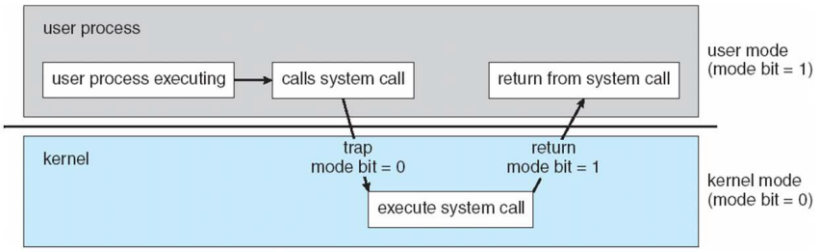
Q1. a. Explain the dual mode operations of monitor mode and user mode.

Answer:

In order to ensure proper execution of operating system, it must be possible to distinguish execution of operating system code and user code. Approach taken by systems is to provide hardware support to differentiate among various modes. At the least two modes: user mode and kernel mode (or system mode or supervisor mode or privileged mode) is provided. This is referred to as dual mode of operations.

A mode bit is provided to indicate user mode(1) or kernel mode(0). When user code is executing then system is in user mode and when operating system code is executing then it is kernel mode.

Following diagram shows how transition between modes take place.



b. Name the scheduler responsible for:

(i) swapping out partially executed programs

(ii) controlling the degree of multiprogramming

(iii) shifting a process from ready state to running state

(iv) selecting a proper mix of CPU bound and I/O bound processes

Answer:

(i) medium term scheduler
(ii) long term scheduler
(iii) short term scheduler or CPU scheduler
(iv) long term scheduler

c. How do one-time passwords help in user authentication?

Answer:

One-time passwords

To avoid problems of password being exposed, a system could use a set of paired passwords. When session begins, system randomly selects and presents one part of a password pair and user must provide the other part. This way for every session a separate password is generated and these are referred to as one-time passwords.

An approach for this is to use a function. For example, use a secret function (known only to user and system) and a random seed. User is challenged with the seed and he/she has to provide the answer on applying seed to the function as password. If results of user and system match then user is authenticated.

There are many other implementations of one time passwords. Some require user to enter PIN (personal identification number) on keypad and then one time password is generated is sent to the user through SMS / mail and user has to enter it also.

d. Explain file system mounting.

Answer:

Book [I] Page No. 379 – 381

e. What is a critical section? Mention the properties that a solution to the critical section should satisfy.

Answer:

Consider system of n processes $\{P_0, P_1, \ldots, P_{n-1}\}$. Each process has a segment of code called a critical section in which the process may be changing common variables, updating table, writing file, etc. When one process in critical section, no other may be in its critical section.

Any solution to critical section should satisfy the following properties:

1. Mutual Exclusion - If process P_i is executing in its critical section, then no other processes can be executing in their critical sections
2. Progress - If no process is executing in its critical section and there exist some processes that wish to enter their critical section, then the selection of the processes that will enter the critical section next cannot be postponed indefinitely
3. Bounded Waiting - A bound must exist on the number of times that other processes are allowed to enter their critical sections after a process has made a request to enter its critical section and before that request is granted

f. What is streaming in multimedia systems and what are its different types?

Answer:

Streaming is delivering a multimedia file from a server to a client - typically the delivery occurs over a network connection. There are two different types of streaming:

1. Progressive download - the client begins playback of the multimedia file as it is delivered. The file is ultimately stored on the client computer.
2. Real-time streaming - the multimedia file is delivered to - but not stored on - the client's computer. There are two types of real-time streaming:
   i. Live streaming - used to deliver a live event while it is occurring
   ii. On-demand streaming - used to deliver media streams such as movies, archived lectures, etc. The events are not delivered in real-time.

g. How is a stateful service different from a stateless service in distributed file systems? **(7 x 4)**

Answer:

There are two approaches for storing server side information when a client accesses remote files: either the server tracks each file being accessed by each client or it simply provides blocks as they are requested by the client without knowledge of how those blocks are used. In the former case, the service is known as stateful; in the latter case, it is stateless.

The mechanism of stateful service is as follows:

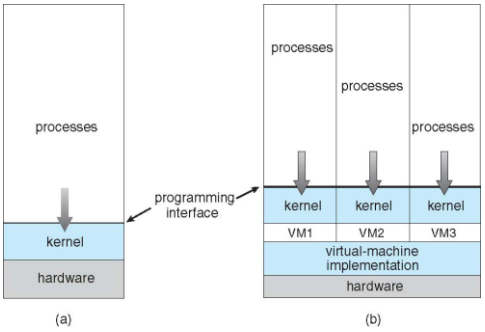
- Client opens a file
- Server fetches information about the file from its disk, stores it in its memory, and gives the client a unique connection identifier to the open file
- Identifier is used for subsequent accesses until the session ends
- Server must reclaim the main-memory space used by clients who are no longer active

However, stateless service avoids state information by making each request self-contained, each request identifies the file and position in the file and there is no need to establish and terminate a connection by open and close operations.

Q2. a. What are virtual machines and what are their advantages? (4 + 2)

Answer:

A virtual machine takes the layered approach to its logical conclusion. It treats hardware and the operating system kernel as though they were all hardware. The operating system host creates the illusion that a process has its own processor and (virtual memory). Each guest provided with a (virtual) copy of underlying computer. Multiple execution environments (different operating systems) can share the same hardware



(a) Nonvirtual machine (b) virtual machine

Advantages of virtual machines:

1. Host system is protected. A virus inside a guest operating system might damage that OS but is unlikely to affect the host or other guests.
2. Useful for development and testing. It is perfect vehicle for OS research and development.

b. Draw a resource allocation graph for the following scenario and determine whether the system is in a deadlock or not. Justify your answer.

P = {P1, P2, P3}

R = {R1, R2, R3, R4}

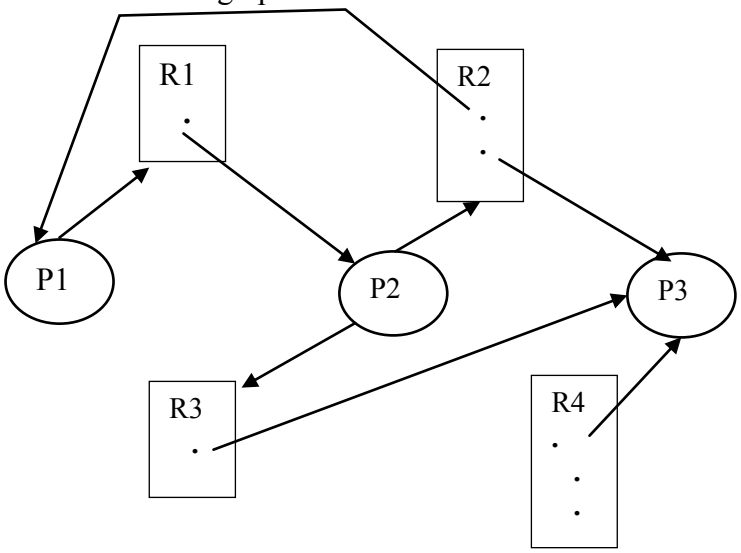
No. of instances of R1, R2, R3 and R4 are 1, 2, 1 and 3 respectively.

E = {P1 → R1, R1 → P2, R2 → P1, R2 → P3, P2 → R3, P2 → R2,

R3 → P3, R4 → P3} **(3 + 1 + 2)**

Answer:

The resource allocation graph is as follows:



The system is not in a deadlock. P3 can complete as it is not waiting for any resource. This releases R2 and R3 which will be available to P2 and it completes, followed by R1 available for P1 and it completes.

c. Explain the four main benefits of multithreaded programming. **(6)**

Answer:

The four main benefits of multithreaded programming are as follows:

1. Responsiveness. Multithreading an interactive application may allow a program to continue running even if part of it is blocked or is performing a lengthy operation, thereby increasing responsiveness to the user. For example, one thread in a web browser may be downloading and other thread may be interacting with the user.
2. Resource Sharing. Processes may only share resources through techniques such as shared memory or message passing explicitly. However, by defaultthreads share memory and resources of the process to which they belong.
3. Economy. Allocating memory and resources for process creation is costly. Because threads share the resources of the process to which they belong, it is more economical to create and context-switch threads.
4. Scalability. 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 processor, regardless of how many are available. Multithreading on multi-CPU machine increases parallelism.

Q3. a. Consider the following set of processes, with the length of CPU burst time given in milliseconds:

Process	Arrival Time	Burst Time	Priority
P1	**0**	**8**	**4**
P2	**1**	**5**	**3**
P3	**3**	**6**	**2**
P4	**5**	**2**	**1 (Highest)**

(i) Draw Gantt charts illustrating the execution of these processes using

(1) FCFS

(2) SJF (preemptive) or SRJF

(3) a non-preemptive priority based

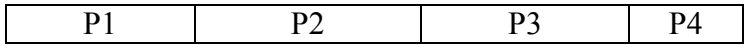
4) round robin with time quantum = 2

(ii) What is the average waiting time of each of the scheduling algorithms given in part (i)? (6 + 4)

Answer:

(i) and (ii)

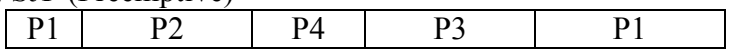
(1) FCFS



0 8 13 19 21

$$\text{Average waiting time} = \frac{0+7+10+14}{4} = \frac{31}{4} = 7.75$$

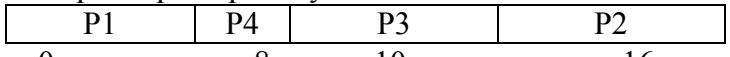
(2) SJF (Preemptive)



0 1 6 8 14 21

$$\text{Average waiting time} = \frac{13+0+5+1}{4} = \frac{19}{4} = 4.75$$

(3) Non-preemptive priority based



0 8 10 16 21

$$\text{Average waiting time} = \frac{0+15+7+3}{4} = \frac{25}{4} = 6.25$$

(4) Round robin (time quantum = 2)

P1	P2	P3	P4	P1	P2	P3	P1	P2	P3	P1

0 2 4 6 8 10 12 14 16 17 19 21

$$\text{Average waiting time} = \frac{13+11+10+1}{4} = \frac{35}{4} = 8.75$$

b. What is address binding? What are three different address binding options? Draw the diagram also depicting the stages. (8)

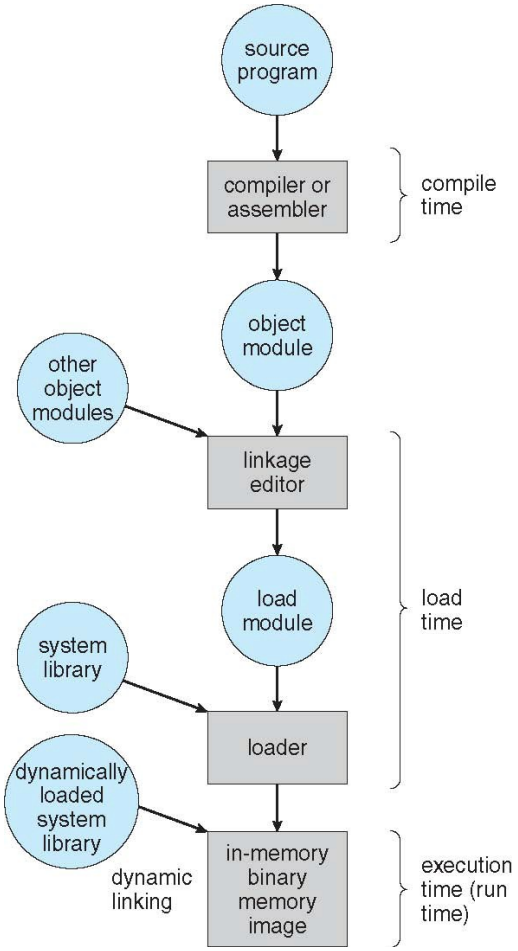
Answer:

A program resides on a disk as a binary executable code. To be executed, the program must be brought into the memory. The addresses of instructions and data present in the executable file (logical addresses) have somehow to be mapped to addresses of primary memory (physical or absolute addresses), corresponding to the locations where the executable file is loaded for execution. This is referred to as address binding.

Address binding of instructions and data to memory addresses can happen at three different stages:

1. Compile time: If memory location known at the time of compilation of program, absolute code can be generated. The code must be recompiled if starting location changes.
2. Load time: Must generate relocatable code if memory location is not known at compile time.
3. Execution time: Binding delayed until run time if the process can be moved during its execution from one memory segment to another.

The following diagram depicts the various stages at which address binding can take place:



Q4. a. Explain segmentation. Consider the following segment table:

Segment	Base	Length
0	200	600
1	1200	20
2	40	150

What are the physical addresses for the following logical addresses?

(i) 1, 30 (ii) 2, 100 (iii) 4, 10 (iv) 0, 412 (4 + 2)

Answer:

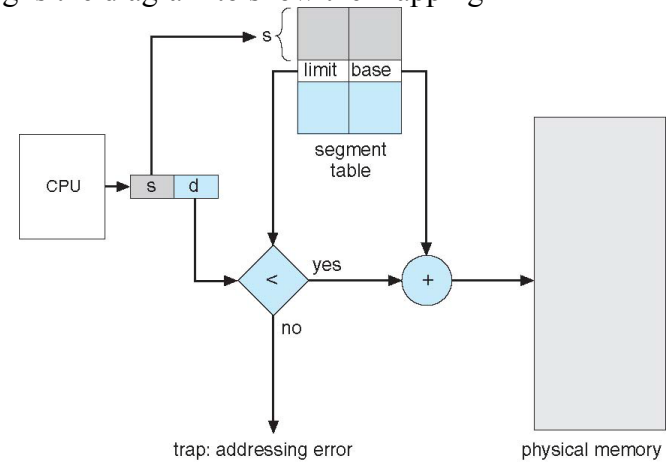
Segmentation: It is a memory-management scheme that supports user view of memory. For example, a C compiler may create separate segments for say functions, global variables, etc. The logical address space is a collection of segments consisting of segment number (s) and an offset (d) within the segment.

Mapping of logical addresses to physical addresses is done using segment table where each table entry has:

base – contains the starting physical address where the segments reside in memory

limit – specifies the length of the segment

Following is the diagram to show the mapping



	Logical Address	Physical Address
(i)	1, 30	Illegal address as offset is more than length
(ii)	2, 100	40 + 100 = 140
(iii)	4, 10	Illegal address as the segment number is till 2 only
(iv)	0, 412	200 + 412 = 612

b. What are Semaphores? How do they implement mutual exclusion. (2+4)

Answer:

Semaphore: It is a synchronization tool. A semaphore S is an integer variable which can be accessed only via two indivisible (atomic) operations: wait() and signal()as defined below:

```
wait (S) {
    while S <= 0
    ; // no-op
    S--;
}
signal (S) {
    S++;
}
```

The semaphores can be used to implement mutual exclusion by executing a wait() before entry into the critical section and signal() at the exit of critical section as the following code segment shows (initialize semaphore mutex to 1):

```
do {
    wait (mutex);
    // Critical Section
    signal (mutex);
    // remainder section
} while (TRUE);
```

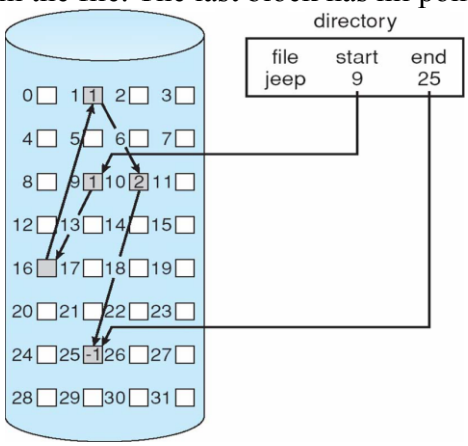
Whichever process will first executes wait(), it will be allowed to enter into the critical section as it will find mutex to be 1 and rest of the processes will have to wait

as the first process that enters the critical section will set mutex to 0 thus ensuring mutual exclusion.

c. What is linked allocation of disk blocks? What are its advantages and disadvantages? (6)

Answer:

Linked allocation – Each file is a linked list of disk blocks which may be scattered anywhere on the disk. Each block contains data and a pointer to next block in sequence in the file. The last block has nil pointer.



Free space management system called when new block needed. Pointers are followed to read a file.

Advantages:

1. No external fragmentation
2. No need to declare file size at the time of creation.

Disadvantages:

1. Reliability can be a problem
2. Locating a block can take many I/Os and disk seeks
3. Internal fragmentation is possible

Q5. a. Consider the following page reference string

2, 4, 1, 2, 3, 2, 0, 1, 2, 4, 0, 4, 3, 2, 3, 4

How many page faults would occur with LRU and FIFO page replacement algorithms assuming three frames? All frames are initially empty. (4 + 4)

Answer:

Note:An empty column represents that there is no page fault and the contents of memory are same as the previous column.

LRU

2	4	1	2	3	2	0	1	2	4	0	4	3	2	3	4
2	2	2		2		2	2		2	2		3	3		

	4	4		3		3	1		1	0		0	2		
		1		1		0	0		4	4		4	4		

The number of page faults is 10.

FIFO

2	4	1	2	3	2	0	1	2	4	0	4	3	2	3	4
2	2	2		3	3	3	1		1			1	2		
	4	4		4	2	2	2		4			4	4		
		1		1	1	0	0		0			3	3		

The number of page faults is 10.

b. Explain naming and transparency in distributed file system (5)

Answer:

c. Explain Bully algorithm and Ring algorithm in Distributed Synchronization (2.5 + 2.5)

Answer:

Book [1] Page No. 583 – 586

Q6. a. Suppose the read/write head is at track 97, moving towards track 0 (the lowest numbered track on the disk) and the disk request queue contains read/write requests for the sectors on tracks 84, 155, 103, 96, 15, 24, 87, 114, 55 and 197, respectively. The highest track number is 199. Calculate the total number of head movements needed to satisfy the requests in the queue using

(i) SSTF

(ii) SCAN

(iii) C-SCAN

(iv) LOOK (4 x 3)

Answer:

(i) SSTF: The sequence in which the requests are services is as follows:
97 - 96 - 103 - 114 - 87 - 84 - 55 - 24 - 15 - 155 - 197
The total number of head movements is (97 – 96) + (114 – 96) + (114 – 15) + (197 – 15) = 300

(ii) SCAN: The sequence in which the requests are services is as follows:
97 - 96 - 87 - 84 - 55 - 24 - 15 - 0 - 103 - 114 - 155 - 197
The total number of head movements is (97 – 0) + (197 - 0) = 294

(iii) C-SCAN: The sequence in which the requests are services is as follows:
97 - 96 - 87 - 84 - 55 - 24 - 15 - 0 - 197 - 155 - 114 - 103
The total number of head movements is (97 – 0) + (197 - 0) + (197 – 103) = 388

OR

The total number of head movements is (97 – 0) + (197 - 103) = 191
Note: There are two possible ways to count the number of head movements as some texts add the head movements on return trip also and some do not, so both the answers are correct. The student can write any one of the answers.

(iv) LOOK: The sequence in which the requests are services is as follows:
97 - 96 - 87 - 84 - 55 - 24 - 15 - 103 - 114 - 155 - 197
The total number of head movements is (97 – 15) + (197 - 15) = 264

Note: The student may draw the diagram also to show the head movement which is also a correct way to answer the question.

b. Write short notes on

(i) need-to-know principle

(ii) encryption **(2 + 4)**

Answer:

(i) Need-to-know principle: A process should be allowed to access only those resources that it requires to complete its task at hand. It is useful in limiting the amount of damage a faulty process can cause in the system. For example, when a process p invokes a procedure A(), the procedure should be allowed to access only its own variables and the formal parameters passed to it; it should not be allowed to access all the variables of process p. The goal of need-to-know principle is to minimize the risk of possible security violations.

(ii) Encryption: It is a means to constrain the possible receivers of a message. An encryption algorithm enables the sender of a message to ensure that only a computer possessing a certain key can read the message.
Encryption algorithm consists of

- Set K of keys
- Set M of Messages
- Set C of ciphertexts (encrypted messages)
- A function E : K → (M→C). That is, for each k ∈ K, E(k) is a function for generating ciphertexts from messages
  Both E and E(k) for any k should be efficiently computable functions
- A function D : K → (C → M). That is, for each k ∈ K, D(k) is a function for generating messages from ciphertexts
  Both D and D(k) for any k should be efficiently computable functions

An encryption algorithm must provide this essential property:

- Given a ciphertext c ∈ C, a computer can compute m such that E(k)(m) = c only if it possesses D(k)
- Thus, a computer holding D(k) can decrypt ciphertexts to the plaintexts used to produce them, but a computer not holding D(k) cannot decrypt ciphertexts

- Since ciphertexts are generally exposed (for example, sent on the network), it is important that it be infeasible to derive D(k) from the ciphertexts

There are two types of encryption algorithms: symmetric and asymmetric. In a symmetric form, the same key is used to encrypt and decrypt and in case of asymmetric, different keys are used.

Q7. a. Explain character and block devices in Unix System. (6)

Answer:

Character devices:

- The device drivers of such devices do not offer random access to fixed blocks of data.
- A character device driver must register a set of functions which implement the driver's various file I/O operations.
- The kernel performs almost no preprocessing of a file read or write request to a character device, but simply passes on the request to the device.
- The main exception to this rule is the special subset of character device drivers which implement terminal devices, for which the kernel maintains a standard interface.

Block devices:

- Provide the main interface to all disk devices in a system
- The block buffer cache serves two main purposes:
  - it acts as a pool of buffers for active I/O
  - it serves as a cache for completed I/O
- The request manager manages the reading and writing of buffer contents to and from a block device driver

b. Briefly explain the four main characteristics of real time systems. (6)

Answer:

The four main characteristics of real-time systems are as follows:

(i) Single purpose: Unlike PCs which are put many uses, a real-time system typically serves only a single purpose, such as controlling antilock brakes or delivering music on MP3 player.

(ii) Small size: Many real-time systems exist in environments where physical space is constrained. Consider the amount of space available in wristwatch or microwave oven – it is considerably less than available in PCs. As a result, most real-time systems lack both the CPU processing power and the amount of memory available in standard PCs.

(iii) Inexpensively mass-produced: They are usually found in home appliances and consumer devices such as digital cameras, microwave ovens, etc. which are mass produced in cost conscious environments. Thus microprocessors must be inexpensively mass-produced.

(iv) Specific timing requirements: This is the defining characteristic of real – time systems. They may be deployed in situations such as antilock brake systems, pacemakers, etc. where if the response is delayed may result in damage.

c. What is the wait-die and wound-wait scheme of deadlock prevention in distributed synchronization? (6)

Answer:

Wait-Die Scheme

- Based on a nonpreemptive technique
- If P_i requests a resource currently held by P_j, P_i is allowed to wait only if it has a smaller timestamp than does P_j (P_i is older than P_j)
- Otherwise, P_i is rolled back (dies)
- Example: Suppose that processes P_1, P_2, and P_3 have timestamps 5, 10, and 15 respectively
    - if P_1 request a resource held by P_2, then P_1 will wait
    - If P_3 requests a resource held by P_2, then P_3 will be rolled back

Wound-Wait Scheme

- Based on a preemptive technique; counterpart to the wait-die system
- If P_i requests a resource currently held by P_j, P_i is allowed to wait only if it has a larger timestamp than does P_j (P_i is younger than P_j). Otherwise P_j is rolled back (P_j is wounded by P_i)
- Example: Suppose that processes P_1, P_2, and P_3 have timestamps 5, 10, and 15 respectively
    - If P_1 requests a resource held by P_2, then the resource will be preempted from P_2 and P_2 will be rolled back
    - If P_3 requests a resource held by P_2, then P_3 will wait

TEXT BOOK

I. Operating System Principles, Abraham Silberschatz, Peter Baer Galvin, Greg Gagne(2009), Johnwiley & Sons (Asia) Pte Ltd

II. Modern Operating Systems" Andre S Tanenbaum(2009) Pearson Education