fixed priority preemptive scheduling.

- Whenever, the CPU becomes idle, the OS must select one of the processes in the ready-queue to be executed.
- The selection process is carried out the short-term scheduler or CPU scheduler. The CPU scheduler selects a process from the ready queue and allocates the CPU to that process.

Lets take an example of multilevel queue scheduling algorithm with five queues listed below in order of priority.

- System process
- Interactive processes
- Interactive editing processes
- Batch processes
- Student processes



Each queue has absolute priority over lower priority queues. No process in the batch queue, for example could run unless the queues for system processes, interactive processes and interactive editing processes were all empty. If an interactive editing process entered the ready queue while a batch process was running, the batch process could be preempted.

6. Attempt **any four** of following: **Marks 16**

a) Describe file management. Enlist the system call for file management.
(Explanation of File Management-2 Marks, any four system call- ½ Marks each)

Ans:

File Management A file is a collected of related information defined by its creator. Computer can store files on the disk (secondary storage), which provide long term storage. Some examples of storage media are magnetic tape, magnetic disk and optical disk. Each of these media has its own properties like speed, capacity, data transfer rate and access

methods. A file systems normally organized into directories to ease their use. These directories may contain files and other directions.

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

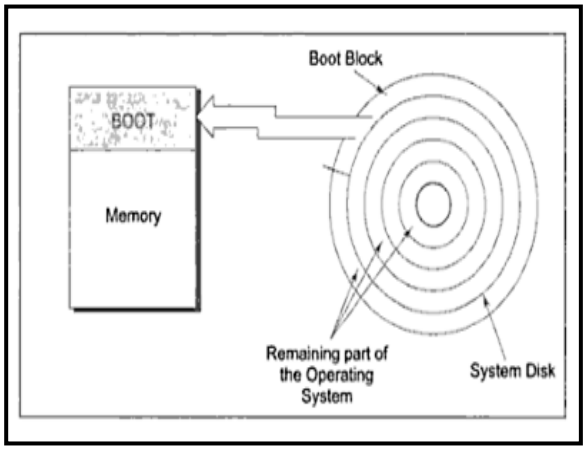
1. The creation and deletion of files.
2. The creation and deletion of directions.
3. The support of primitives for manipulating files and directions.
4. The mapping of files onto secondary storage.
5. The backup of files on stable storage media.

System calls Related to File management:

- Create file
- Delete file
- Open file
- Close file
- Create directory
- Read, Write, Reposition file
- Get file attribute
- Set file attribute
- Create a link
- Change the working directory

b) Describe stepwise booting process of UNIX along with diagram.
(Explanation - 3 Marks, Diagram - 1 Mark)

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



c) Explain generation of operating system.
(Explanation of four generations- 1 Mark each)

Ans: Generations of operating system

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

First generation 1945 – 1955 - vacuum tubes, plug boards

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

The 1950's - Second Generation

Second generation 1955 – 1965 - transistors, batch systems.

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

The 1960's - Third Generation

Third generation 1965 – 1980 - ICs and multiprogramming.

The systems of the 1960's were also batch processing systems, but they were able to take better advantage of the computer's resources by running several jobs at once. So operating systems designers developed the concept of multiprogramming in which several jobs are in

main memory at once; a processor is switched from job to job as needed to keep several jobs advancing while keeping the peripheral devices in use.

The Fourth Generation

Fourth generation 1980 – present personal computers

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

d) State and explain any four file attributes.
(Any four attributes- 1 Mark each)

Ans:

File Attributes

When a file is named, it becomes independent of the process, the user, and even the system that created it.

For instance, one user might create the file example.c, and another user might edit that file by specifying its name.

A file's attributes vary from one OS to another but typically consist of these:

- **Name**. It is a string of characters which is in human readable form.
- **Identifier**. This unique tag, usually a number, identifies the file within the file system; it is the non-human-readable name for the file.
- **Type**. This is the information used by the system to support different types of the files.
- **Location**. This information is a pointer to a device and to the location of the file on that device.

- **Size**. The current size of the file (in bytes, words, or blocks) and possibly the maximum allowed size are included in this attribute.
- **Protection**. Access-control information determines who can do reading, writing, executing, and so on.
- **Time, date, and user identification**. This information may be kept for creation, last modification, and last use.

The information about all files is kept in the directory structure, which also resides on secondary storage. Typically, a directory entry consists of the file's name and its unique identifier.

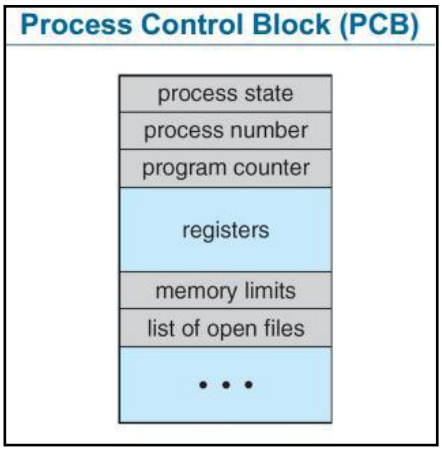
Attribute	Meaning
Protection	Who can access the file and in what way
Password	Password needed to access the file
Creator	ID of the person who created the file
Owner	Current owner
Read-only flag	0 for read/write; 1 for read only
Hidden flag	0 for normal; 1 for do not display in listings
System flag	0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

e) Draw and explain process control block in detail.

(Explanation- 2 Marks, Diagram – 2 Marks)

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

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



In general, a PCB may contain information regarding:

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

necessary to allow the process to be continued correctly after the completion of an interrupted process.

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