CS & IT ENGINEERING

Operating Systems

1500 Series

Lecture No.- 01

By- Dr. Khaleel Khan sir

Recap of Previous Lecture

Topic

Topics to be Covered

- Topic
- Topic
- Topic
- Topic
- Topic

CPU-scheduling

[MSQ]

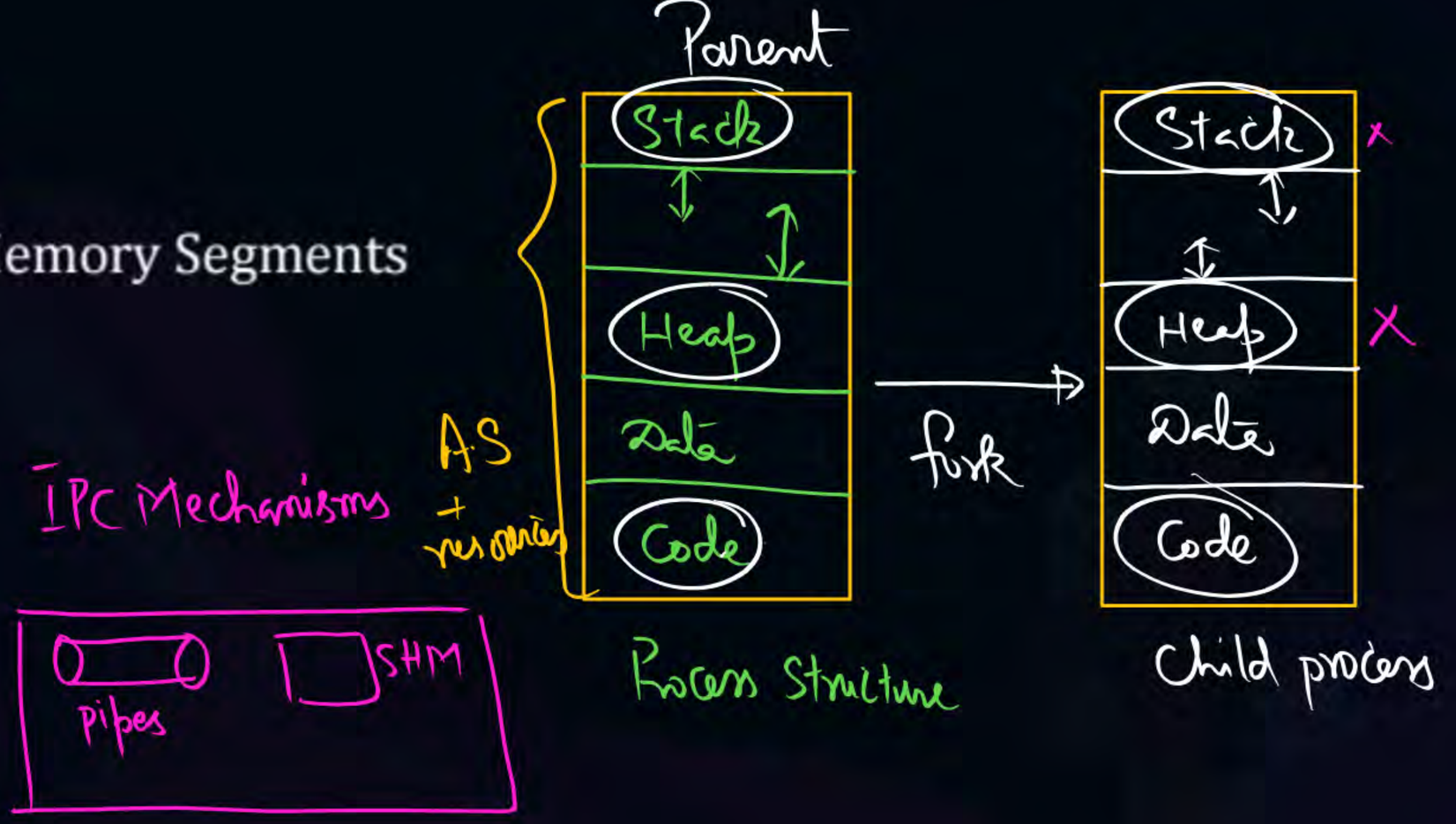
#Q. When a process creates a new process using the fork() operation, which of the following states is shared between the parent process and the child process?

A. Stack ×

B. Heap ×

C. Shared Memory Segments ✓

D. Pipes ✓

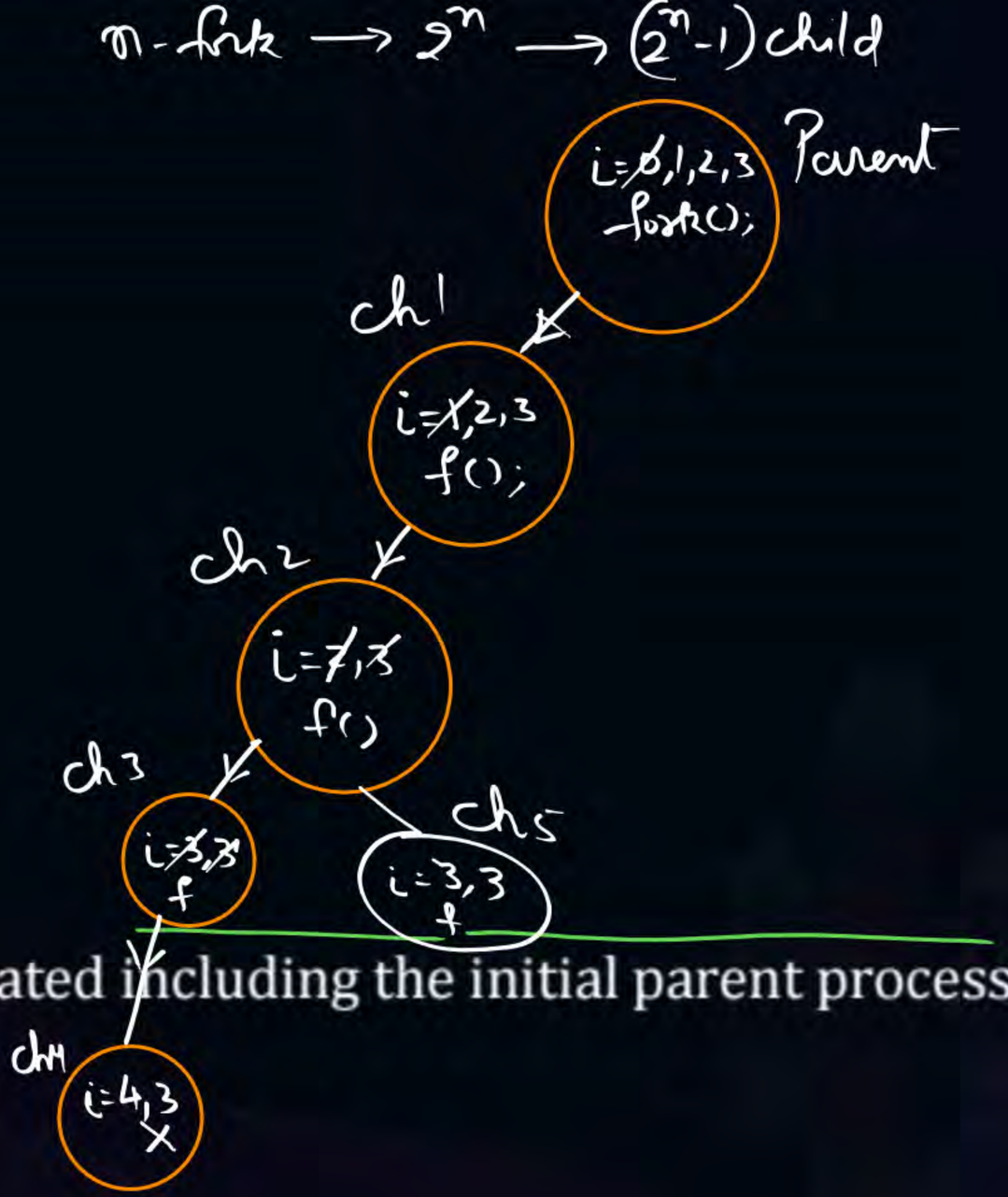


[NAT]

#Q.

```
#include <stdio.h>
#include <unistd.h>
int main()
{
        int i;
        for (i = 0; i < 4; i++)
                fork();
        return 0;
}
```

How many processes are created including the initial parent process?



[MSQ]

#Q. Which of the following components of program state are NOT shared across Threads in a Multithreaded process?

A Register values

B Heap memory

C Global variables

D Stack memory

Shared

Shared

<A ; D>

Thread

Stack + Register

[MCQ]

#Q. What system calls have to be executed by a command interpreter or shell in order to start a new process, which should execute an application?

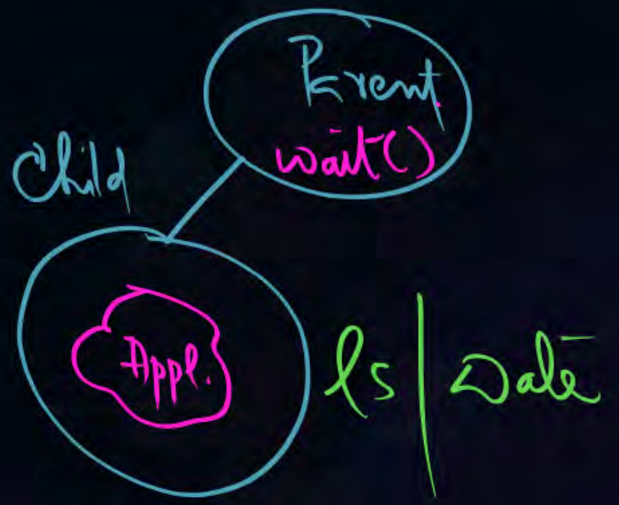
a) fork

b) fork, exec, getpid

c) fork, exec, wait

d) fork, exec, wait, exit

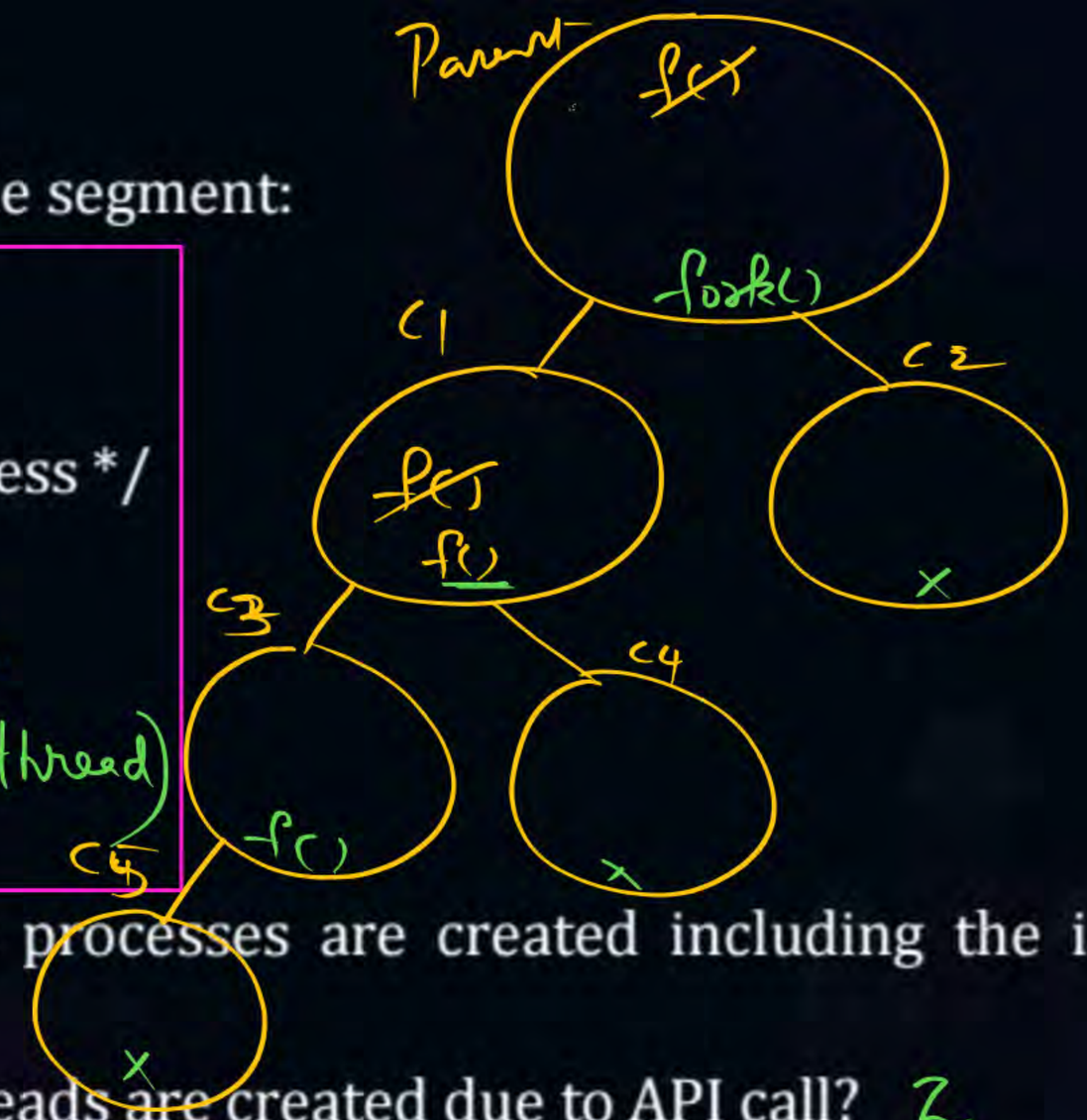
exec : is a System call to Execute a Command/ablf. → run a new Program



[NAT]

#Q. Consider the following code segment:

```
pid_t pid;
pid = fork();
if (pid == 0) { /* child process */
        fork();
        thread_create( . . . );
}
fork();
```



(a) How many unique processes are created including the initial parent process? 6

(b) How many new threads are created due to API call? 2

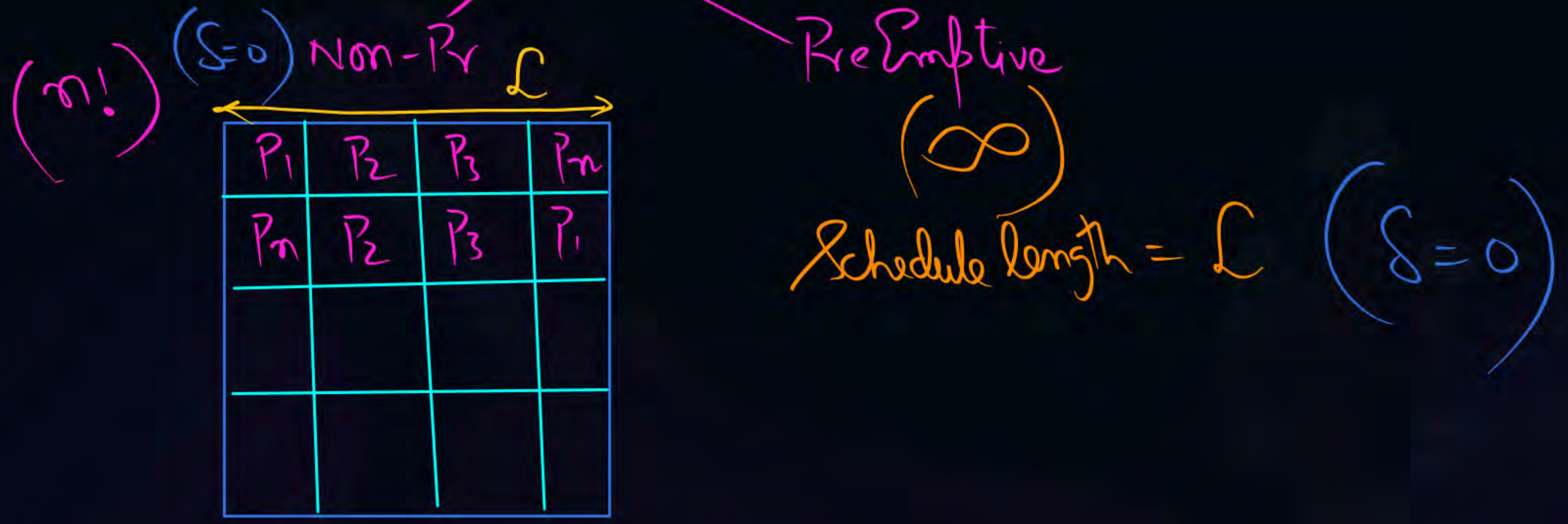
[NAT]

#Q. Which of the following instructions should be privileged?

a. Set value of timer.
b. Read the clock.
c. Clear memory.
d. Issue a trap instruction.
e. Turn off interrupts.
f. Modify entries in device-status table.
g. Switch from user to kernel mode.
h. Access I/O device

[NAT]

#Q. 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 schedules are possible? Give a formula in terms of n.



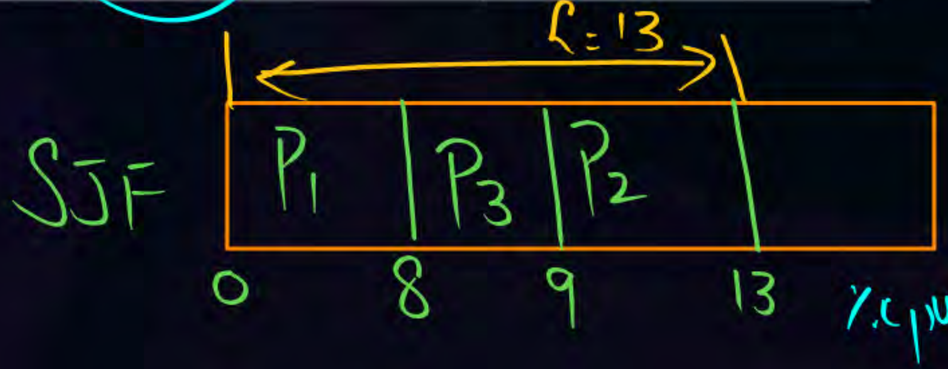
[NAT]

#Q. Suppose that the following processes arrive for execution at the times indicated. Each process will run for the amount of time listed. In answering the questions, use non-preemptive scheduling, and base all decisions on the information you have at the time the decision must be made.

Process	Arrival Time	Burst Time
P1	0.0	8
P2	0.4	4
P3	1.0	1

FCFS:

Av. TAT = 10.53
Av. RT = 6.86
} Av. wt = 6.86



CPU Idleness = 0

S.J.F

Av. TAT = 9.53

Av RT = $\frac{7+8.6}{3} = \frac{15.6}{3} = 5.2$

Av. wt = 5.2

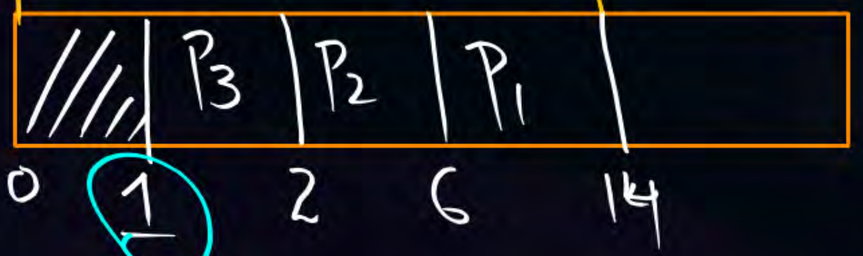
(a) What is the average Turnaround Time and Response Time for these processes with the FCFS scheduling algorithm?

(b) What is the average turnaround time and Response Time for these processes with the SJF Scheduling algorithm?

(c) The SJF algorithm is supposed to improve performance, but notice that we chose to run process P_1 at time 0 because we did not know that two shorter processes would arrive soon. Compute what the average TAT and WT will be if the CPU is left idle for the first 1 unit and then SJF Scheduling is used. Remember that processes P_1 and P_2 are waiting during this idle time, so their waiting time may increase. This algorithm could be known as Future-Knowledge Scheduling. ⟨New-SJF⟩

$Av.R.T = \frac{6 + 1.6 + 0}{3} = \frac{7.6}{3} = 2.5$ ✓

$L = 14$



$Av.TAT = \frac{14 + 5.6 + 1}{3} = \frac{20.6}{3} = 6.8$

$\%CPU\ Idleness = \frac{1}{14}$

$L = Max(CT) - Min(AT)$

$14 - 0 = 14$

Priority Scheduling w/Round-Robin

Hybrid

11:30

- Run the process with the highest priority. Processes with the same Priority run Round-Robin
- Example:

Process	Burst Time	Priority
P_1	4	3 (L)
P_2	5	2
P_3	8	2
P_4	7	1 (H)
P_5	3	3

R.Q: P_1 -- P_5

TAT = 20.8

WT = 15.4

R.T = 4

$= \frac{58}{5} = 11.6$

- Time quantum is 2. Draw the Gantt Chart and Compare Avg. TAT and RT with Pure Round Robin.

CPU

Av. WT = 13.8

Av. TAT = 19.2

Av. RT = $\frac{(20+7+9+0+22)}{5}$

2 mins Summary

Topic One **Priority Scheduling w/Round-Robin**

Topic Two

Topic Three

Topic Four

Topic Five

THANK - YOU

CS & IT ENGINEERING

Operating Systems

1500 Series

Lecture No.- 02

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

Priority Scheduling w/Round-Robin

Hybrid

11:30

- Run the process with the highest priority. Processes with the same Priority run Round-Robin
- Example:

Process	Burst Time	Priority
P_1	4	3 (L)
P_2	5	2
P_3	8	2
P_4	7	1 (H)
P_5	3	3

R.Q: P_1 -- P_5

TAT = 20.8

WT = 15.4

R.T = 4

$= \frac{58}{5} = 11.6$

- Time quantum is 2. Draw the Gantt Chart and Compare Avg. TAT and RT with Pure Round Robin.

CPU

Av. WT = 13.8

Av. TAT = 19.2

Av. RT = $\frac{(20+7+9+0+22)}{5}$

[NAT]

TQ = 10 TAT

#Q. The following processes are being scheduled using a preemptive, priority-based, Round-Robin scheduling algorithm. <Hybrid>

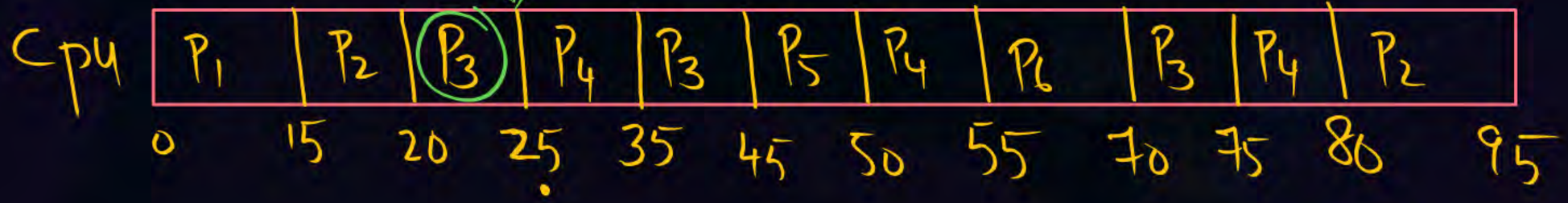
a) 39.17 b) 41.6
c) 34.16 d) 40
e) 38.33 f) 45

Process	Priority	Burst	Arrival
P_1	8	15	0
P_2	3	20	0
P_3	4	20	20
P_4	4	20	25
P_5	5	5	45
P_6	5	15	55

Av. TAT
Av. R.T

P1, P2, P3, P2, P4 P3 P3, P4, P5, P3, P4, P6

Pre



Continues

Each process is assigned a numerical priority, with a higher number indicating a higher relative priority. The scheduler will execute the highest-priority process. For processes with the same priority, a Round-Robin scheduler will be used with a time quantum of 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?

[NAT]

TQ = 10

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

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

R.Q

P_1 ; P_2 ; P_3 ; P_2 ; P_3 ; P_4 ; P_3 ; P_2

CPU

P_1	P_{idle}	P_2	P_3	P_2	P_3	P_4	P_2	P_3	P_{idle}	P_5	P_6	P_5

0 20 (5) 25 30 40 50 60 75 85 90 (10) 100 105 115 120

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?

→ 100 - 12.5 %

∴ cpu Idleness = $\frac{15}{120} = \frac{3}{24} = \frac{1}{8} = 0.125 = 12.5\%$

[NAT]

M.L.

#Q. A variation of the Round-Robin scheduler is the Regressive Round-Robin scheduler. This scheduler assigns each process a time quantum and a priority. The initial value of a time quantum is 50 milliseconds. However, every time a process has been allocated the CPU and uses its entire time quantum (does not block for I/O), 10 milliseconds is added to its time quantum, and its priority level is boosted. (The time quantum for a process can be increased to a maximum of 100 milliseconds.) When a process blocks before using its entire time quantum, its time quantum is reduced by 5 milliseconds, but its priority remains the same. What type of process (CPU-bound or I/O-bound) does the Regressive Round-Robin scheduler favor?

<CPU-Bound>

[MSQ]

#Q. Which of the following scheduling algorithms could result in starvation?

<B, D>

A First-come, First-served

B Shortest Job First

C Round Robin

D Priority

[NAT]

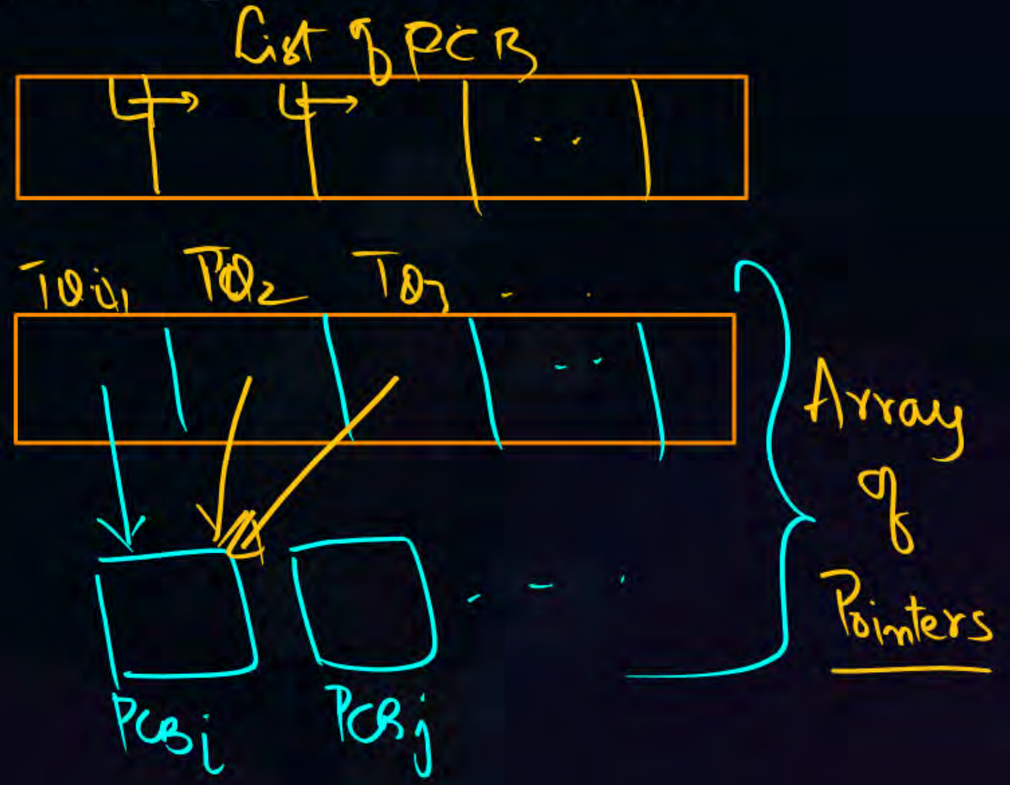
#Q. Consider a variant of the RR scheduling algorithm in which the entries in the ready queue are pointers to the PCBs.

(a) What would be the effect of putting two pointers to the same process in the ready queue? (Prioritising the Process)

(b) What would be major advantages and disadvantages of this scheme?

a) Improve Priority of desired Process

Starv. Lock of Resp.



[MCQ]

Definition

#Q. A process may be defined as:

A. A set of instructions to be executed by a computer.

B. A program in execution.

C. A piece of hardware that executes a set of instructions.

D. The main procedure of a program.

[MCQ]

#Q. A processor in the context of computing is:

GPU

A. A set of instructions to be executed on a computer.

B. A program in execution.

C. A piece of hardware that executes a set of instructions.

D. The main procedure of a program.

2 mins Summary

- Topic One
- Topic Two
- Topic Three
- Topic Four
- Topic Five

THANK - YOU

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;
}
```

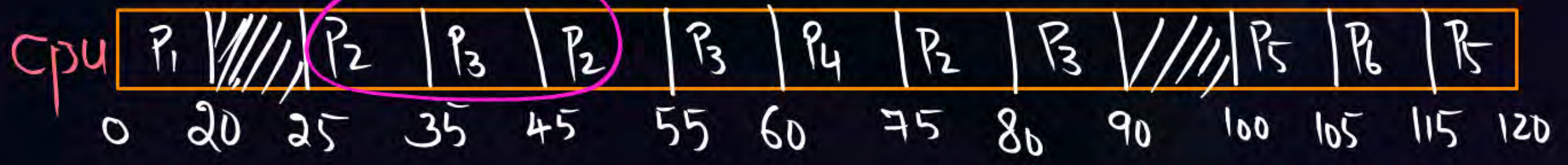
[NAT]

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?

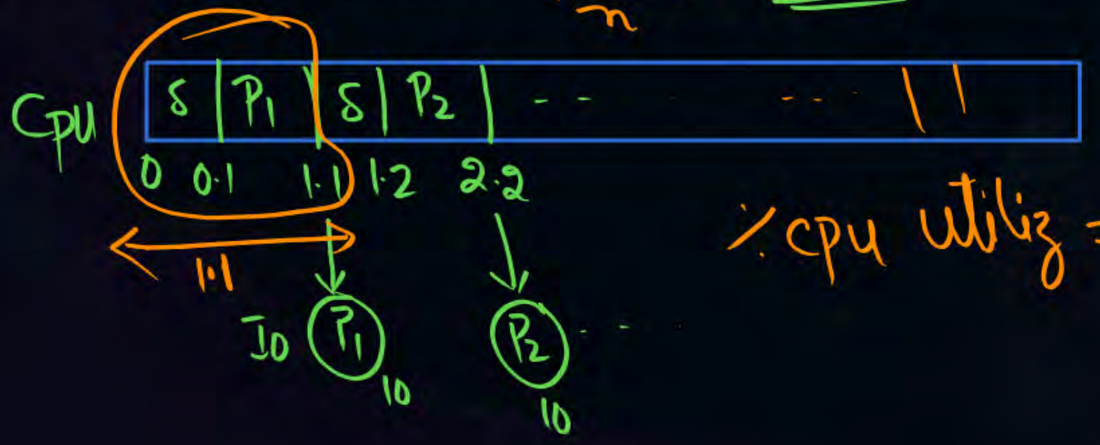
[NAT]

#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%

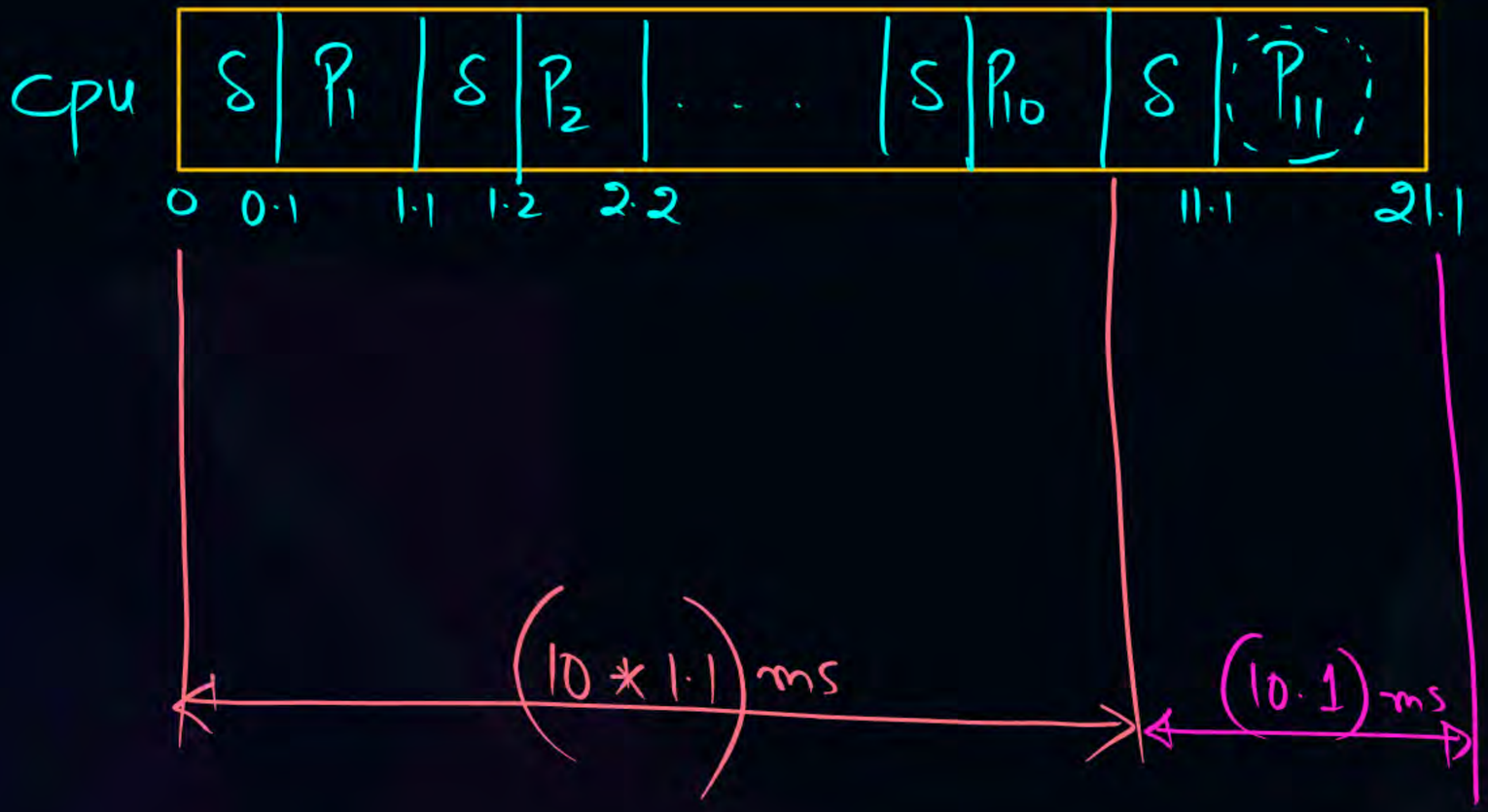
$T_0 = 1\,ms$; $\delta = 0.1\,ms$



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

TQ = 10 ms; S = 0.1 ms

RQ | P_1 - - P_{10} ; (P_{11}) → cpu



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

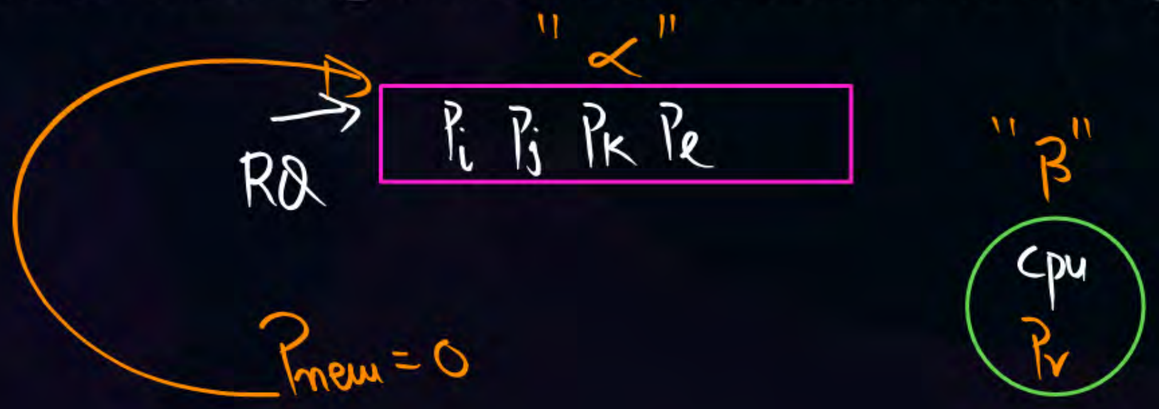
= 94.7 %

[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:

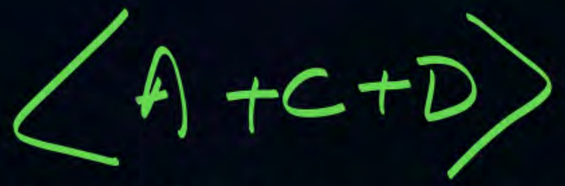
A. Binary semaphores can only take the values 0 and 1, while counting semaphores can take any integer values.

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

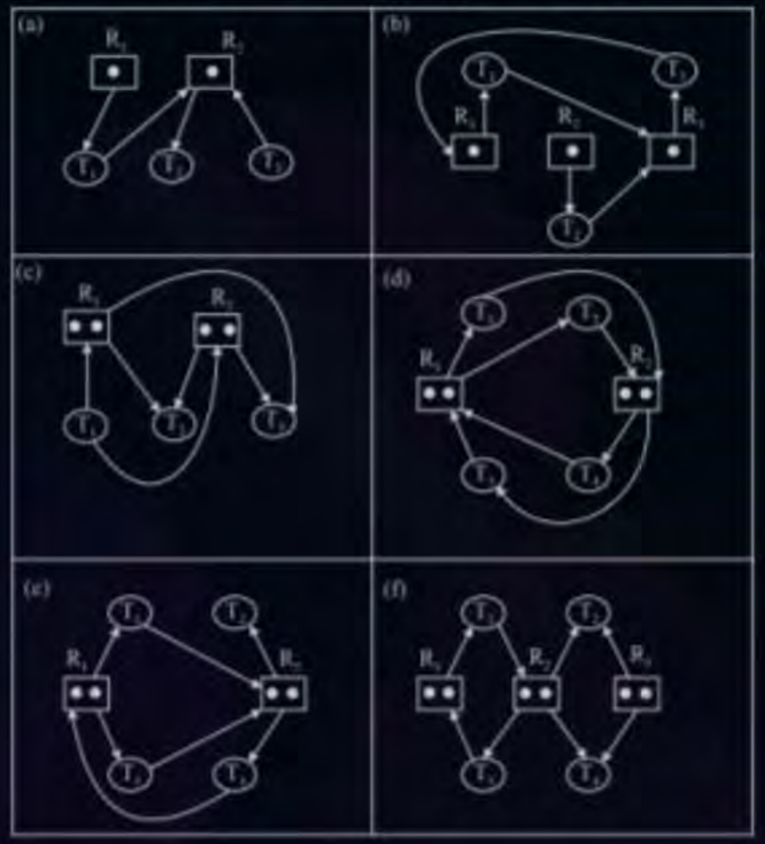
Using the banker's algorithm, determine whether or not each of the following states is unsafe. If the state is safe, illustrate the order in which the threads may complete. Otherwise, illustrate why the state is unsafe.

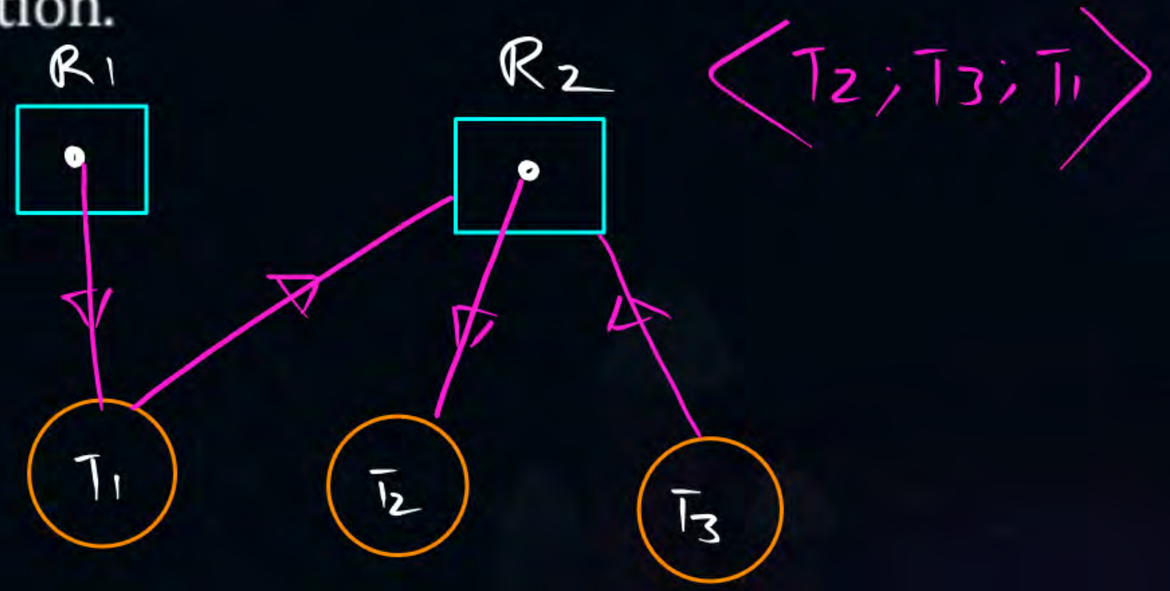
a. Available = (0,3,0,1)

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

CS & IT ENGINEERING

Operating Systems

1500 Series

Lecture No.- 04

By- Dr. Khaleel Khan sir

Recap of Previous Lecture

Topic

Questions Practice

Topics to be Covered

- Topic
- Topic
- Topic
- Topic
- Topic

Dining-Philosophers

[NAT]

#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.

"Do not grant the request, if there is no other Philosopher with 2 chopsticks and if there is only chopstick remaining"

[NAT]

#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).


```
BSEM mutex=1;
Acquire-Bridge()
{
    P(mutex);
}
<Use_Bridge>
Release-Bridge()
{
    V(mutex);
}
```

[NAT]

#Q. Consider a logical address space of 64 pages of 1,024 words each, mapped onto a physical memory of 32 frames.

(a) How many bits are there in the logical address? 16 bits

(b) How many bits are there in the physical address? 15 bits

$N = 64$; $P.S = 1024$; $M = 32$

$L.A.S = 64 \times 1KW = 64KW$

$L.A = 16$ bits

$P.A.S = 32KW$

$P.A = 15$ bits

[NAT]

1024 By

#Q. Assuming a 1-KB page size, what are the page numbers and offsets for the following address references (provided as decimal numbers):

(a) 3085 (b) 42095

(c) 215201 (d) 650000

(e) 2000001

$P = V.A / P.S$

$d = V.A \% P.S$

$V.A = 3085$

$P = 3085 / 1024 = 3$ ✓

$d = 3085 \% 1024 = 13$ ✓

[NAT]

#Q. The BTV operating system has a 21-bit virtual address, yet on certain embedded devices, it has only a 16-bit physical address. It also has a 2-KB page size. How many entries are there in each of the following?

(a) A conventional, single-level page table : $2^{21}/2^{11} = 2^{10} = 1024$ ✓

(b) An inverted page table → $2^{16}/2^{11} = 2^{5} = 32$ ✓

[NAT]

#Q. Consider a paging system with the page table stored in memory.

(a) If a memory reference takes 50 nanoseconds, how long does a paged memory reference take? (2m) = 100 ns

(b) If we add TLBs, and 75 percent of all page-table references are found in the TLBs, what is the effective memory reference time? (Assume that finding a page-table entry in the TLBs takes 2 nanoseconds, if the entry is present.)

$$EMAT = \alpha(c+m) + (1-\alpha)(c+2m)$$

$$EMAT_{TLB} = 0.75(2+50) + 0.25(2+100)$$

$$= 64.5$$

$c \sim 0$

[NAT]

#Q. Consider the following segment table:

Segment	Base	Length
0	219	600
1	2300	14
2	90	100
3	1327	580
4	1952	96

219+430

What are the physical addresses for the following logical addresses?

(a) 0,430 (b) 1,10 ⇒ 2300+10

(c) 2,500 ✗ (d) 3,400 ⇒ 1327+400

(e) 4,112 ✗

[MCQ]

<Deadlock + MM>

#Q. Consider a system in which the total available memory is 48K and in which memory once allocated to a process cannot be preempted from that process. Three processes A, B, and C have declared in advance that the maximum amount of memory that they will require is 25K, 15K, and 41K words respectively. When the three processes are all in execution and using 3K, 9K, and 24K words of memory respectively, which one of the following requests for additional allocation can be granted with a guarantee that deadlock will not occur as a result of the allocation.

A. A requests 9K words.

B. C requests 7K words.

C. B requests 6K words.

D. A requests 6K words.

Total = 48K

	Max	Alloc	Need	Avail
A	25K	3K	22K	12K
B	15K	9K	6K	
C	41K	24K	17K	
		Total 36K		

#Q. Suppose the page table for the process currently executing on the processor looks like the following. All numbers are decimal, everything is numbered starting from zero, and all addresses are memory byte addresses. The page size is 1,024 bytes.

Virtual page number	Valid bit	Reference bit	Modify bit	Page frame number
0	1	1	0	4
1	1	1	1	7
2	0	0	0	-
3	1	0	0	2
4	0	0	0	-
5	1	0	1	0

a. Describe exactly how, in general, a virtual address generated by the CPU is translated into a physical main memory address.

b. What physical address, if any, would each of the following virtual addresses correspond to? (Do not try to handle any page faults, if any.)
(i) 1,052
(ii) 2,221
(iii) 5,499

V.A = | P | d

(i) 1,052

P, d

P.A = (7 * 1024) + 52

= 7168
52
7220

(7220) / 1024 = 7 f

d = 52

P.A: | f = 7 | d = 52 |

#Q. Assuming a page size of 4 Kbytes and that a page table entry takes 4 bytes, how many levels of page tables would be required to map a 64-bit address space, if the top-level page table fits into a single page?

Condition for O.P.T Size to fit in one Page-frame

$V.A.S = 2^{S}$ By $\Rightarrow 2^{64}$

$P.S = 2^{x}$ By $\Rightarrow 2^{12}$

$PTE = 2^{c}$ Bytes $\Rightarrow 2^{2}$

No. of levels of Paging $= l$

$$\Rightarrow \left[2^{S-l\cdot x+l\cdot c}\right] = 2^{x}$$

$$S - l\cdot x + l\cdot c = x$$

$$64 - l\cdot 12 + l\cdot 2 = 12$$

$\therefore l = 6$ ✓

#Q. Assume in a computer supporting paging, virtual addresses are 36 bits long. Page size is 16 KB. Assume a page table entry is 8 bytes long. The computer has 4 GB of RAM (physical memory).

We have three processes (A, B, C) in the system at the moment. The virtual memory (VM) layouts of the processes are like the following :

- A is using 64 MB from the start of its VM and 32 MB from the end. The rest of its VM is unused.
- B is using 128 MB from the start and 64 MB from the end. The rest of its VM is unused.
- C is using 32 MB from the start and 32 MB from the end. The rest of its VM is unused.

What will be the amount of physical memory required (in K Bytes) to hold the page table information for these three processes in total, if two-level page table is used with address division scheme (11,11,14)?

P.S = 16KB ; V.A = 36 bits , P.T.E = 8B ; PAS = 4GB

Process "A"

16KB

2 Pages

4K entries

1_level P.T

8B

P_0

P_1

P_n

4m entries

2K

P.T = 32MB

11 2K entries chunk

8B

0 ✓

1 ✓

n ✓

2K

11 bits

OPT

16KB + 3*16KB

4*16KB = 64KB ✓

V.A.S = 2^{36}

$\frac{2^{36}}{2^{12}} = 2^{22}$

4m

64MB

N = 64MB / 16KB

$= \frac{2^{26}}{2^{14}} = 2^{12}$

= 4K

32MB / 16KB

$= \frac{2^{25}}{2^{14}}$

$= 2^{11}$

= 2K ✓

#Q. Assume that a task is divided into four equal-sized segments and that the system builds an eight-entry page descriptor table for each segment. Thus, the system has a combination of segmentation and paging. Assume also that the page size is 2 Kbytes.

H/w

1. What is the maximum size of each segment?
2. What is the maximum logical address space for the task?
3. Assume that an element in physical location 00021 ABC is accessed by this task. What is the format of the logical address that the task generates for it? What is the maximum physical address space for the system?

#Q. A Virtual Memory has a page size IK words. There are eight pages and four blocks. The associative memory page table contains the following entries:

frame

Page	Block
0	3
1	1
4	2
6	0

H/W

Make a list of all virtual addresses (in decimal) that will cause a page fault if used by the CPU.

#Q. A virtual memory system has an address space of 8K words, a memory space of 4K words, and page and block sizes of IK words. The following page reference changes occur during a given time interval. (Only page changes are listed. If the same page is referenced again, it is not listed twice.)

4 2 0 1 2 6 1 4 0 1 0 2 3 5 7

Determine the four pages that are resident in main memory after each page reference change if the replacement algorithm used is

(a) FIFO (c) Optimal

(b) LRU

[MCQ]

#Q. In partition-memory management scheme, Internal Fragmentation occurs when:

A. Sufficient memory is available to run a program, but it is scattered between existing partitions.

B. Insufficient memory is available to run a program.

C. The partition allocated to a program is larger than the memory required by the program.

D. A program is larger than the size of memory on the computer.

[MCQ]

#Q. The FIFO page-replacement policy:

A. Is based on program locality.

B. Sometimes can cause more page faults when memory size is increased.

C. Is not easy to implement, and hence, most systems use an approximation of FIFO.

[MSQ]

<A + C>

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

A. With the Least Recently Used (LRU) page-replacement policy, when the page size is halved, the number of page faults can be more than double the original number of page faults. — F

B. The working set size is a monotonically nondecreasing (increasing) function of the working set parameter. — T

C. When the working set policy is used, main memory can never contain pages which do not belong to the working set of any program. — F

[MSQ]

#Q. It is advantageous for the page size to be large because:

A. Less unreferenced data will be loaded into memory.

B. Virtual addresses will be smaller.

C. Page tables will be smaller.

D. Leads to less page faults.

[MSQ]

#Q. It is advantageous for the page size to be small because:

A. Less unreferenced data will be loaded into memory.

B. Virtual addresses will be smaller.

C. Page tables will be smaller.

D. Leads to Less Internal Fragmentation.

[MCQ]

#Q. For a certain page trace starting with no page in the memory, a demand-paged memory system operated under the LRU replacement policy results in 9 and 11 page faults when the primary memory is of 6 and 4 pages, respectively. When the same page trace is operated under the optimal policy, the number of page faults may be:

A. 9 and 7.

B. 7 and 9.

C. 10 and 12.

D. 6 and 7.

L.R.U

7

(More than 6 Unique Pages)

6 Frames → 9 PF's

4 Frames → 11 PF's

Optimal

6 Frames →

4 Frames →

[MCQ]

#Q. In a Paged Segmented scheme of memory management, the segment table points to a page table because:

A. The segment table may occasionally be too large to fit in one page.

B. Each segment may be spread over a number of pages.

C. Page size is usually larger than the segment size.

D. The page table may be too large to fit into a single segment.

[MSQ]

#Q. Sharing in a paged memory system is done by:

A. Giving a copy of the shared pages to each process.

B. Dividing the program into procedures and data and allowing only the procedures to be shared.

C. Several page table entries pointing to the same frame in the main memory.

[MSQ]

#Q. Which of the following is/are FALSE with regard to sharing segments in a segmented system.

A. Maintaining a common segment table containing the information about shared segments.

B. Dividing the program into procedures and data and allowing only the procedures to be shared.

C. Dividing the shared segment into a set of pages and allowing only certain pages to be shared.

D. Sharing of Segments is never possible.

[MSQ]

#Q. Advantage of Dynamic Linking is/are that:

A. The segments that are not used in a run need not be linked into the process address space.

B. It reduces execution time overhead.

C. Only one copy of the Library can be maintained in Memory.

D. The linker need not construct the known segment table.

[NAT]

#Q. Assume that you have a page-reference string for a process with single frame (initially empty). The page-reference string has length p, and n distinct page numbers occur in it. Answer these questions for any page-replacement algorithms:

(a) What is a lower bound on the number of page faults?

(b) What is an upper bound on the number of page faults?

[NAT]

#Q. Consider the following page reference string:

1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6.

How many page faults would occur for the following replacement algorithms, assuming one, two, three, four, five, six, and seven frames? Remember that all frames are initially empty, so your first unique pages will cost one fault each.

- LRU replacement
- FIFO replacement
- Optimal replacement

[MSQ]

#Q. Which is not relevant with regard to Disk scheduling :

A. Allocating disk space to users in a fair manner.

B. Validating the file control information stored in the file.

C. Examining pending disk requests to determine the most efficient way to service the requests.

D. Reorganizing disk requests to minimize seek time.

[MSQ]

#Q. Which of the following is normally contained in the directory entry of a file?

<A + B + D>

A. Creation date.

B. Access control list.

C. A count of the number of free blocks in the disk.

D. Filename and extension.

[MCQ]

#Q. Which of the following is an example of a spooled device?

A. A line printer used to print the output of a number of jobs.

B. The terminal used to enter the input data for a Fortran program being executed.

C. The secondary memory device in a virtual memory system.

D. The swapping area on a disk used by the swapper.

#Q. A UNIX file system has 1-KB blocks and 4-byte disk addresses. What is the maximum file size if i-nodes contain 10 direct entries, and one single, double, and triple indirect entry each?

DBS = 1KB; DBA = 4B ; $N = \frac{1KB}{4B} = \underline{256}$.

File Size = $10 \times 1KB + 256 * 1KB + 256 * 256 * 1KB = 2^{26} = 64MB$

$+ (256 * 256 * 256 * 1KB)$

$2^8 \times 2^8 \times 2^8 \times 2^{10} = 2^{34} = \underline{16GB}$

$\Rightarrow$ 16.064 GB

#Q. In UNIX System V, the length of a block is 1 Kbyte, and each block can hold a total of 256 block addresses. Using the i-node scheme, what is the maximum size of a file?

D. + S.I + D.I

[NAT]

#Q. Consider a file system that uses I-nodes to represent files. Disk blocks are 8 KB in size, and a pointer to a disk block requires 4 bytes. This file system has 12 direct disk blocks, as well as single, double, and triple indirect disk blocks. What is the maximum size of a file that can be stored in this file system?

#Q. Consider a file system that uses i-nodes to represent files. Disk blocks are 8 KB in size, and a pointer to a disk block requires 4 bytes. This file system has 12 direct disk blocks, as well as single, double, and triple indirect disk blocks. What is the maximum size of a file that can be stored in this file system?

#Q. Ignoring overhead for directories and file descriptors, consider a file system in which files are stored in blocks of 16K. bytes. For each of the following file sizes, calculate the percentage of wasted file space due to incomplete filling of the last block: 41,600 bytes; 640,000 bytes; 4,064,000 bytes.

D.B.S = 16KB

1) File Size = 41,600B $\Rightarrow N_{Blocks}$ =

[NAT]

Disk Scheduling

#Q. Suppose that a disk drive has 5,000 cylinders, numbered 0 to 4,999. The drive is currently serving a request at cylinder 2,150, and the previous request was at cylinder 1,805. The queue of pending requests, in FIFO order, is:

2,069, 1,212, 2,296, 2,800, 544, 1,618, 356, 1,523, 4,965, 3681

Starting from the current head position, what is the total distance (in cylinders) that the disk arm moves to satisfy all the pending request for each of the following disk-scheduling algorithms?

a. FCFS
b. SSTF
c. SCAN
d. LOOK
e. C-SCAN
f. C-LOOK

2 mins Summary

Topic One

Topic Two

Topic Three

Topic Four

Topic Five

Disk Scheduling

THANK - YOU