CS & IT ENGINEERING

Operating Systems

1500 Series

Lecture No.- 03

By- Dr. Khaleel Khan sir

Recap of Previous Lecture

Topic

Questions Practice

Topics to be Covered

- Topic
- Topic
- Topic
- Topic
- Topic

Priority Scheduling w/Round-Robin

```
#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>
int main()
{ pid_t pid;
        /* fork a child process */
    1)  pid = fork();

if (pid < 0) { /* error occurred */
        fprintf(stderr, "Fork Failed");
        return 1;
}
else if (pid == 0) { /* child process */
    2)  execlp("/bin/ls", "ls", NULL);
```

Path — Program

```
}
else { /* parent process */
        /* parent will wait for the child to
complete */
    3)  wait (NULL);
        printf("Child Complete");
}
return 0;
}
```

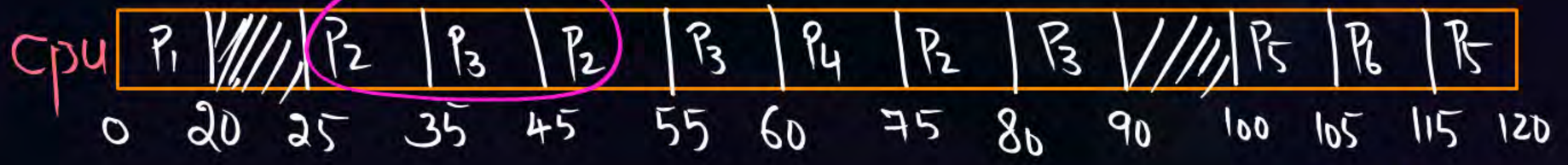
[NAT]

TQ = 10; Preemptive Priority - R.R :

#Q. The following processes are being scheduled using a preemptive, Round-Robin scheduling algorithm.

Process	Priority	Burst	Arrival
P_1	40	20	0
P_2	30	25	25
P_3	30	25	30
P_4	35	15	60
P_5	5	10	100
P_6	10	10	105

R.Q $P_1; P_2\ P_3; P_2; P_3; P_2; P_4; P_3; P_5; P_6; P_5$



Continues

Each process is assigned a numerical priority, with a higher number indicating a higher relative priority. In addition to the processes listed below, the system also has an idle task (which consumes no CPU resources and is identified as P_{idle}). This task has priority 0 and is scheduled whenever the system has no other available processes to run. The length of a time quantum is 10 units. If a process is preempted by a higher-priority process, the preempted process is placed at the end of the queue.

a. Show the scheduling order of the processes using a Gantt chart.

b. What is the turnaround time for each process?

c. What is the waiting time for each process?

d. What is the CPU utilization rate?

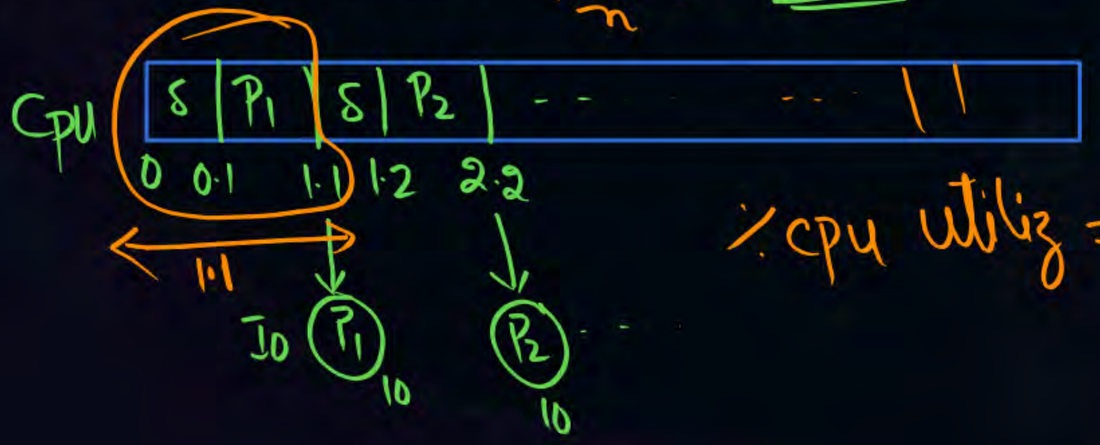
[NAT]

#Q. Consider a system running ten I/O-bound task and one CPU-bound task. Assume that the I/O-bound task issue on I/O operation once for every millisecond of CPU computing and that each I/O operation takes 10 milliseconds to complete. Also assume that the context-switching overhead is 0.1 millisecond and that all processes are long-running tasks. Describe the CPU utilization for a round-robin scheduler when:

(a)The time quantum is 1 millisecond ⟹ 91%

(b)The time quantum is 10 milliseconds ⟹ 94.7%

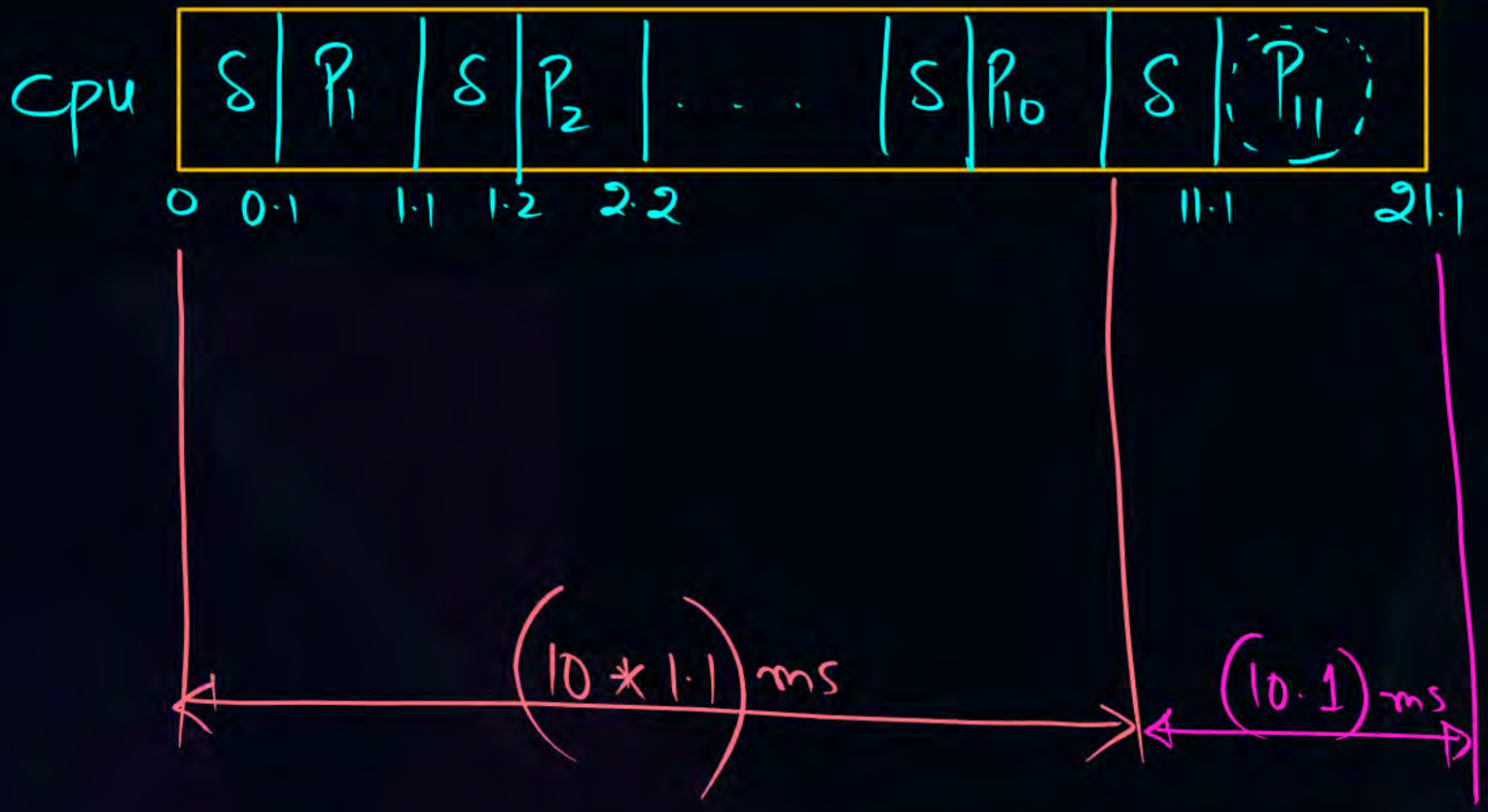
$T_0 = 1\,ms$; $\delta = 0.1\,ms$



% cpu utiliz = $\frac{1}{1.1}$ = 91%

TQ = 10 ms; S = 0.1 ms

RQ | P_1 - - P_{10} ; (P_{11}) → cpu



% cpu utiliz = $\frac{20}{21.1}$

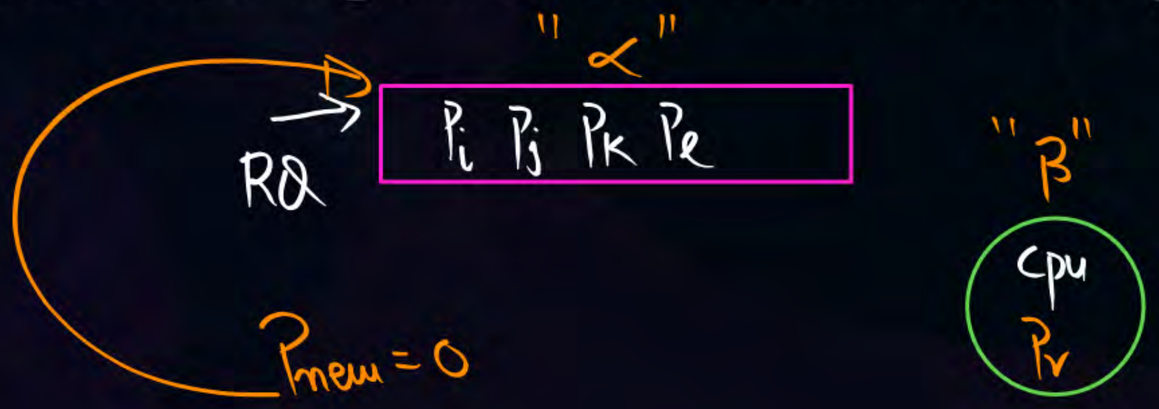
= 94.7 %

[NAT]

#Q. Consider a preemptive priority scheduling algorithm based on dynamically changing priorities. Larger priority numbers imply higher priority. When a process is waiting for the CPU (in the ready queue, but not running), its priority changes at a rate α. When it is running, its priority changes at a rate β. All processes are given a priority of 0 when they enter the ready queue. The parameters α and β can be set to give many different scheduling algorithms.

(a) What is the algorithm that results from $\beta > \alpha > 0$? FCFS

(b) What is the algorithm that results from $\alpha < \beta < 0$? LCFS



[MCQ]

#Q. A multiprogramming system may be defined as one in which:

A. Programs are divided into pages.

B. Input is accepted in batches of many jobs.

C. Several programs can reside in memory at the same time.

D. Many processes may share the same program residing in main memory.

[MCQ]

#Q. The main distinction between a multiprocessor system and a multiprogrammed system is that in a multiprocessor system:

A. The main storage is shared by several programs.

B. The input is accepted in batches of many jobs.

C. Processor time is shared among several processes

D. Many processors may be active simultaneously.

[MSQ]

#Q. A user process can become blocked only if it is:

<A ; B>

A. In the ready state.

B. In the running state.

C. In the blocked (or waiting) state.

D. Waiting for a resource.

Process is already in Block State

[MSQ]

#Q. Which of the following action may result in a process becoming blocked?

<A C>

A. A process executed a P(wait) operation on a semaphore

B. A process executes a V(signal) operation on a semaphore.

C. A process activating a System Call.

D. A process within a critical section changes the value of a shared variable.

[MSQ]

#Q. Non-Preemptive Process-scheduling policies: <C ; D>

unavoidable / compulsory

A ✗ Are indispensable for interactive systems.

B ✗ Allocate the processor to a process for a fixed time period.

C ✓ Will have same Average Waiting Time and Response Time.

D ✓ Make short jobs wait for long jobs.

[MCQ]

#Q. The pure-Round-Robin-scheduling policy:

A. Responds poorly to short processes if the time slice is small.

B. Does not use any a priori information about the service times of processes.

C. Becomes equivalent to the Shortest-Job-First Policy when the time slice is made infinitely large.

[MSQ]

#Q. Which of the following statements is/are TRUE?

A. I/O-bound processes should be given priority in scheduling over CPU-bound processes to ensure good turnaround time.

B. Users can exploit a multilevel feedback- scheduling policy by breaking a long job into several small jobs.

C. The processor scheduler normally classifies a process as being a CPU-bound process if it uses most of the previous time slice allocated to it.

D. The Round-Robin-scheduling policy allocates a time slice to a process depending on the number of time slices it has already used.

[MCQ]

#Q. A counting semaphore was initialized to 9. Then 27P (wait) operations and 23 V (signal) operations were completed on this semaphore. The resulting value of the semaphore is:

A 5

B 0

C 17

D 13

$9 - 27 + 23 = 5$

[MSQ]

<A + C>

#Q. The main difference between binary semaphores and counting semaphores is that:

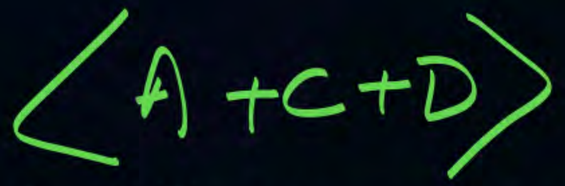
A. Binary semaphores can only take the values 0 and 1, while counting semaphores can take any integer values.

B. Binary semaphores can only be used to solve problems involving up to two processes sharing the same resource, while counting semaphores can be used to solve problems involving more than two processes sharing the same resource.

C. Binary semaphores can solve all the problems that can be solved by counting semaphores.

D. Counting semaphores must be controlled by a monitor, while binary semaphores are called directly by user processes.

[MSQ]



#Q. Which of the following statements is/are TRUE?

A. Disjoint processes need not use critical section

B. Processes with critical sections can never be pre-empted while executing critical section.

C. A process wanting to enter a critical section currently in use must wait for the process utilizing the critical section to Leave it.

D. Two different critical sections may be executed concurrently if they do not use the same shared variables.

[MSQ]

<C+D>

#Q. The basic principle of a Monitor is that:

A. Several resource can only be controlled by a monitor.

B. Several processes may concurrently execute a procedure of a given monitor.

C. Only one process may execute a procedure of a given monitor at any given time.

D. Condition Variables of Monitor are never associated with State. Value.

Q# Two Concurrent Processes P1 and P2 want to use two resources R1 and R2 in a mutually exclusive manner. Initially RI and R2 are free. The programs executed by the two processes are given below.

Program for P1:

```
SI:  while (R1 is busy) do no-op;
S2:     Set    R1← busy;
S3:  while (R2 is busy) do no-op;
S4:     Set    R2 ← busy;
S5:  ⟨use  R1 and R2;⟩
S6:  set  R1 ← free;
S7:  set  R2 ← free;
```

Entry

CS

Exit

Contd..

Program for P2:

```
Q1:  while (R2 is busy) do no-op;
Q2:     Set R2 ← busy;
Q3:  while (RI is busy) do no-op;
Q4:     Set  RI ← busy;
Q5:  use   RI  and R2;
Q6:  Set   R2  ←  free;
Q7:  Set   RI  ←  free;
```

Entry

CS

Exit

(a) Is a mutual exclusion guaranteed for R1 and R2? If not, show a possible interleaving of the statements of P1 and P2 such that mutual exclusion is violated (i.e., both P1 and P2 use R1 or R2 at the same time)

(b) Can Deadlock occur in the above program? If yes, show a possible interleaving of the statements of P1 and P2 leading to Deadlock.

(c) Exchange the statements Q1 and Q3 and statements Q2 and Q4. Is mutual exclusion guaranteed now? Can Deadlock occur?

M.E : Violated

Deadlock : Never

[MCQ]

M.E + H&W + NO PreEmp + C.W

#Q. Which of the following is not a necessary condition for a deadlock?

A. Mutually exclusive use of a resource by processes.

B. Partial allocation of resources to a process. (H&W)

C. Preemptive scheduling of resources.

D. Circular waiting by processes.

[MSQ]

#Q. Solutions to the Dining Philosophers problem which avoids Deadlock is:

A. Non-preemptive CPU scheduling.

B. Have one additional fork than the number of Philosophers.

C. Ensuring that first (n-1) philosophers pick up their right fork before they pick up their left fork and the last one in opposite direction.

D. Ensuring that odd philosophers pick up their left fork before they pick up their right fork and even philosophers pick up their right fork before they pick up their left fork.

[NAT]

#Q. Consider the following snapshot of a system:

	Allocation	Max	Available	Need
	A B C D	A B C D	A B C D	A B C D
P0	0 0 1 2	0 0 1 2	1 5 2 0	0 0 0 0
P1	1 0 0 0	1 7 5 0		0 7 5 0
P2	1 3 5 4	2 3 5 6		1 0 0 2
P3	0 6 3 2	0 6 5 2		0 0 2 0
P4	0 0 1 4	0 6 5 6		0 6 4 2

Answer the following questions using the Banker's algorithm:

(a) What is the content of the matrix Need?

(b) Is the system in a safe state? SAFE

(c) If a request from process P1 arrives for (0, 4, 2, 0), can the request be granted immediately? granted ✓

[NAT]

#Q. Consider a system consisting of four resources of the same type that are shared by three processes, each of which needs at most two resources. Is the system Deadlock free ?

YES

[MSQ]

#Q. Which of the following statements is/are FALSE for the Banker's algorithm?

A. It can be used for a system with many resources, each of which is unique with no multiple copies. T

B. It is used to detect deadlock F

C. It is not applicable when a resource is shared simultaneously by many users. T

D. An unsafe situation will always lead to a deadlock. F

[NAT]

#Q. Consider the following snapshot of a system:

	Allocation	Max
	ABCD	ABCD
T_0	3014	5117
T_1	2210	3211
T_2	3121	3321
T_3	0510	4612
T_4	4212	6325

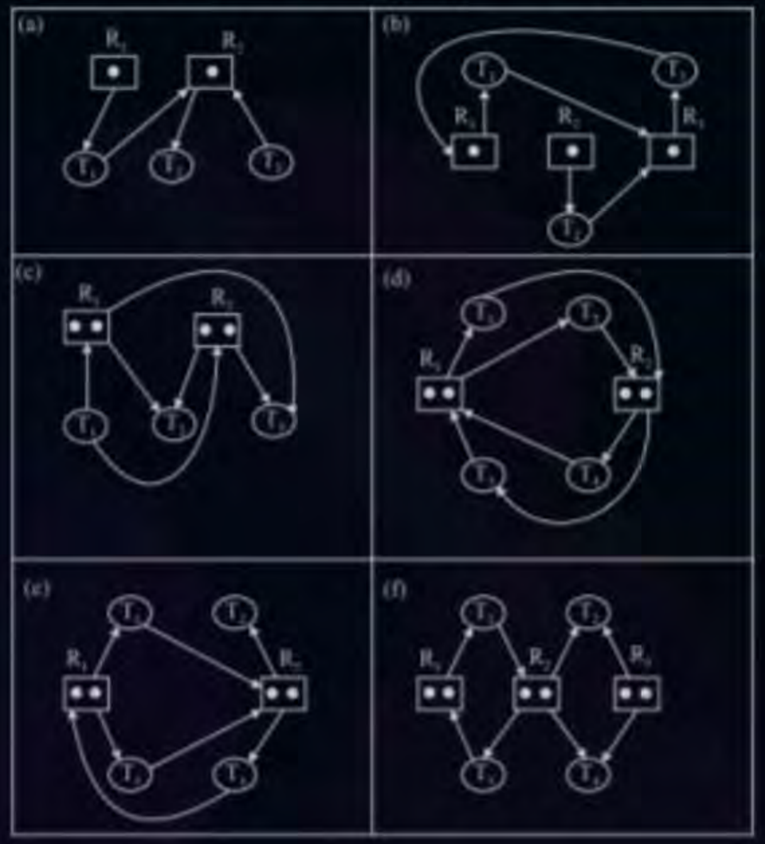
Using the banker's algorithm, determine whether or not each of the following states is unsafe. If the state is safe, illustrate the order in which the threads may complete. Otherwise, illustrate why the state is unsafe.

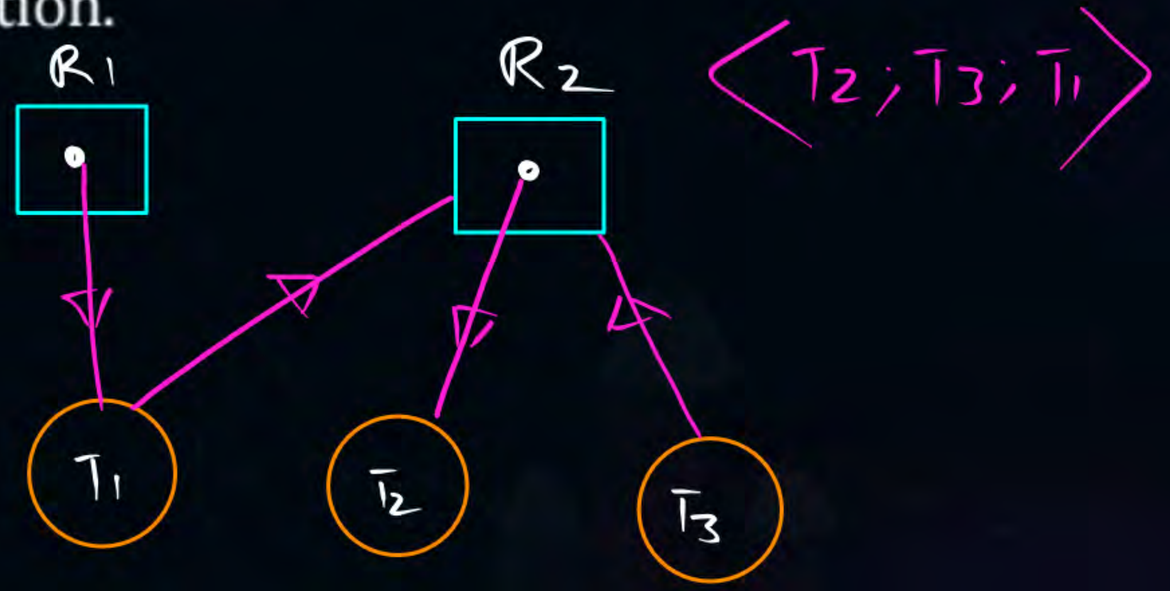
a. Available = (0,3,0,1)

b. Available = (1,0,0,2)

[NAT]

#Q. Which of the six resource-allocation graphs shown illustrate Deadlock? For those situations that are deadlocked, provide the cycle of threads and resources. Where there is not a deadlock situation, illustrate the order in which the threads may complete execution.





[NAT]

H/W

#Q. Consider the version of the Dining-Philosophers problem in which the chopsticks are placed at the center of the table and any two of them can be used by a philosopher. Assume that requests for chopsticks are made one at a time. Describe a simple rule for determining whether a particular request can be satisfied without causing deadlock given the current allocation of chopsticks to philosophers.

[NAT]

H/w

#Q. A single-lane bridge connects the two Vermont villages of North Tunbridge and South Tunbridge. Farmers in the two villages use this bridge to deliver their produce to the neighboring town. The bridge can become deadlocked if a northbound and a southbound farmer get on the bridge at the same time. (Vermont farmers are stubborn and are unable to back up.) Using semaphores and/or mutex locks, design an algorithm in pseudocode that prevents deadlock. Initially, do not be concerned about starvation (the situation in which northbound farmers prevent southbound farmers from using the bridge, or vice versa).

[NAT]

#Q. The following is a question about Dining Computer Scientists. There are 6 computer scientists seated at a circular table. There are 3 knives at the table and 3 forks. The knives and forks are placed alternately between the computer scientists. A large bowl of food is placed at the center of the table. The computer scientists are quite hungry, but require both a fork and knife to eat.

Consider the following policies for eating and indicate, if it can result in Deadlock.

<No - Deadlock occurs>

- Attempt to grab the fork that sits between you and your neighbor until you are successful.
- Attempt to grab the knife that sits between you and your neighbor until you are successful.
- Eat
- Return the fork
- Return the knife

[MCQ]

[GATE : 2021]

#Q. A RAM chip has a capacity of 1024 words of 8 bits each (1K × 8). The number of 2 × 4 decoders with enable line needed to construct a 16K × 16 RAM from 1K × 8 RAM is

A 4

B 5

C 6

D 7

[MCQ]

[GATE : 2009]

#Q. How many 32K × 1 RAM chips are needed to provide a memory capacity of 256K bytes?

A 8

B 32

C 64

D 128

[NAT]

M.F.T

#Q. Given six memory partitions of 300 KB, 600 KB, 350 KB, 200 KB, 750 KB, and 125 KB (in order), how would the first-fit, best-fit, and worst-fit algorithms place processes of size 115 KB, 500 KB, 358 KB, 200 KB, and 375 KB (in order)? Rank the algorithms in terms of how efficiently they use memory.

(i) First Fit ⟶ F.F

(ii) B.F ⟶

(iii) W.F ⟶

2 mins Summary

Topic One

Priority Scheduling w/Round-Robin

Topic Two

Topic Three

Topic Four

Topic Five

THANK - YOU