Q.1 a. What do you understand by Process, Thread and Fiber? Give a suitable example for each of them.

Answer:

Process is a program in execution and is an active entity. It is associated with certain items. Thread is finer execution entity and number of items possessed by the thread are much lesser than process. User threads are called fiber, whereas kernel thread is called thread.

b. Explain what is multilevel feedback scheduling with suitable example.

Answer:

In multilevel feedback many queues of the processes are created. These queues are operated with different scheduling policy. The processes which may not get the turn for CPU in any of these queues are fed back to the first one again.

c. Discuss file system in MS-DOS. Compute number of entries required in G FAT table for the given following parameters:
Disk capacity-30Mbyte, Block size-512 bytes, Blocks/cluster-2.

Answer:

File Storage in MS-DOS is based on a variant of linked allocation. It uses the link in a separate place called the FAT. The number of entries is $30*2^{10.}$

d. A system is in unsafe state. Is it possible for processes to complete their execution without entering into deadlock? If yes, show how?

Answer:

Yes it is possible to complete the execution without entering in deadlock. It is because all unsafe state is not a deadlocked state. One of the processes may complete without demanding any more resources and thus other processes may also get the resources released and completes their execution.

e. How are critical section and the principle of mutual exclusion related to each other?

Answer:

Critical section is the portion of the program that requires to be executed mutually exclusively. Semaphore is often used on which the actions can be performed in an atomic manner. Whenever an action is to be performed on mutually exclusive shared resources, it is better to use semaphore.

f. Discuss the merits and demerits of buddy.

Answer:

Buddy system operates with allocating the memory to the processes in chunks of some power of two. Initially, memory is considered to be a big space and eventually divided by halving it, unless it is just enough to accommodate the process. When the memory is freed by the process it is merged with adjacent buddy of same size, if it is available.

Q.2 a. Following is the snapshot of a CPU

Process	CPU Burst	Arrival Time
P1	**10**	**0**
P2	**29**	**1**
P3	**03**	**2**
P4	**07**	**3**

Draw the Gantt chart and calculate the turnaround time and waiting time of the jobs for FCFS (First Come First Served), SJF (Shortest Job First), SRTF (Shortest Remaining Time First) and RR (Round Robin with time quantum 10) scheduling algorithms.

Answer:

Policy	Turnaround	Waiting
FCFS	P1 10 P2 39 P3 42 P4 49	P1 0 P2 10 P3 39 P4 42
SJF	P1 20 P2 49 P3 03 P4 10	P1 10 P2 13 P3 0 P4 03
SRTF	P1 20 P2 49 P3 05 P4 12	P1 15 P2 19 P3 00 P4 02
Round Robin	P1 10 P2 49 P3 23 P4 30	P1 0 P2 20 P3 20 P4 23

b. A CPU scheduling algorithm determines an order for the execution of its scheduled processes. Given n processes to be scheduled on one processor, how many different possible schedules are there? Give a formula in terms of n.

Answer:

n!

Q.3 a. Assuming a cluster size of 512 bytes, calculate the percentage of wastage in file space due to incomplete filling of last clusters, if the file sizes are:
(i) 1200 bytes
(ii) 20,000 bytes.

Answer:

(i) 1200/512= 2.34, so 3 blocks are required. Wasted space in third block is 1526-1200= 326 bytes. Calculate in %.

(ii) Similar calculation for this.

b. At some point in time, the following holes (in the order) are created by a variable partition of memory. 20K, 15K, 40K, 60K, 10K, 25K. For a new process of 25 K, which hole would be filled using best fit, first fit, and worst fit?

Answer:

For 25 K using best fit the hole of 25 K is preferred. For first fit, 40 K is preferred and for worst fit 60K is preferred.

Q.4 a. What is the "Locality of Reference" concept and why it is important? What is the need to have a logical to physical map? Is it by design or incidental that the page sizes are chosen to be power of two?

Answer:

Locality of reference is the term used to determine the memory requirement of the processes. It is basically the reference during the execution of a program which moves from one locality to another. Logical and Physical memory separates the views of the user and OS towards the memory. User may view the memory as a contiguous space whereas the memory may be fragmented. It is by design that the page sizes is in power of 2. It is to facilitate the mapping.

b. Consider a system consisting of *m* resources of the same type, being shared by *n* processes. Resources can be requested and released by processes only one at a time. Show that the system is deadlock-free if the following two conditions hold:

(i) The maximum need of each process is between 1 and *m* resources

(ii) The sum of all maximum needs is less than *m + n*

Answer:

Using the terminology, we have:

a. $\sum_{i=1}^{n} Max_i < m + n$

b. $Max_i >= 1$ for all i

Proof: $Need_i = Max_i - Allocation_i$

If there exists a deadlock state then:

c. $\sum_{i=1}^{n} Allocation_i = m$

Use a. to get: $\sum Need_i + \sum Allocation_i = \sum Max_i < m + n$

Use c. to get: $\sum Need_i + m < m + n$

Rewrite to get: $\sum Need_i < n$

This implies that there exists a process *Pi* such that $Need_i = 0$. Since $Max_i >= 1$ it follows that P_i has at least one resource that it can release. Hence the system cannot be in a deadlock state.

Q.5 a. Assume that we have a paging system with page table stored in memory. If a memory reference takes 200 ns, how long does a paged memory reference take? If we add associative registers and 75% of all page table references are found in the associative registers, than what is the effective memory reference time? Assume that finding a page table entry in the associative registers takes zero time if the entry is there.

Answer:

The paged memory reference will take 400 ns.

Adding associative register will be 200*.75+400+.25=250 ns.

b. For the given snapshot of a system:

	Allocation	Max	Available
	A B C D	A B C D	A B C D
P1	0 0 1 2	0 0 1 2	1 5 2 0
P2	1 0 0 0	1 7 5 0	
P3	1 3 5 4	2 3 5 6	
P4	0 6 3 2	0 6 5 2	
P5	0 0 1 4	0 6 5 6	

Answer the following using Banker's Algorithm:

(i) What is the content of the matrix *Need*?

(ii) Is the system in a safe state?

(iii) If a request from process P2 arrives for (0, 4 2, 0), will it be granted?

Answer:

(i) Need matrix can be calculated as Max-Allocation.

(ii) Yes, it is safe as applying safety algorithm it leaves the system in safe state.

(iii) Yes, it will be granted as applying Banker's algo it leaves the system in safe state.

Q.6 a. Why the interrupt disable method to achieve mutual exclusion does not work in a multiprocessor system? Write your answer citing an example.

Answer:

Disabling the interrupt may cause the system to collapse if the process forgets to enable it after processing in the critical section. This is more so in Multiprocessor OS as there are many processes from many processors.

Q.7 Write short notes on the following:

(i) Page replacement methods

Answer:

It is when physical memory is full and cannot accommodate any other pages from the existing processes. There are many page replacement policy. FIFO, OPT, LRU to name a few.

(ii) Multimedia system

Answer:

Realtime system emphasizes over the jobs that have time constraints. There are two types of realtime system; hard real time and soft real time system. Multimedia system considers about the GUI aspect of the system.

(iii) Bernstein's condition for concurrency

Answer:

First define read ser R and write set W for the processes. For the two processes let it be R(P1), W(P1) and R(P2), W(P2). Bernstein's condition are:

I. $R(P1) \cap W(P2) = \varphi$
II. $R(P2) \cap W(P1) = \varphi$
III. $W(P1) \cap W(P2) = \varphi$

Text Books

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne (2010), "Operating System Principles" Johnwiley & Sons (Asia) Pte Ltd

2. Andrew S Tanenbaum "Modern Operating Systems" (2009) Pearson Education