



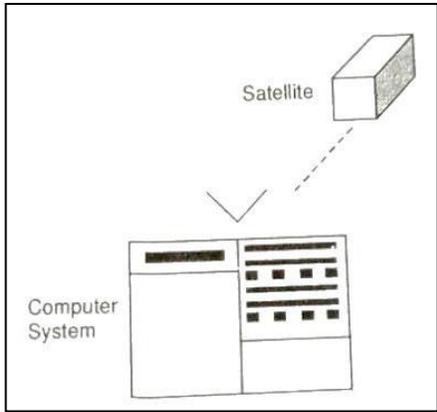
MODEL ANSWER
SUMMER- 17 EXAMINATION

Subject Title: Operating System

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.	Answer	Marking Scheme
1.	a)	Attempt any THREE of the following:	12 Marks
	(i)	With neat diagram, explain real time system. List its any four application.	4 M
	Ans:	<p>A real time system has well defined fixed time constraints. Processing should be done within the Defined constraints. Types of real time system Hard and Soft real time system.</p> <div style="text-align: center;">  </div> <p>Satellite Application of real time OS: Satellite application of real time OS The satellite connected to the computer system sends the digital samples at the rate of 1000 samples per second. The computer system has an application program that stores these samples</p>	<p>(Diagram: 1 mark, Explanation: 1 mark, any four Applications: ½ each)</p>



MODEL ANSWER
SUMMER- 17 EXAMINATION

Subject Title: Operating System

Subject Code: 17512

	<p>in a file. The sample sent by the satellite arrives every millisecond to the application. So computer must store or respond the sample in less than 1 millisecond. If the computer does not respond to the sample within this time, the sample will lost.</p> <p>Applications:</p> <ol style="list-style-type: none"> 1. Flight Control System 2. Simulations 3. Industrial control 4. Military applications 	
(ii)	Sate any four types of system call provided by an operating system.	4M
Ans:	<p>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.</p> <p>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.</p> <p>System calls Related to Device Management: Request a device, Release a device Read, Write, Reposition, Get device attribute, Set device attribute</p> <p>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.</p> <p>System calls Related to Communication: create,delete communication connection send, receive messages, transfer status information, attach or detach remote devices</p>	(Any four types of System calls: 1 mark each)
(iii)	Explain Bit map free-space management technique.	4M
Ans:	<p>Frequently, the free-space list is implemented as a bit map or bit vector. Each block is represented by a 1 bit. If the block is free, the bit is 0; if the block is allocated, the bit is 1. For example, consider a disk where blocks 2, 3, 4, 5, 8, 9, 10, 11, 12, 13, 17, 18, 25, 26, and 27 are free, and the rest of the blocks are allocated. The free-space bit map would</p> <p style="text-align: right;">be:</p> <p>11000011000000111001111110001111...</p> <p>The main advantage of this approach is that it is relatively simple and efficient to find n consecutive free blocks on the disk. Unfortunately, bit vectors are inefficient unless the entire vector is kept in memory for most accesses. Keeping it main memory is possible for smaller disks such as on microcomputers, but not for larger ones.</p>	(Explanation : 4 marks)



MODEL ANSWER
SUMMER- 17 EXAMINATION

Subject Title: Operating System

Subject Code: 17512

(iv)	Describe first generation of operating system with its advantages and disadvantages.	4M
Ans:	<p>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).</p> <p>Advantages:</p> <ul style="list-style-type: none">• That was the only electronic during those old days.• Those computers were very fast to calculate.• Vacuum tube technology made possible the advent of electronic digital computers.• Use of vacuum tube technology made possible to make electronic digital computer.• These computers could calculate in milliseconds. <p>Disadvantages:</p> <ul style="list-style-type: none">• The computers were very larger in size• They consumed a large amount of energy.• They heated very soon due to thousands of vacuum tubes.• They were not very reliable.• Air conditioning is required.• Constant maintenance was required.• Not-portable.• Costly commercial production.• Very slow speed.• Limited programming capabilities.• Used machine language only.• Used punch card for input.• Not versatile and less accurate.	(Explanation : 2 marks, any two Advantages: 1/2 mark each, any two Disadvantages: 1/2 mark each)
b)	Attempt any <u>ONE</u> of the following	6 Marks
(i)	Differentiate between paging and segmentation. (any six points)	6M



MODEL ANSWER
SUMMER- 17 EXAMINATION

Subject Title: Operating System

Subject Code: 17512

	<p>Ans:</p>	<table border="1"> <thead> <tr> <th data-bbox="219 294 787 346">Paging</th> <th data-bbox="787 294 1377 346">Segmentation</th> </tr> </thead> <tbody> <tr> <td data-bbox="219 346 787 493">Paging divides the computer's primary memory into fixed-size units called page frames, and the program's address space into pages of the same size.</td> <td data-bbox="787 346 1377 493">Segmentation is the only memory management technique that does not provide the user's program with a 'linear and contiguous address space.'.</td> </tr> <tr> <td data-bbox="219 493 787 640">The hardware memory management unit maps pages to frames.</td> <td data-bbox="787 493 1377 640">Segments are areas of memory that usually correspond to a logical grouping of information such as a code procedure or a data array.</td> </tr> <tr> <td data-bbox="219 640 787 829">The physical memory can be allocated on a page basis while the address space appears contiguous.</td> <td data-bbox="787 640 1377 829">Segments require hardware support in the form of a segment table which usually contains the physical address of the segment in memory, its size, and other data such as access protection bits and status.</td> </tr> <tr> <td data-bbox="219 829 787 903">Pages are used for swapping or managing memory.</td> <td data-bbox="787 829 1377 903">Small pieces called segments are used for memory management.</td> </tr> <tr> <td data-bbox="219 903 787 976">Page is indicated by its number and offset.</td> <td data-bbox="787 903 1377 976">Segment is indicated by segment number and its offset</td> </tr> <tr> <td data-bbox="219 976 787 1018">Page table is formed</td> <td data-bbox="787 976 1377 1018">Segment table is formed.</td> </tr> <tr> <td data-bbox="219 1018 787 1081">Do not support user's view of memory.</td> <td data-bbox="787 1018 1377 1081">Supports user's view of memory.</td> </tr> </tbody> </table>	Paging	Segmentation	Paging divides the computer's primary memory into fixed-size units called page frames, and the program's address space into pages of the same size.	Segmentation is the only memory management technique that does not provide the user's program with a 'linear and contiguous address space.'.	The hardware memory management unit maps pages to frames.	Segments are areas of memory that usually correspond to a logical grouping of information such as a code procedure or a data array.	The physical memory can be allocated on a page basis while the address space appears contiguous.	Segments require hardware support in the form of a segment table which usually contains the physical address of the segment in memory, its size, and other data such as access protection bits and status.	Pages are used for swapping or managing memory.	Small pieces called segments are used for memory management.	Page is indicated by its number and offset.	Segment is indicated by segment number and its offset	Page table is formed	Segment table is formed.	Do not support user's view of memory.	Supports user's view of memory.	<p>(Any six points: 1 mark each (any other valid point shall be considered))</p>
Paging	Segmentation																		
Paging divides the computer's primary memory into fixed-size units called page frames, and the program's address space into pages of the same size.	Segmentation is the only memory management technique that does not provide the user's program with a 'linear and contiguous address space.'.																		
The hardware memory management unit maps pages to frames.	Segments are areas of memory that usually correspond to a logical grouping of information such as a code procedure or a data array.																		
The physical memory can be allocated on a page basis while the address space appears contiguous.	Segments require hardware support in the form of a segment table which usually contains the physical address of the segment in memory, its size, and other data such as access protection bits and status.																		
Pages are used for swapping or managing memory.	Small pieces called segments are used for memory management.																		
Page is indicated by its number and offset.	Segment is indicated by segment number and its offset																		
Page table is formed	Segment table is formed.																		
Do not support user's view of memory.	Supports user's view of memory.																		
	<p>(ii)</p>	<p>State and describe services provided by an operating system.</p>	<p>6M</p>																
	<p>Ans:</p>	<p>Operating Services:</p> <ol style="list-style-type: none"> 1. Program execution 2. I/O operations 3. File-system manipulation 4. Communications 5. Error detection 6. Accounting <p>1. Program execution: The operating system loads the contents (or sections) of a file into memory and begins its execution. A user-level program could not be trusted to properly allocate CPU time.</p> <p>2. I/O operations: Disks, tapes, serial lines, and other devices must be communicated with at a very low level. The user need only specify the device and the operation to perform on it, while the system converts that request into device- or controller-specific commands. User-level programs cannot be trusted to access only devices they should have access to and to access them only when they are otherwise unused.</p> <p>3. File-system manipulation: There are many details in file creation, deletion, allocation, and naming that users should not have to perform. Blocks of disk space are used by files and must be tracked. Deleting a file requires removing the name file information and freeing the allocated blocks. Protections must also be checked to assure proper file access. User programs could neither ensure adherence to protection methods nor be trusted to allocate only free blocks and deallocate blocks on file deletion.</p> <p>4. Communications: Message passing between systems requires messages to be turned into packets of information, sent to the network controller, transmitted across a</p>	<p>(Any six services : 1 mark each)</p>																



MODEL ANSWER
SUMMER- 17 EXAMINATION

Subject Title: Operating System

Subject Code: 17512

communications medium, and **reassembled** by the destination system. Packet ordering and data correction must take place. Again, user programs might not coordinate access to the network device, or they might receive packets destined for other processes.

5. Error detection: Error detection occurs at both the hardware and software levels. At the hardware level, all data transfers must be inspected to ensure that data have not been corrupted in transit. All data on media must be checked to be sure they have not changed since they were written to the media. At the software level, media must be checked for data consistency; for instance, whether the number of allocated and unallocated blocks of storage match the total number on the device. There, errors are frequently process-independent (for instance, the corruption of data on a disk), so there must be a global program (the operating system) that handles all types of errors. Also, by having errors processed by the operating system, processes need not contain code to catch and correct all the errors possible on a system.

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

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

16 Marks

a) With neat diagram, explain structure of unix, operating system.

4M

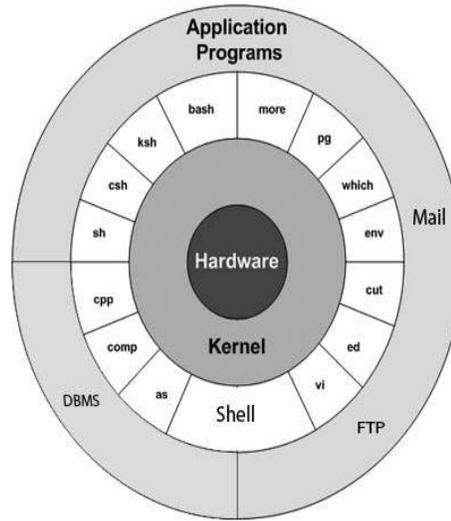
Ans:

- **Kernel:** The kernel is the heart of the operating system. It interacts with the hardware and 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 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.

(Diagram: 2 marks,
Explanations : 2 marks)



- **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**.

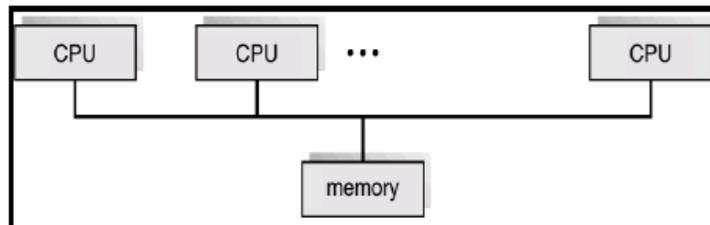


b) Explain multiprocessor system and its two types.

4M

Ans: Multiprocessor Systems:

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



Symmetric multiprocessing (SMP):

- Each processor runs an identical copy of the operating system.
- Many processes can run at once without performance deterioration.

(Explanation of multiprocessor system : 2 marks, Explanation of each type 1 mark)



- Most modern operating systems support SMP
- Asymmetric multiprocessing:**
- Each processor is assigned a specific task; master processor schedules and allocated work to slave processors.
 - More common in extremely large systems.

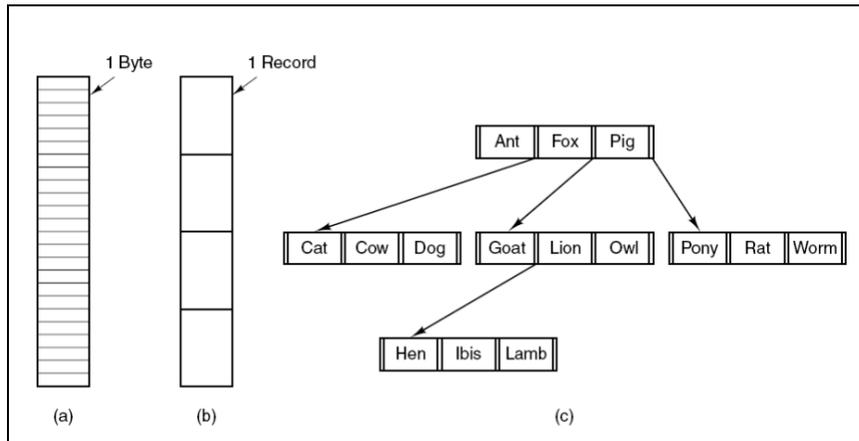
c) Explain file structure with example.

4M

Ans: Files can be structured in any of several ways. Three common possibilities are depicted in fig. The file in fig is an unstructured sequence of bytes. In this model, a file is a sequence of fixed-length records, each with some internal structure. The third kind of file structure is shown in fig. In this organization, a file consists of a tree of records, not necessarily all the same length, each containing a key field in a fixed position in the record.

(Explanation : 2 marks, Example: 2 marks)

FILE STRUCTURE



Three Kinds of files (a) Byte Sequence . (b) Record Sequence. (c) Tree

d) Describe stepwise booting process of unix along with diagram.

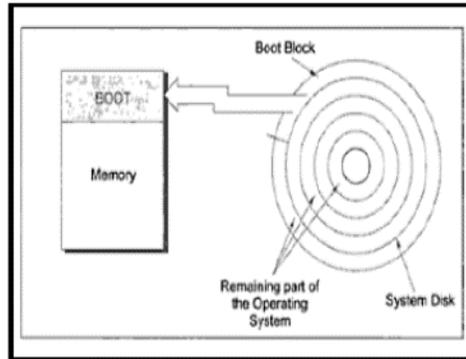
4M

Ans: The loading of the operating system is done with the help of a special program called BOOT. This program is stored in one or two sectors on the disk with a pre-determined address. This portion is called as BOOT Block. The ROM contains minimum program called as bootstrap program. When the computer is turn ON, the control is transferred to this program automatically by the hardware itself. This program in ROM locates the BOOT program and loads it into predetermined memory locations. This BOOT program

(Explanation : 2 marks, Diagram: 2 marks)



loads the operating System into the memory.



e) Describe the following:

- (i) Schedulers
- (ii) Context switch

4M

Ans: (i) **Schedulers:** Schedulers are special system software's which handles process scheduling in various ways. A process migrates between the various scheduling queues throughout its life time. The operating system must select, for scheduling purposes, processes from these queues in some fashion. The selection process is carried out by the appropriate scheduler. Scheduler is the system program which schedules processes from the scheduling queues. Their main task is to select the jobs to be submitted into the system and to decide which process to run.

Schedulers are of three types:

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

(ii) **Context switch:**

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

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

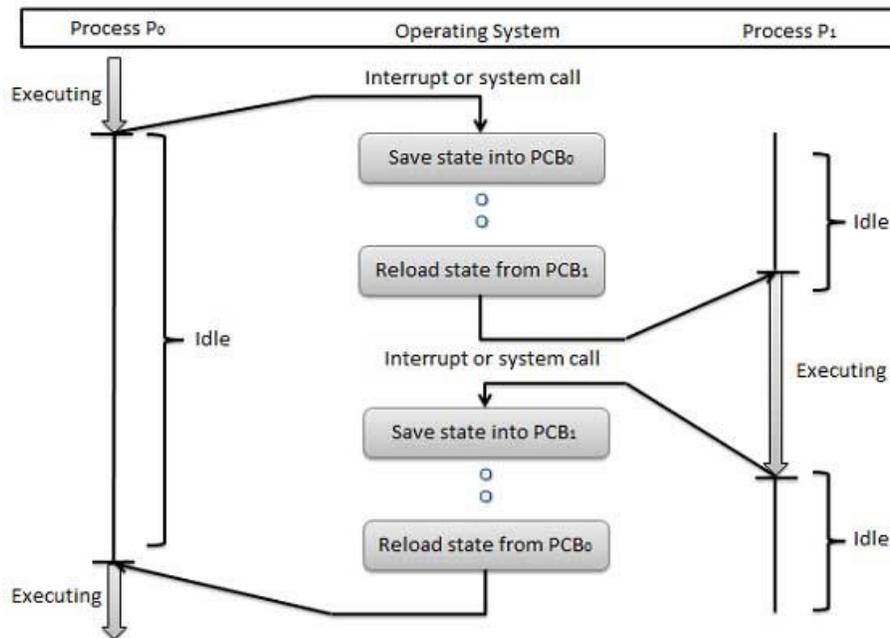
Context switch time is pure overhead. Context switching can significantly affect

(Explanation of scheduler: 2 marks)

(Explanation of context switch: 2 marks)



performance as modern computers have a lot of general and status registers to be saved. Content switching times are highly dependent on hardware support



f) State and explain four scheduling criteria.

4M

Ans: **CPU utilization:** In multiprogramming the main objective is to keep CPU as busy as possible. CPU utilization can range from 0 to 100 percent.

Throughput: It is the number of processes that are completed per unit time. It is a measure of work done in the system. When CPU is busy in executing processes, then work is being done in the system. Throughput depends on the execution time required for any process. For long processes, throughput can be one process per unit time whereas for short processes it may be 10 processes per unit time.

Turnaround time: The time interval from the time of submission of a process to the time of completion of that process is called as turnaround time. It is the sum of time period spent waiting to get into the memory, waiting in the ready queue, executing with the CPU, and doing I/O operations. It indicates the time period for which a process exists in the system.

Waiting time: - It is the sum of time periods spent in the ready queue by a process. When a process is selected from job pool, it is loaded into the main memory (ready

(Any four scheduling criteria: 1 mark each)



MODEL ANSWER
SUMMER- 17 EXAMINATION

Subject Title: Operating System

Subject Code: 17512

queue).A process waits in ready queue till CPU is allocated to it. Once the CPU is allocated to the process, it starts its execution and if required request for resources. When the resources are not available that process goes into waiting state and when I/O request completes, it goes back to ready queue. In ready queue again it waits for CPU allocation.

Response time:-The time period from the submission of a request until the first response is produced is called as response time. It is the time when system responds to the process request not the completion of a process. In the system, a process can Produce some output fairly early and can continue computing new results while previous results are being output to the user.

3.	Attempt any <u>FOUR</u> of the following:	16 Marks
a)	Describe activities of I/O system and secondary storage management. (four each)	4M
Ans:	<p>I/O System: Input / Output device management provides an environment for the better interaction between system and the I / O devices such as printers, scanners, tape drives etc. To interact with I/O devices in an effective manner, the operating system uses some special programs known as device driver. The device drivers take the data that operating system has defined as a file and then translate them into streams of bits or a series of laser. A device driver is a specific type of computer software that is developed to allow interaction with hardware devices. Typically this continues an interface for communicating with the I/O device, through the specific computer bus or communication subsystem that the hardware is connected with. The device driver is a specialized hardware dependent computer program that enables another program, typically an operating system to interact transparently with a hardware device, and usually provides the required interrupt handling necessary for the time dependent hardware interfacing.</p> <p>Activities:</p> <ul style="list-style-type: none"> • Providing interfaces to other system components. • Managing devices • Transferring data • Detecting I/O completion <p>Secondary storage Management: The computer system provides secondary storage to back up main memory. Secondary storage is required because main memory is too small to accommodate all data and programs, and the data that it holds is lost when power is lost. Most modern computer systems use disks as the principal on-line storage medium for programs and data. Most of the programs including compilers, assemblers, word</p>	(Description of four activities of I/O system: 2 marks, Description of four activities secondary storage:2 marks)



MODEL ANSWER
SUMMER- 17 EXAMINATION

Subject Title: Operating System

Subject Code: 17512

processors, editors, and formatters are stored on a disk until loaded into memory. Secondary storage consists of tapes drives, disk drives, and other media.

Activities:

- Free space management
- Storage allocation
- Storage de-allocation
- Disk scheduling.

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

b) With neat diagram, explain file access methods.

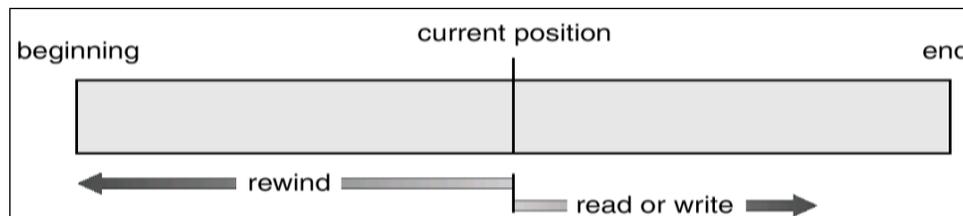
4M

Ans: There are two method to access file:

1. Sequential access
2. Direct access

1. Sequential Access Method: Information in the file is processed in order, one record after the other. This mode of access is by far the beginning current position most common; for example, editors and compilers usually access files in this fashion. Reads and writes make up the bulk of the operations on a file.

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



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

{(Two Methods Description of each: 2 marks (1 mark Diagram, 1 mark Explanation)}



MODEL ANSWER
SUMMER- 17 EXAMINATION

Subject Title: Operating System

Subject Code: 17512

he has to either fast-forward over all of the songs that come before it or listen to them. There is no way to jump directly to a specific song.

Direct Access Method: A file is made up of fixed-length logical records that allow programs to read and write records rapidly in no particular order. Thus, we may read block 14, then read block 53, and then write block 7. There are no restrictions on the order of reading or writing for a direct-access file. The direct-access method is based on a disk model of a file, since disks allow random access to any file block. Direct-access files are of great use for immediate access to large amounts of information. Databases are often of this type. For the direct-access method, the file operations must be modified to include the block number as a parameter. The block number provided by the user to the OS is normally a relative block number.

- A relative block number is an index relative to the beginning of the file.
- Thus, the first relative block of the file is 0, the next is 1, and so on, even though the actual absolute disk address of the block may be 14703 for the first block and 3192 for the second. When you work with a direct access file (which is also known as a random access file), you can jump directly to any piece of data in the file without reading the data that comes before it. This is similar to the way a CD player or an MP3 player works. You can jump directly to any song that you want to listen to. Sequential access files are easy to work with, and you can use them to gain an understanding of basic file operations.

Implementation for direct access

Cp = 0;

Read cp;

Cp = cp+1;

Write cp;

Cp = cp+1

c) Describe the critical-section problem.

4M

Ans: Each process contains two sections. One is critical section where a process may need to access common variable or objects and other is remaining section containing instructions for processing of sharable objects or local objects of the process. Each process must request for permission to enter inside its critical section. The section of code

(Relevant Description: 4 marks)

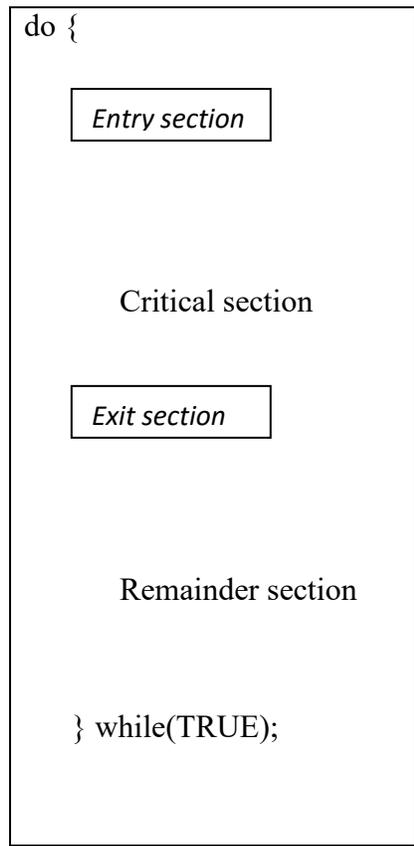


MODEL ANSWER
SUMMER- 17 EXAMINATION

Subject Title: Operating System

Subject Code: 17512

implementing this request is the entry section. In entry section if a process gets permission to enter into the critical section then it works with common data. At this time all other processes are in waiting state for the same data. The critical section is followed by an exit section. Once the process completes its task, it releases the common data in exit section. Then the remaining code placed in the remainder section is executed by the process.



Two processes cannot execute their critical sections at the same time. The critical section problem is to design a protocol that the processes can use to cooperate i.e. allowing entry to only one process at a time inside the critical section. Before entering into the critical section each process must request for permission to entry inside critical section.

d)	State and describe necessary conditions for dead lock.	4M
Ans:	1. Mutual Exclusion: The resources involved are non-shareable. At least one resource (thread) must be held in a non-shareable mode, that is, only one process at a time claims	(Four Conditions:



MODEL ANSWER
SUMMER- 17 EXAMINATION

Subject Title: Operating System

Subject Code: 17512

exclusive control of the resource. If another process requests that resource, the requesting process must be delayed until the resource has been released.

2. Hold and Wait: Requesting process hold already, resources while waiting for requested resources. There must exist a process that is holding a resource already allocated to it while waiting for additional resource that are currently being held by other processes.

3. No-Preemptive: Resources already allocated to a process cannot be preempted. Resources cannot be removed from the processes are used to completion or released voluntarily by the process holding it.

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

Description of each: 1 mark)

e) **With neat diagram, explain multilevel queue scheduling.**

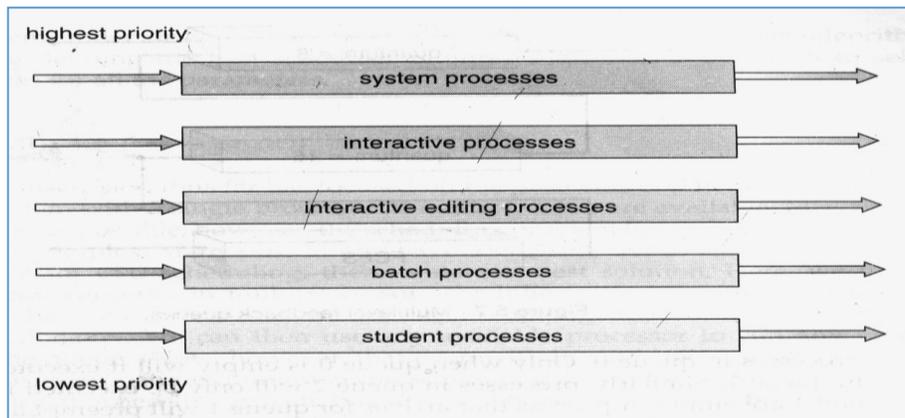
4M

Ans: Any relevant diagram shall be considered.

Multilevel queue scheduling classifies processes into different groups. It partitions the ready queue into several separate queues. The processes are permanently assigned to one queue based on some properties such as memory size, priority, process type, etc. Each queue has its own scheduling algorithm. In a system there are foreground processes and background processes. So system can divide processes into two queues: one for background and other for foreground. Foreground queue can be scheduled with Round Robin algorithm where as background queue can be scheduled by First Come First Serve algorithm. Scheduling is done for all the processes inside the queue as well as for all separate queues.

Example: Consider all the processes in the system are divided into four groups: system, interactive, interactive editing, batch and student processes queue. Each queue contains processes. CPU is first scheduled for all queues on may be priority, total burst time or process type.

(Explanation : 2 marks, Diagram: 2 marks)





MODEL ANSWER
SUMMER- 17 EXAMINATION

Subject Title: Operating System

Subject Code: 17512

There can many different ways to schedule various queues.

1. On the basis of priority, suppose system process queue has highest priority then processes from all other queue can be executed only when system process queue is empty. When a process from batch queue is executing, if new process arrives in the system queue then process from batch queue is pre-empted and process from system queue will be executed.
2. System can use Round Robin algorithm to schedule various queues. Time quantum can be defined for CPU allocation. For specified time, each queue will execute its own processes. For example, time quantum 40 milliseconds. CPU will be assigned first to system queue for 40 milliseconds. Processes from system queue executes one by one for 40 ms. Once the time quantum expires current process is pre-empted and CPU is assigned to interactive queue for time quantum of 40 ms. like this each queue executes one by one in circular form i.e starting with system queue then interactive queue then batch queue then student queue and again system queue and so on in circular fashion.

4. a) Attempt any **THREE** of the following:

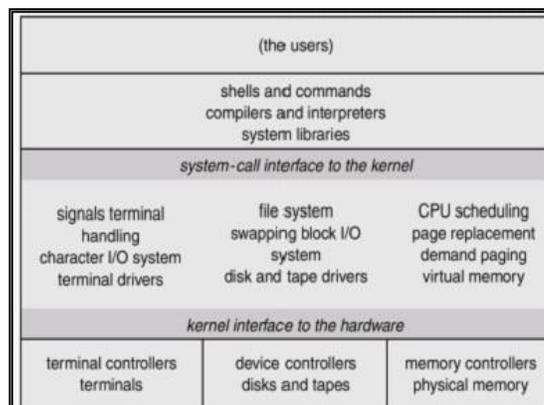
12 Marks

(i) With neat diagram, Explain Monolithic structure of operating system.

4M

Ans: **Monolithic Systems:** System is divided into multiple modules written as a collection of procedures. Each module is designed for performing specific task such as file management, I/O management or memory management and so on. Any module can call any other module without any major restrictions. Operating system distinguishes between system mode and user mode while executing an application program. An application program runs in the user mode. A user makes a request for a service using application programs. Application programs request for a system call to system call interface. The operating system locates a system call and executes it in the system mode. Once execution of system call is over, the execution of the application programs resumes in the user mode.

(Description: 2 marks, Diagram: 2 marks)



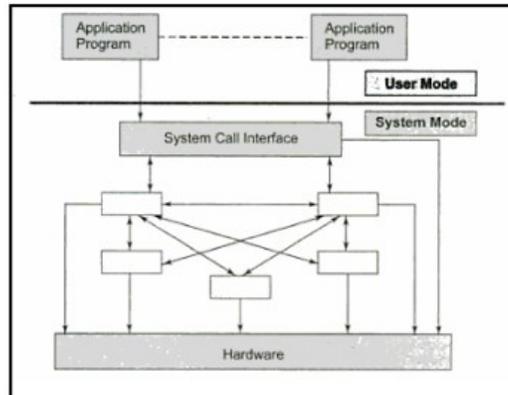


MODEL ANSWER
SUMMER- 17 EXAMINATION

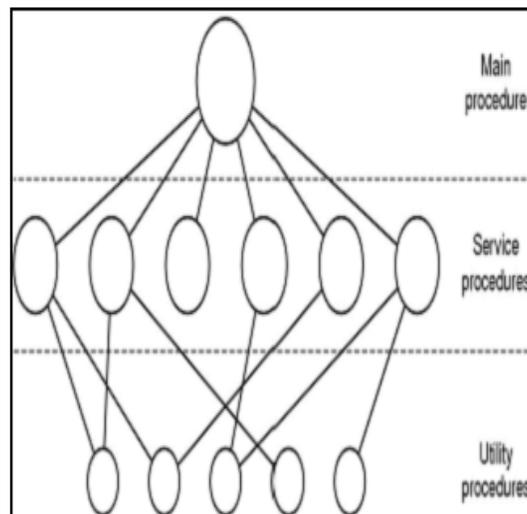
Subject Title: Operating System

Subject Code: 17512

OR



OR



(ii) Describe Process Control Block (PCB) with suitable diagram.

4M

Ans: Each process is represented as a process control block (PCB) in the operating system. It contains information associated with specific process.

(Description:
2 marks,



Process State: It indicates current states of a process. Process state can be new, ready, running, waiting and terminated.

Process number: Each process is associated with a unique number.

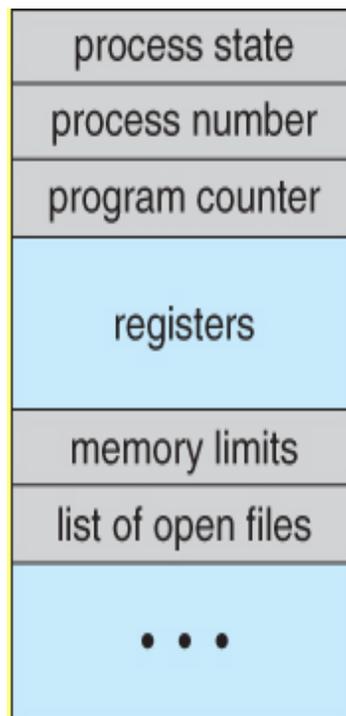
Process Counter: It indicates the address of the next instruction to be executed for the process.

CPU Registers: The registers vary in number and type depending on the computer architecture. Register includes accumulators, index registers, stack pointers and general purpose registers plus any condition code information.

Memory Management Information: It includes information such as value of base and limit registers, page tables, segment tables, depending on the memory system used by OS.

Accounting Information: This information includes the amount of CPU used, time limits, account holders, job or process number and so on. It also includes information about listed I/O devices allocated to the process such as list of open files.

Diagram: 2 marks)





MODEL ANSWER
SUMMER- 17 EXAMINATION

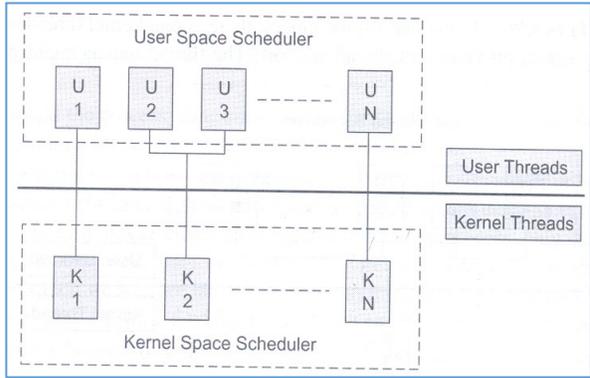
Subject Title: Operating System

Subject Code: 17512

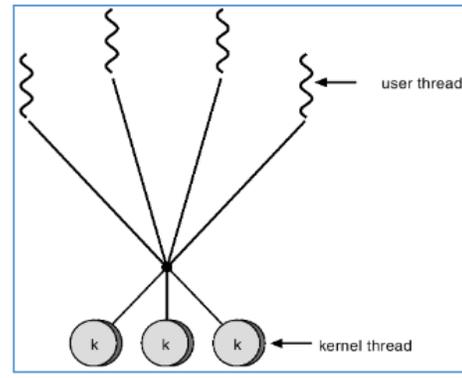
	(iii)	Write use of following system calls. 1) fork() 2) exec() 3) abort() 4) end()	4M
	Ans:	1. fork () This system call is used to create a new process. 2. exec() This system call is used to replace the process's memory space with a new program. It loads a binary file into memory and starts its execution. 3. abort() This system call is used to halt process execution abnormally. 4. end () This system call is used to halt process execution normally.	(Use of each System Call: 1 mark)
	(iv)	Write benefits of using threads.	4M
	Ans:	The benefits of using threads: 1. Responsiveness:- Multithreading an interactive application may allow a program to continue running even if part of it is blocked or is performing a lengthy operation, thereby increasing responsiveness to the user. For instance, a multithreaded web browser could still allow user interaction in one thread while an image was being loaded in another thread. 2. Resource sharing:- By default, threads share the memory and the resources of the process to which they belong .The benefit of sharing code and data is that it allows an application to have several different threads of activity within the same address space. 3. Economy: - Allocating memory and resources for process creation is costly. Because threads share resources of the process to which they belong, it is more economical to create and context-switch threads. Empirically gauging the difference in overhead can be difficult, but in general it is much more time consuming to create and manage processes than threads. In Solaris, for example, creating a process is about thirty times slower than is creating a thread, and context switching is about five times slower. 4. Utilization of multiprocessor architectures: - The benefits of multithreading can be greatly increased in a multiprocessor architecture, where threads may be running in parallel on different processors. A single threaded process can only run on one CPU, no matter how many are available. Multithreading on a multi-CPU machine increases concurrency.	(Four Benefits:1 mark each)



b)	Attempt any <u>ONE</u> of the following:	6 Marks
i)	With neat diagram, explain many to one and many to many multithreading model with its advantages and disadvantages.	6M
Ans:	<p>Many to one multithreading model: This model maps many user level threads to one kernel level thread. Thread management is done by thread library in user space.</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div data-bbox="225 825 807 1176"> </div> <div data-bbox="829 993 873 1020">OR</div> <div data-bbox="894 825 1326 1176"> </div> </div> <p>Advantages:</p> <ul style="list-style-type: none"> • It is an efficient model as threads are managed by thread library in user space. <p>Disadvantages:</p> <ul style="list-style-type: none"> • 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. <p>Many to many multithreading model: This model maps many user level threads to a smaller or equal number of kernel threads. Number of kernel threads may be specific to either a particular application or particular machine.</p>	<p>(Description of many to one: 2 marks ,one Advantage: 1/2mark, one Disadvantage:1/2mark , Description of many to many: 2 marks ,one Advantage: 1/2mark , one Disadvantage-1/2mark)</p>



OR



Advantages:

- Developer can create as many user threads as necessary.
- Threads can run in parallel on a multiprocessor.
- When a thread performs a blocking system call, the kernel can schedule another thread for execution.

Disadvantage:

- Complex to implement
- Debugging an application is complicated.

(ii) Explain any two file allocation methods with the help of diagram.

6M

Ans:

1. Contiguous Allocation

The contiguous allocation method requires each file to occupy a set of contiguous address on the disk. Disk addresses define a linear ordering on the disk. With this ordering, accessing block $b+1$ after block b normally requires no head movement. Contiguous allocation of a file is defined by the disk address and the length of the first block. If the file is n blocks long, and starts at location b , then it occupies blocks $b, b+1, b+2, \dots, b+n-1$. The directory entry for each file indicates the address of the starting block and the length of the area allocated for this file.

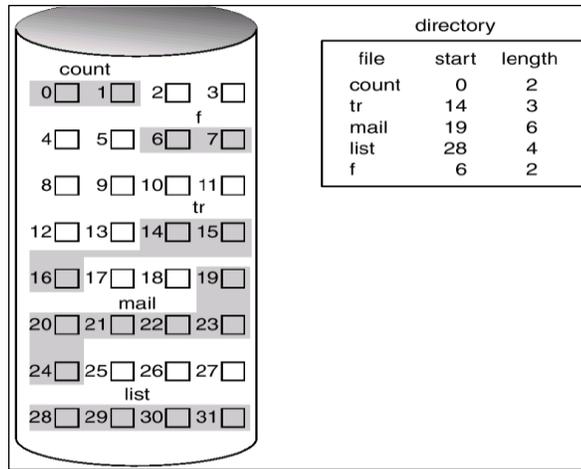
(Description of Any two, each allocation method Description :2 marks, Diagram)



MODEL ANSWER
SUMMER- 17 EXAMINATION

Subject Title: Operating System

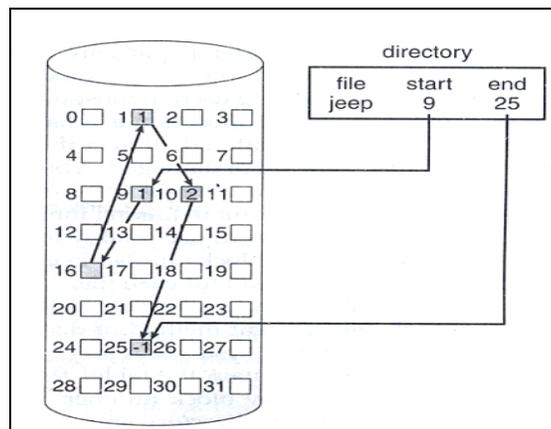
Subject Code: 17512



2. linked Allocation:-

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

For example, a file of five blocks starting with block 9 and continue with block 16, then block 1, then block 10 and finally block 25. Each allocated block contains a pointer to the next block.



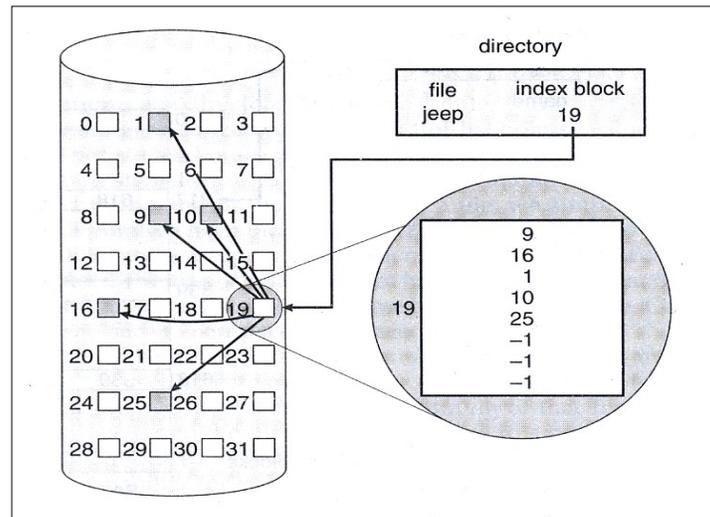


MODEL ANSWER
SUMMER- 17 EXAMINATION

Subject Title: Operating System

Subject Code: 17512

3. Indexed Allocation:- In this method, each file has its own index block. This index block is an array of disk block addresses. When a file is created, an index block and other disk blocks according to the file size are allocated to that file. Pointer to each allocated block is stored in the index block of that file. Directory entry contains file name and address of index block. When any block is allocated to the file, its address is updated in the index block. Any free disk block can be allocated to the file. Each i^{th} entry in the index block points to the i^{th} block of the file. To find and read the i^{th} block, we use the pointer in the i^{th} index block entry.



5. Attempt any TWO of the following:

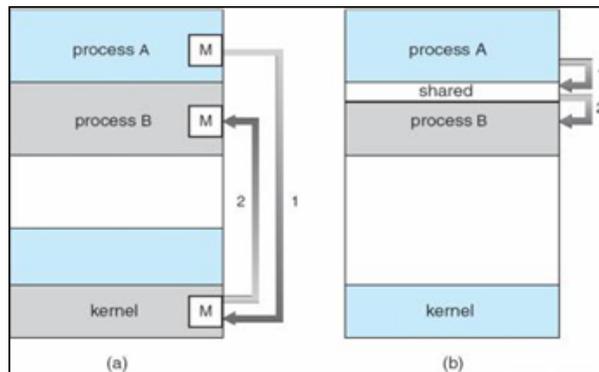
16 Marks

a) With neat diagram, explain Message passing system. Also describe the following:
(i) Naming
(ii) Synchronization
(iii) Buffering

8M

Ans: Message passing system
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.

(Diagram: 3 marks,
Explanation: 2 marks,
Description of given terms: 1 mark each)



(i) Naming:

Processes which wish to communicate with each other need to know each other with the name for identification. There are two types of communications :

1. Direct Communication
2. Indirect Communication

In direct communication each process that want to communicate must be explicitly use name for the sender as well as receiver while communication.

In this type the send() and receive() primitives are defined as follows:

Send(P,message) – Send message to process P

Receive (Q, message) – Receive a message from process Q.

In an indirect communication the messages could be send or receive from mailboxes or ports. A mailbox can be viewed as an object in which messages could be kept or even removed. Each mailbox is associated with the unique number.

In this type the send() and receive() primitives are defined as follows:

Send(A,message) – Send message to mailbox A.

Receive (A, message) – Receive a message from mailbox A.

(ii) Synchronization:

Communication between the processes takes place through the system calls. OS has to maintain proper synchronization between the sending and receiving processes.

To send() and receive() primitives, a special design is required for the implementation of these primitives.

These are also known as synchronous and asynchronous communication.



MODEL ANSWER
SUMMER- 17 EXAMINATION

Subject Title: Operating System

Subject Code: 17512

(iii) Buffering:

The communication could be direct or indirect. The message exchanged by the communicating processes resides or stores in a temporary queue.

The OS will buffer the messages into the buffers that are created in the system Address space. A sender's message will be copied from the sender's address space to the next free slot in the system buffers. From this system buffer, the messages will be delivered to the receiver process in FCFS order when receiver process executes receive calls.

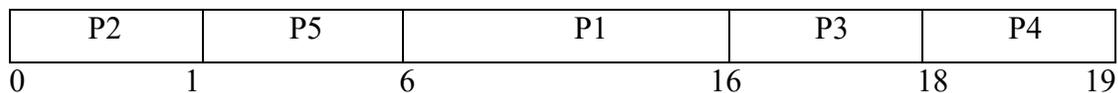
b) Consider the following set of processes, with the length of the CPU burst given in milliseconds. 8M

Process	Burst Time	Priority
P ₁	10	3
P ₂	1	1
P ₃	2	3
P ₄	1	4
P ₅	5	2

Find out average waiting time by using

- (i) nonpreemptive priority**
- (ii) Round-Robin (RR) (quantum = 1)**

Ans: (i) nonpreemptive priority (lowest number- highest priority):



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

(Gantt chart: 2 marks each, Average waiting time: 2 marks each)



MODEL ANSWER
SUMMER- 17 EXAMINATION

Subject Title: Operating System

Subject Code: 17512

(ii) Round-Robin (RR) (quantum = 1):

P1	P2	P3	P4	P5	P1	P3	P5	P1	P5	P1	P5	P1	P5	P1	P1	P1	P1	P1	
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

Waiting time for each process: $p_1(14-5)=9, P_2=1, P_3(6-1)=5, P_4=3, P_5(13-4)=9$
Average waiting time $= (9+1+5+3+9)/5 = 5.4$ milliseconds

c) Write steps for Banker's algorithm to avoid dead lock. Also give one example showing working of Banker's Algorithm. 8M

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 updating 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 to any process in Matrix B and not a single process can complete 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 resources 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.

(Steps: 4 marks, any relevant Example : 4 marks)



Example:

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

Snapshot at time T_0 :

	<u>Allocation</u>			<u>Max</u>			<u>Available</u>		
	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*

	<u>Need</u>		
	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 $\langle P_1, P_3, P_4, P_2, P_0 \rangle$ satisfies safety criteria

6. Attempt any **FOUR** of the following:

16 Marks

a) What is system call? With the help of diagram explain open () system call.

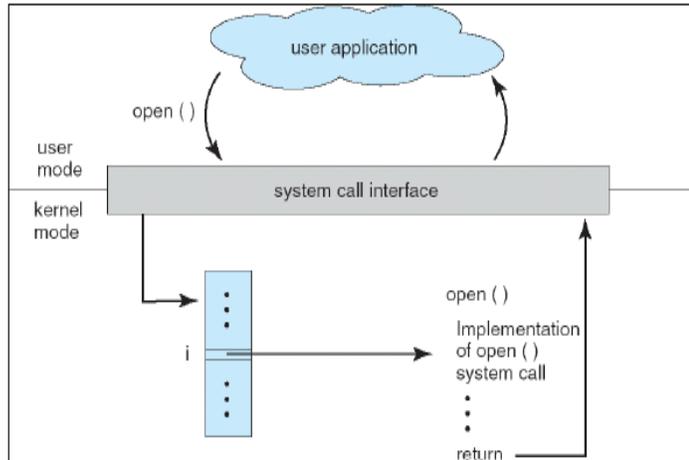
4 M

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 and 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

(Definition : 1½ marks , Diagram: 1½ marks, Open System Call:1 mark)



managed by run-time support library.



Open () system call

For most file systems, a program initializes access to a file in a file system using the **open** system call. This allocates resources associated to the file (the file descriptor), and returns a handle that the process will use to refer to that file.

b) Compare UNIX and LINUX.

4M

Ans:

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

(Any four points 1 mark each)



MODEL ANSWER

SUMMER- 17 EXAMINATION

Subject Title: Operating System

Subject Code: 17512

			server program model were essential elements in the development of the Internet
	Manufacturer	Linux kernel is developed by the community. Linus Torvalds oversees things.	Three biggest distributions are Solaris (Oracle), AIX (IBM) & HP-UX Hewlett Packard. And Apple Makes OSX, an unix based os..
	Usage	Linux can be installed on a wide variety of computer hardware, ranging from mobile phones, tablet computers and <u>video game consoles</u> , to mainframes and supercomputers.	The UNIX operating system is used in internet servers, workstations & PCs. Backbone of the majority of finance infrastructure and many 24x365 high availability solutions.
	Development and Distribution	Linux is developed by Open Source development i.e. through sharing and collaboration of code and features through forums etc and it is distributed by various vendors.	Unix systems are divided into various other flavors, mostly developed by AT&T as well as various commercial vendors and non-profit organizations.
	GUI	Linux typically provides two GUIs, KDE and Gnome. But there are millions of alternatives such as LXDE, Xfce, Unity, Mate, twm, ect.	Initially Unix was a command based OS, but later a GUI was created called Common Desktop Environment. Most distributions now ship with Gnome.
	File system support	Ext2, Ext3, Ext4, Jfs, ReiserFS, Xfs, Btrfs, FAT, FAT32, NTFS	jfs, gpfs, hfs, hfs+, ufs, xfs, zfs format
	Text mode interface	BASH (Bourne Again SHell) is the Linux default shell. It can support multiple command interpreters.	Originally the Bourne Shell. Now it's compatible with many others including BASH, Korn & C.
	Price	Free but support is available for a price.	Some free for development use (Solaris) but support is available for a price.



MODEL ANSWER
SUMMER- 17 EXAMINATION

Subject Title: Operating System

Subject Code: 17512

		Security	Linux has had about 60-100 viruses listed till date. None of them actively spreading nowadays.	A rough estimate of UNIX viruses is between 85 -120 viruses reported till date.	
		Threat detection and solution	In case of Linux, threat detection and solution is very fast, as Linux is mainly community driven and whenever any Linux user posts any kind of threat, several developers start working on it from different	Because of the proprietary nature of the original Unix, users have to wait for a while, to get the proper bug fixing patch. But these are not as common.	
		Processors	Dozens of different kinds.	x86/x64, Sparc, Power, Itanium, PA-RISC, PowerPC and many others.	
		Examples	Ubuntu, Fedora, Red Hat, Debian, Archlinux, Android etc.	OS X, Solaris, All Linux	
		Architectures	Originally developed for Intel's x86 hardware, ports available for over two dozen CPU types including ARM	is available on PA-RISC and Itanium machines. Solaris also available for x86/x64 based systems. OSX is PowerPC(10.0-10.5)/x86(10.4)/x64(10.5-10.8)	
		Inception	Inspired by MINIX (a Unix-like system) and eventually after adding many features of GUI, Drivers etc, Linus Torvalds developed the framework of the OS that became LINUX in 1992. The LINUX kernel was released on 17th September, 1991	In 1969, it was developed by a group of AT&T employees at Bell Labs and Dennis Ritchie. It was written in "C" language and was designed to be a portable, multi-tasking and multi-user system in a time-sharing configuration	
	c)	Explain multiprogrammed O.S with suitable diagram.			4M
	Ans:	<p>Multiprogramming is a rudimentary form of parallel processing in which several programs are run at the same time on a uniprocessor. Since there is only one processor, there can be no true simultaneous execution of different programs. Instead, the operating system executes part of one program, then part of another, and so on. To the user it appears that all programs are executing at the same time.</p> <p>Why Multiprogramming?</p> <p>Multiprogramming needed for efficiency:</p> <ul style="list-style-type: none"> • Single user cannot keep CPU and I/O devices busy at all times. 			(Diagram: 2 marks, Explanation : 2 marks)

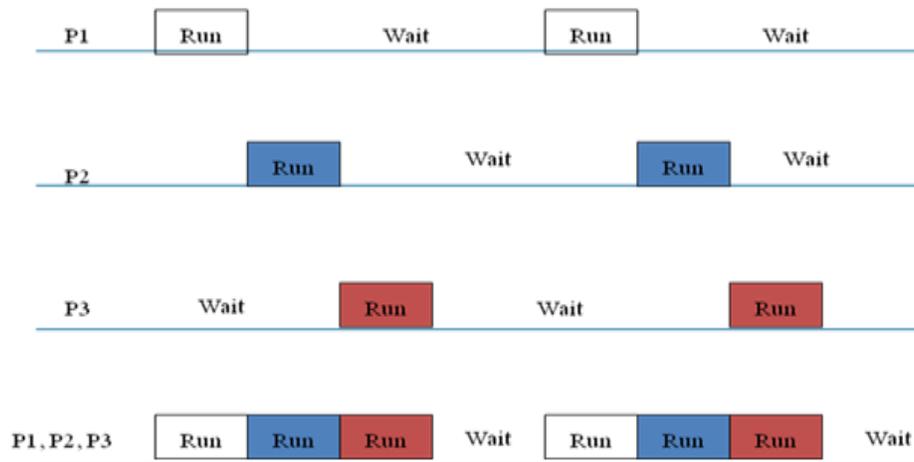


MODEL ANSWER
SUMMER- 17 EXAMINATION

Subject Title: Operating System

Subject Code: 17512

- Multiprogramming organizes jobs (code and data) so CPU always has one to execute.
- A subset of total jobs in system is kept in memory.
- One job selected and run via job scheduling.
- When it has to wait (for I/O for example), OS switches to another job.



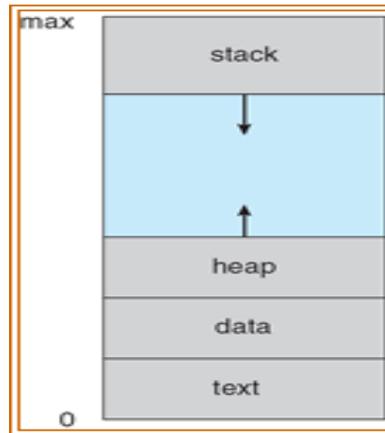
Multiprogramming is a technique used to utilize maximum CPU time by running multiple programs simultaneously. The execution begins with the first program and continues till an instruction waiting for a peripheral is reached, the context of this program is stored, and the second program in memory is given a chance to run. The process continued until all programs finished running. Multiprogramming has no guarantee that a program will run in a timely manner. Usually on a mainframe - the computer has a number of programs loaded into memory and the operating system switches quickly between them, processing a little bit of each one in turn. The high speed of the processor makes it seem like more than one program is being run at the same time. On a PC it is usually called multitasking. If the machine has the capability of causing an interrupt after a specified time interval, then the operating system will execute each program for a given length of time, regain control, and then execute another program for a given length of time, and so on. In the absence of this mechanism, the operating system has no choice but to begin to execute a program with the expectation, but not the certainty, that the program will eventually return control to the operating system. If the machine has the capability of protecting memory, then a bug in one program is less likely to interfere with the execution of other programs. In a system without memory protection, one program can change the contents of storage assigned to other programs or even the storage assigned to the operating system. The resulting system crashes are not only disruptive, they may be very difficult to debug since it may not be obvious which of several programs is at fault. If memory can hold several programs, then CPU can switch to another one whenever a program is awaiting for an I/O to complete.



d)	<p>Explain two level Directory Structure with suitable diagram.</p>	4M
Ans:	<p><u>Two level directory:</u> The standard solution to limitations of single-level directory is to create a separate directory for each user. In the two-level directory structure, each user has his own user file directory (UFD). The UFDs have similar structures, but each lists only the files of a single user. When a user job starts or a user logs in, the system's master file directory (MFD) is searched. The MFD is indexed by user name or account number, and each entry points to the UFD for that user.</p> <p>Advantages: i) Path name ii) Can have the same file name for different user iii) Efficient searching</p> <p>Disadvantages: No grouping capability</p> <div data-bbox="264 947 1354 1392" data-label="Diagram"> </div>	(Diagram: 2 marks, Explanation: 2 marks)
e)	<p>Describe Process in memory with diagram.</p>	4M
Ans:	<p>Process is program in execution. A process in execution needs resources like CPU, memory, I/O. Current machines allow several processes to share resources. In reality one processor is shared among many processes. Process is a program in execution. A process does not mean only program but it could contain some part called as text section. It may contain the current activity, represented by the value of the program counter & the contents of CPU register. Another part of process is stack. Stack is basically used to store the temporary values. That values can be parameter or return values of functions, local variables, addresses of the return values etc. There is another part called data section. Data section stores the global variable. Global variables are accessible</p>	(Explanation :2 marks, Diagram: 2 marks)



throughout the program. Sometimes it is necessary to use dynamic memory allocation while executing the program. The heap is a part, which is used for dynamic memory allocation.



Text Segment: The Text segment (the Instruction segment) contains the executable program code and constant data. The text segment is marked by the operating system as read-only and cannot be modified by the process. Multiple processes can share the same text segment. Processes share the text segment if a second copy of the program is to be executed concurrently. In this setting, the system references the previously loaded text segment with the pointer rather than reloading a duplicated. If needed, shared text, which is the default when using the C/C++ compiler, can be turned off by using the -N option on the compile time.

Data Segment: The data segment, which is contiguous (in a virtual sense) with the text segment, can be subdivided into initialized data (e.g. in C/C++, variables that are declared as static or are static by virtue of their placement) and uninitialized (or 0-initialized) data. The uninitialized data area is also called BSS (Block Started by Symbol). For example, Initialized Data section is for initialized global variables or static variables, and BSS is for uninitialized. **Stack Segment**

The stack segment is used by the process for the storage of automatic identifier, register variables, and function call information. Stack is basically used to store the temporary values. The temporary values can be parameter or return values of functions, local variables, addresses of the return values etc. In the above figure, the stack grows towards the uninitialized data segment.

The user area: In addition to the text, data, and stack segment, the OS also maintains for each process a region called the u area (User Area). The u area contains information specific to the process (e.g. open files, current directory, and signal action, accounting information) and a system stack segment for process use. If the process makes a system call (e.g., the system call to *write* in the function in main), the stack frame information



Subject Title: Operating System

Subject Code: 17512

		for the system is stored in the system stack segment. Again, this information is kept by the OS in an area that the process doesn't normally have access to. Thus, if this information is needed, the process must use special system call to access it. Like the process itself, the contents of the u area for the process are paged in and out by the OS.	
--	--	--	--