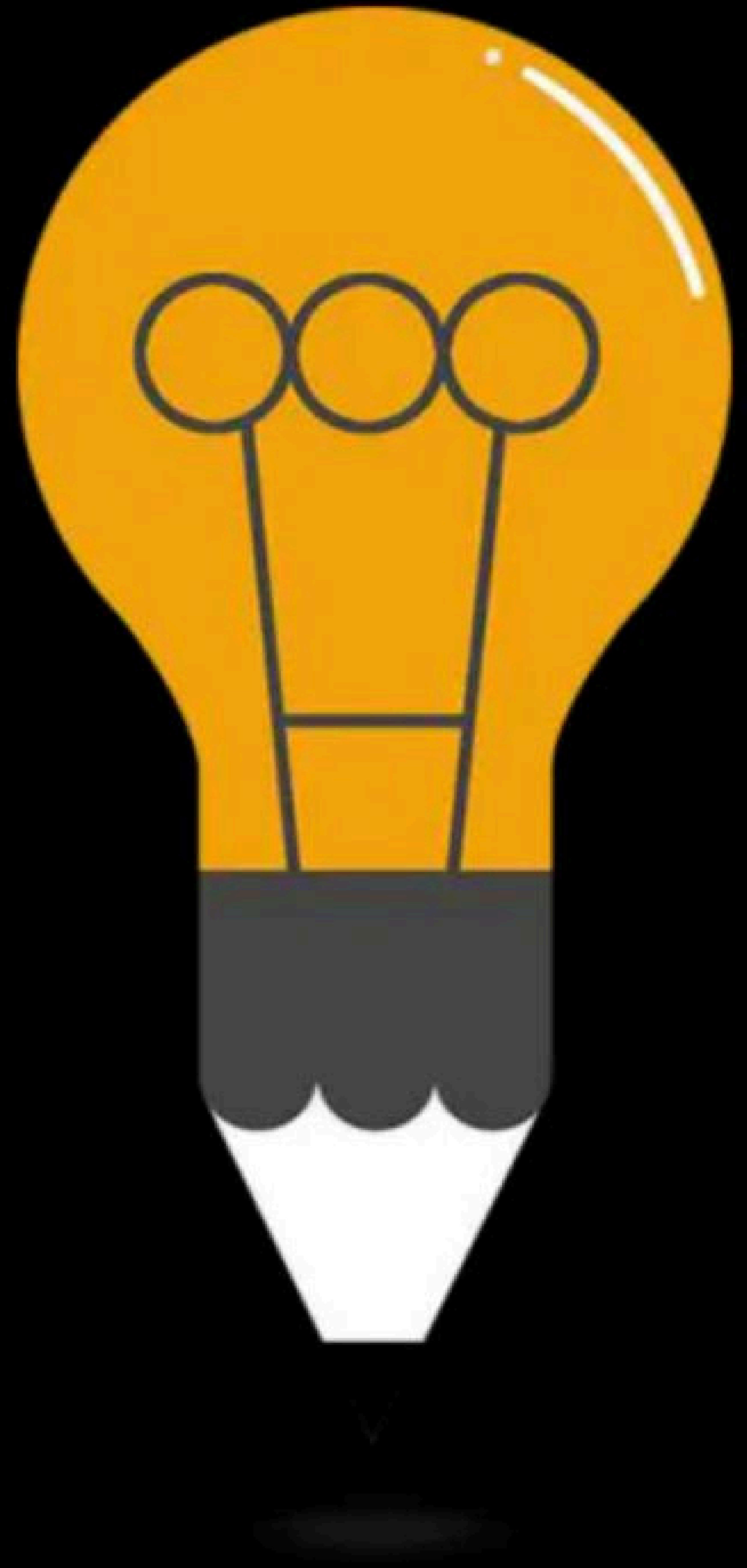
Process Synchronization: Introduction

Comprehensive Course on Operating System for GATE - 2024/25

Vishvadeep Gothi • Lesson 11 • May 2, 2023

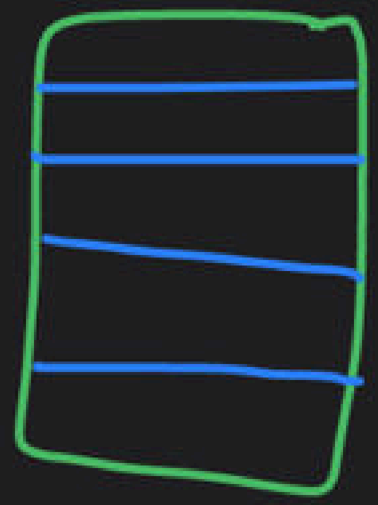


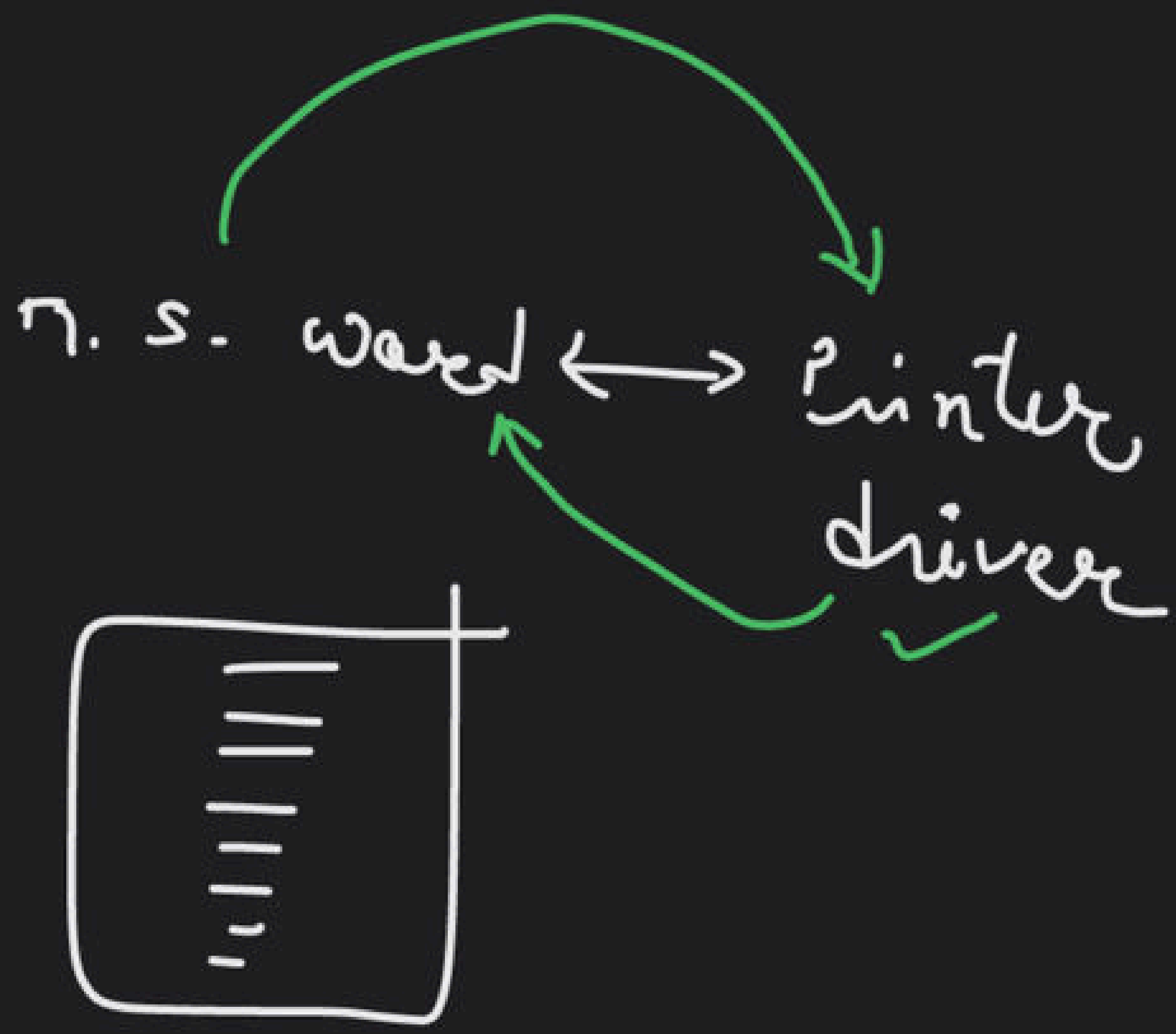
Operating System
Process Synchronization

By: Vishvadeep Gothi

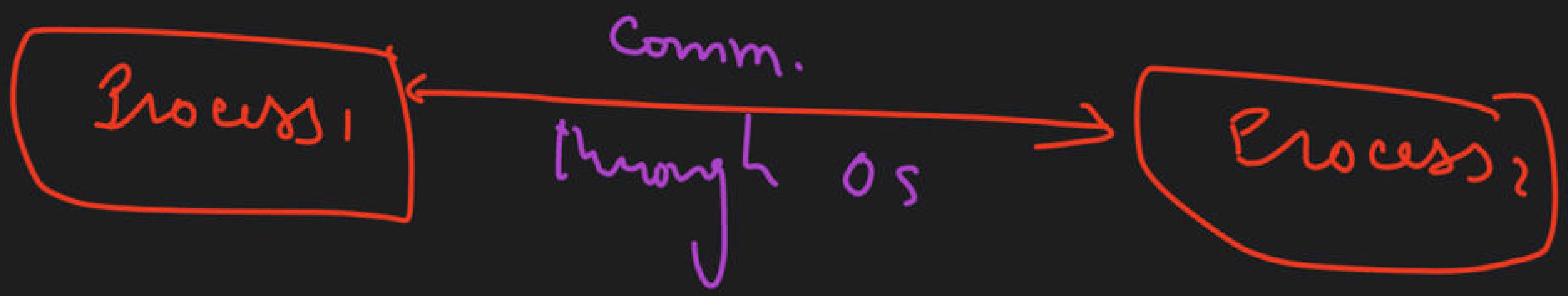
Process Mgmt

1. Process
2. Process implementatⁿ
3. Process states
4. scheduling





OS does Inter-process communicatn (IPC)



Process Types

1. Independent → Processes which do not require any communication with any other process.
2. Cooperating/Coordinating/Communicating

→ Processes which require communication with other processes.

→ can affect or can be affected by other processes.

Need Of Synchronization

B/w co-ordinating processes to get expected result.

↓

done by O.S.

Problems Without Synchronization

Problems without Synchronization:

- Inconsistency
- Loss of Data
- Deadlock

1 Parallel executⁿ of process :-

More than one CPU.

CPU 1	CPU 2	CPU 3
P1	P2	P3
runs	runs	runs

all P1, P2, P3 run simultaneously.

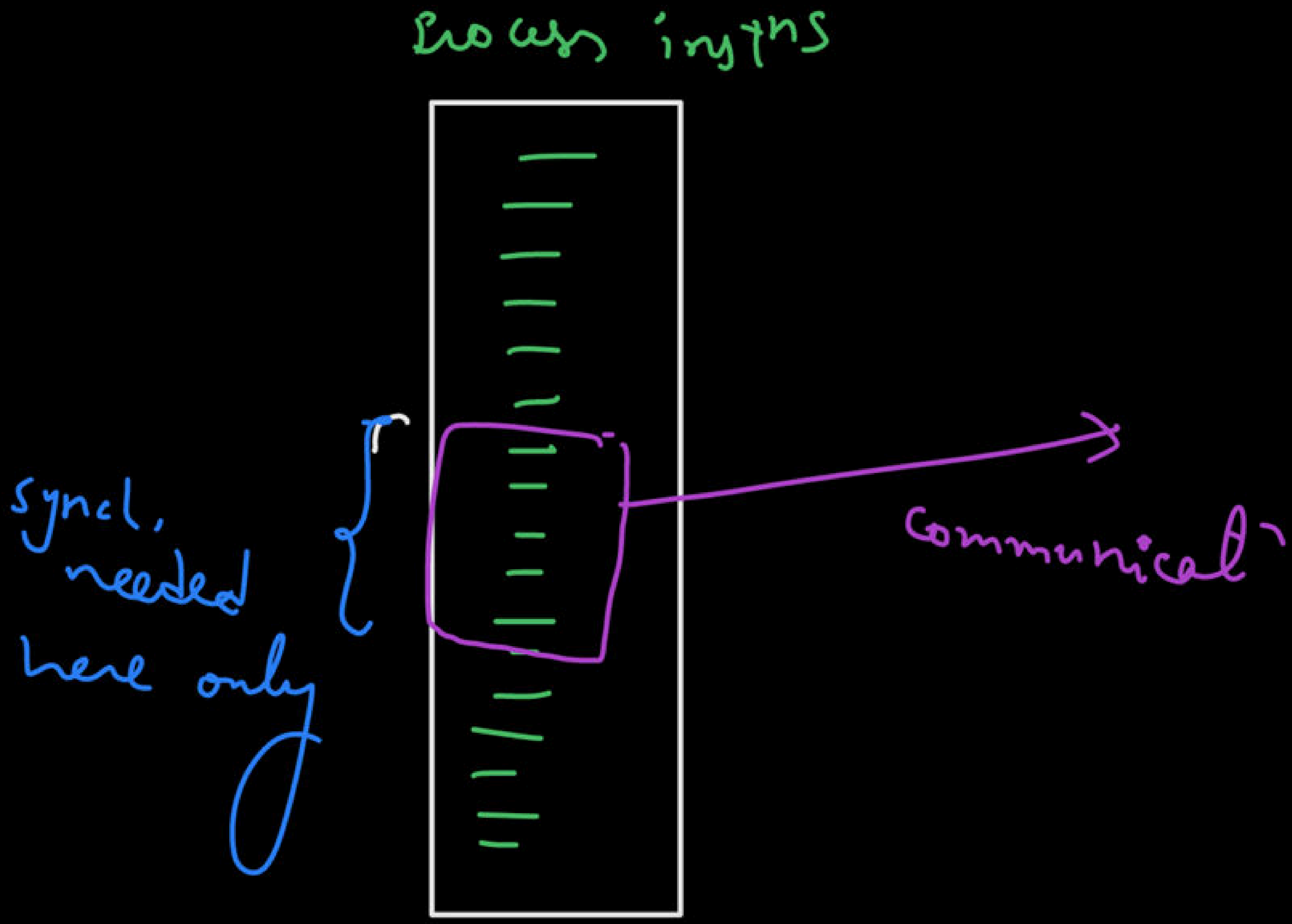
2 Concurrent executⁿ of processes:-

one CPU

P1	P2	P3	.....	P1	P2	...
OS	DBMS	AI		OS	D__	__
1 lec	1 lec	1 lec				

Entire Process Requires Synchronization?

NO



Critical Section

The critical section is a code segment where the shared variables can be accessed

or
array
or
resource

because
this needs
synchronization

Race Condition

A race condition is an undesirable situation, it occurs when the final result of concurrent processes depends on the sequence in which the processes complete their execution.

$\boxed{x = 5}$ memory

P1

$R1 = x$

$R1 = R1 + 3$

$x = R1$

P2

$R2 = x$

$R2 = R2 + 5$

$x = R2$

Run

$R1 = x$ $\quad R1 = 5$

P1 preempt

P2 runs

$R2 = x$ $\quad R2 = 5$

$R2 = R2 + 5 = 10$

P2 preempt

P1 runs

$R1 = R1 + 3 = 8$

$x = \cancel{5}\ \cancel{8}\ 10$

$x = \cancel{5}\ \cancel{10}\ 8$

$x = R1$

$x = R2$

P1 finishes, then

P2 - 1L

$x = 10$

P1 preempt

P2 runs

$x = R2$

P2 finish

P1 runs

$x = R1$

P2 finishes, then

P1 - 1L

$x = 8$

If P1 runs completely

then P2 runs - 1L

$x = 13$

Ans = 4

Question

X = 10

P1

X = X / 2

$R1 = x$

$R1 = R1/2$

$x = R1$

$R1 = \cancel{10}\ 5$

P2

X = X+4

$R2 = x$

$R2 = R+4$

$x = R2$

$R2 = \cancel{10}\ 14$

How many different values of X are possible after both processes finish executing?

1. P1 runs completly then P2 runs $\Rightarrow$ $x = \cancel{5}\ 9$
2. P2 runs completly then P1 runs $\Rightarrow$ $x = \cancel{14}\ 7$

③ P1 and P2 run first instr one-by-one; and read value of x as 10.

if P1 finishes first

$x = \cancel{5}\ 14$

if P2 finishes first

$x = \cancel{14}\ 5$

Possible values of $x \Rightarrow 5, 7, 9, 14$

Ans = 4

P1	P2	P1	P2
$R1 = x$	$R2 = x$	$R1 = R1/2 \quad x = R1$	$R2 = R2 + 4 \quad x = R2$
$R1 = 10$	$R2 = 10$	$R1 = 5$	$R2 = 14$

$\boxed{\cancel{10}\ \cancel{5}\ 14}$

x

Question

$x = 4$

The following pair of processes share a common variable X.

Process A
int Y;
8 Y = X*2;
X = Y;

Process B
int Z;
5 Z = X+1;
X = Z;

X is set to 4 before either process begins execution. As usual, statements within a process are executed sequentially, but statements in process A may execute in any order with respect to statements in process B.
How many different values of X are possible after both processes finish executing?

9, 10, 5, 8

Ans $\Rightarrow$ 4

$x = 16$

$x = 18$

P1

P2

P3

$x = x/2$

$x = x - 2$

$x = x + 2$

no. of distinct answers possible in x ?

$x = 8, 9, 7, 14, 18, 6, 10, 16$

Ans = 8

(a) > 6 ⇒ 30.43 %

(b) 6 ⇒ 19.5 %

(c) 7 ⇒ 15.22 %

(d) 8 ⇒ 19.57 %

(e) > 8 ⇒ 15 %

float

$x = 8$

P1

P2

P3

$x = x + 3$

$x = x / 2$

$x = x * 4$

no. of distinct values of x = ?

22, 28, 19, 17.5, 11, 4, 32, 5.5, 44, 7, 16,

35

$x = 16$ ~~8~~ ~~24~~ $\underline{\underline{96}}$

P1

P2

P3

$x = x/2$

$x = x * 4$

$x = x + 8$

max possible value of $x \Rightarrow$? $\underline{\underline{96}}$

min. — " —————— $\Rightarrow \underline{\underline{8}}$

Solution of Critical Section Problem

Requirements of Critical Section problem solution:

1. Mutual Exclusion
2. Progress
3. Bounded Waiting

2-Process Solution

Solution 1

```
Boolean lock=false;

while(true)
{
   while(lock);
   lock=true;
     CS
   lock=false;
     RS;
}
```

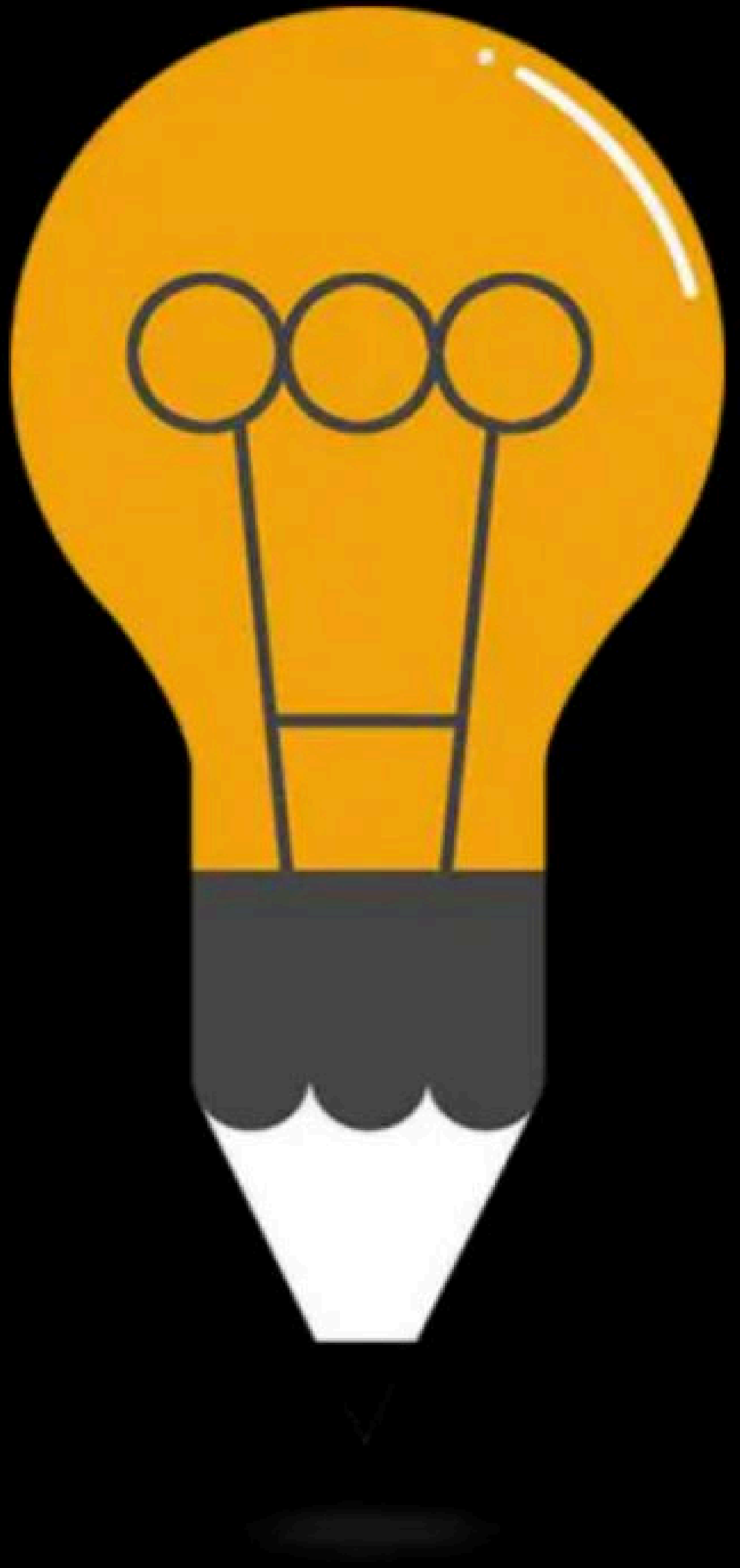
```
while(true)
{
   while(lock);
   lock=true;
     CS
   lock=false;
     RS;
}
```

Solution 2

```
int turn=0;

while(true)
{
  while(turn!=0);
   CS
  turn=1;
   RS;
}

while(true)
{
  while(turn!=1);
   CS
  turn=0;
   RS;
}
```

Solution 3: Peterson's Solution

```
Boolean Flag[2];
int turn;

while(true) {
  Flag[0]=true;
  turn=1;
  while(Flag[1] && turn==1)
     CS
  Flag[0]=False;
     RS;
}

while(true){
  Flag[1]=true;
  turn=0;
  while(Flag[0] && turn==0)
     CS
  Flag[1]=False;
     RS;
}
```



DPP

By: Vishvadeep Gothi

Question 1 GATE-2015

The following two functions $P1$ and $P2$ that share a variable B with an initial value of 2 execute concurrently.

```
P1() {
    C = B - 1;
    B = 2 * C;
}
```

```
P2(){
    D = 2 * B;
    B = D - 1;
}
```

The number of distinct values that B can possibly take after the execution is________________________.

Question 2 GATE-2019

Consider three concurrent processes P1, P2 and P3 as shown below, which access a shared variable D that has been initialized to 100.

P1	P2	P3
:	:	:
:	:	:
D = D + 20	D = D – 50	D = D + 10
:	:	:
:	:	:

The processes are executed on a uniprocessor system running a time-shared operating system. If the minimum and maximum possible values of D after the three processes have completed execution are X and Y respectively, then the value of Y – X is ______________.

unacademy

CPU Scheduling PYQ

By: Vishvadeep Gothi

GATE-1990

Fill in the blanks:

The highest-response ratio next scheduling policy favours ___________ jobs, but it also limits the waiting time of _________ jobs.

GATE-1993

Assume that the following jobs are to be executed on a single processor system

Job Id	CPU Burst Time
p	4
q	1
r	8
s	1
t	2

The jobs are assumed to have arrived at time 0^+ and in the order p, q, r, s, t. Calculate the departure time (completion time) for job p if scheduling is round robin with time slice 1

A. 4
B. 10
C. 11
D. 12
E. None of the above

GATE-1995

Which scheduling policy is most suitable for a time shared operating system?

A. Shortest Job First
B. Round Robin
C. First Come First Serve
D. Elevator

GATE-1995

The sequence …….. is an optimal non-preemptive scheduling sequence for the following jobs which leaves the CPU idle for ………. unit(s) of time.

Job	Arrival Time	Burst Time
1	0.0	9
2	0.6	5
3	1.0	1

A. $\{3,2,1\},1$
B. $\{2,1,3\},0$
C. $\{3,2,1\},0$
D. $\{1,2,3\},5$

GATE-1996

Four jobs to be executed on a single processor system arrive at time 0 in the order A, B, C, D. Their burst CPU time requirements are $4, 1, 8, 1$ time units respectively. The completion time of A under round robin scheduling with time slice of one time unit is

A. 10
B. 4
C. 8
D. 9

GATE-1998

Consider n processes sharing the CPU in a round-robin fashion. Assuming that each process switch takes s seconds, what must be the quantum size q such that the overhead resulting from process switching is minimized but at the same time each process is guaranteed to get its turn at the CPU at least every t seconds?

A. $q \leq \frac{t-ns}{n-1}$

B. $q \geq \frac{t-ns}{n-1}$

C. $q \leq \frac{t-ns}{n+1}$

D. $q \geq \frac{t-ns}{n+1}$

GATE-1998

In a computer system where the 'best-fit' algorithm is used for allocating 'jobs' to 'memory partitions', the following situation was encountered:

Partitions size in KB	$4K$ $8K$ $20K$ $2K$
Job sizes in KB	$2K$ $14K$ $3K$ $6K$ $6K$ $10K$ $20K$ $2K$
Time for execution	4 10 2 1 4 1 8 6

When will the $20K$ job complete?

GATE-2002

Which of the following scheduling algorithms is non-preemptive?

A. Round Robin
B. First-In First-Out
C. Multilevel Queue Scheduling
D. Multilevel Queue Scheduling with Feedback

GATE-2003

A uni-processor computer system only has two processes, both of which alternate $10\ \mathrm{ms}$ CPU bursts with $90\ \mathrm{ms}$ I/O bursts. Both the processes were created at nearly the same time. The I/O of both processes can proceed in parallel. Which of the following scheduling strategies will result in the *least* CPU utilization (over a long period of time) for this system?

A. First come first served scheduling

B. Shortest remaining time first scheduling

C. Static priority scheduling with different priorities for the two processes

D. Round robin scheduling with a time quantum of $5\ \mathrm{ms}$

GATE-2004

Consider the following set of processes, with the arrival times and the CPU-burst times gives in milliseconds.

Process	Arrival Time	Burst Time
$P1$	0	5
$P2$	1	3
$P3$	2	3
$P4$	4	1

What is the average turnaround time for these processes with the preemptive shortest remaining processing time first (SRPT) algorithm?

A. 5.50
B. 5.75
C. 6.00
D. 6.25

GATE-2005

We wish to schedule three processes $P1$, $P2$ and $P3$ on a uniprocessor system. The priorities, CPU time requirements and arrival times of the processes are as shown below.

Process	Priority	CPU time required	Arrival time (hh:mm:ss)
$P1$	10(highest)	20 sec	$00:00:05$
$P2$	9	10 sec	$00:00:03$
$P3$	8 (lowest)	15 sec	$00:00:00$

We have a choice of preemptive or non-preemptive scheduling. In preemptive scheduling, a late-arriving higher priority process can preempt a currently running process with lower priority. In non-preemptive scheduling, a late-arriving higher priority process must wait for the currently executing process to complete before it can be scheduled on the processor.
What are the turnaround times (time from arrival till completion) of $P2$ using preemptive and non-preemptive scheduling respectively?

A. 30 sec, 30 sec
B. 30 sec, 10 sec
C. 42 sec, 42 sec
D. 30 sec, 42 sec

unacademy

GATE-2006

Consider three CPU-intensive processes, which require 10, 20 and 30 time units and arrive at times 0, 2 and 6, respectively. How many context switches are needed if the operating system implements a shortest remaining time first scheduling algorithm? Do not count the context switches at time zero and at the end.

A. 1
B. 2
C. 3
D. 4

GATE-2006

Consider three processes (process id $0, 1, 2$ respectively) with compute time bursts $2, 4$ and 8 time units. All processes arrive at time zero. Consider the longest remaining time first (LRTF) scheduling algorithm. In LRTF ties are broken by giving priority to the process with the lowest process id. The average turn around time is:

A. 13 units
B. 14 units
C. 15 units
D. 16 units

GATE-2006

Consider three processes, all arriving at time zero, with total execution time of 10, 20 and 30 units, respectively. Each process spends the first 20% of execution time doing I/O, the next 70% of time doing computation, and the last 10% of time doing I/O again. The operating system uses a shortest remaining compute time first scheduling algorithm and schedules a new process either when the running process gets blocked on I/O or when the running process finishes its compute burst. Assume that all I/O operations can be overlapped as much as possible. For what percentage of time does the CPU remain idle?

A. 0%
B. 10.6%
C. 30.0%
D. 89.4%

GATE-2006

The arrival time, priority, and duration of the CPU and I/O bursts for each of three processes P_1, P_2 and P_3 are given in the table below. Each process has a CPU burst followed by an I/O burst followed by another CPU burst. Assume that each process has its own I/O resource.

Process	Arrival time	Priority	Burst duration(CPU)	Burst duration(I/O)	Burst duration(CPU)
P_1	0	2	1	5	3
P_2	2	3 (lowest)	3	3	1
P_3	3	1 (highest)	2	3	1

The multi-programmed operating system uses preemptive priority scheduling. What are the finish times of the processes P_1, P_2 and P_3?

A. 11, 15, 9
B. 10, 15, 9
C. 11, 16, 10
D. 12, 17, 11

GATE-2007

Group 1 contains some CPU scheduling algorithms and Group 2 contains some applications. Match entries in Group 1 to entries in Group 2.

Group I	Group II
(P) Gang Scheduling	(1) Guaranteed Scheduling
(Q) Rate Monotonic Scheduling	(2) Real-time Scheduling
(R) Fair Share Scheduling	(3) Thread Scheduling

A. P-3; Q-2; R-1
B. P-1; Q-2; R-3
C. P-2; Q-3; R-1
D. P-1; Q-3; R-2

GATE-2007

An operating system used Shortest Remaining System Time first (SRT) process scheduling algorithm. Consider the arrival times and execution times for the following processes:

Process	Execution Time	Arrival time
$P1$	20	0
$P2$	25	15
$P3$	10	30
$P4$	15	45

What is the total waiting time for process $P2$?

A. 5
B. 15
C. 40
D. 55

GATE-2007

Consider n jobs $J_1, J_2 \dots J_n$ such that job J_i has execution time t_i and a non-negative integer weight w_i. The weighted mean completion time of the jobs is defined to be $\frac{\sum_{i=1}^{n} w_i T_i}{\sum_{i=1}^{n} w_i}$, where T_i is the completion time of job J_i. Assuming that there is only one processor available, in what order must the jobs be executed in order to minimize the weighted mean completion time of the jobs?

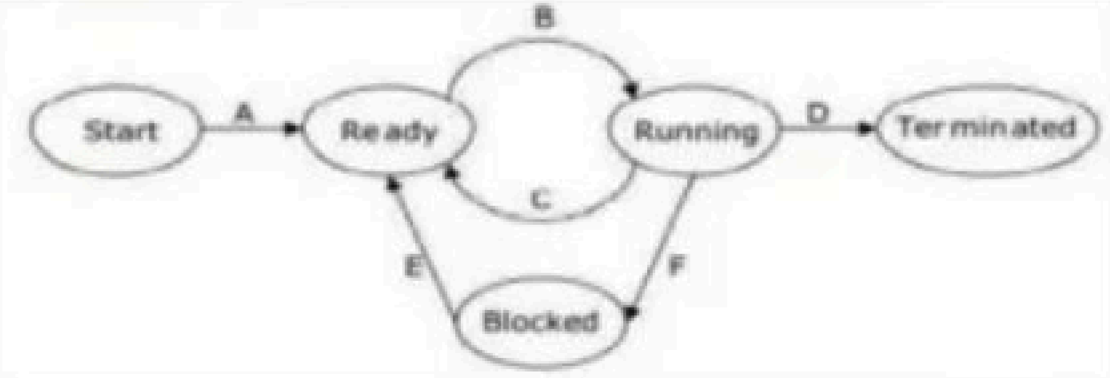
A. Non-decreasing order of t_i
B. Non-increasing order of w_i
C. Non-increasing order of $w_i t_i$
D. Non-increasing order of w_i / t_i

GATE-2008

If the time-slice used in the round-robin scheduling policy is more than the maximum time required to execute any process, then the policy will

A. degenerate to shortest job first
B. degenerate to priority scheduling
C. degenerate to first come first serve
D. none of the above

GATE-2009



Now consider the following statements:

I. If a process makes a transition D, it would result in another process making transition A immediately.
II. A process P_2 in blocked state can make transition E while another process P_1 is in running state.
III. The OS uses preemptive scheduling.
IV. The OS uses non-preemptive scheduling.

Which of the above statements are TRUE?

A. I and II
B. I and III
C. II and III
D. II and IV

GATE-2010

Which of the following statements are true?

I. Shortest remaining time first scheduling may cause starvation
II. Preemptive scheduling may cause starvation
III. Round robin is better than FCFS in terms of response time

A. I only
B. I and III only
C. II and III only
D. I, II and III

GATE-2011

Consider the following table of arrival time and burst time for three processes P0, P1 and P2.

Process	Arrival Time	Burst Time
P0	0 ms	9 ms
P1	1 ms	4 ms
P2	2 ms	9 ms

The pre-emptive shortest job first scheduling algorithm is used. Scheduling is carried out only at arrival or completion of processes. What is the average waiting time for the three processes?

A. 5.0 ms
B. 4.33 ms
C. 6.33 ms
D. 7.33 ms

GATE-2012

Consider the 3 processes, P1, P2 and P3 shown in the table.

Process	Arrival time	Time Units Required
P1	0	5
P2	1	7
P3	3	4

The completion order of the 3 processes under the policies FCFS and RR2 (round robin scheduling with CPU quantum of 2 time units) are

A. **FCFS**: P1, P2, P3 **RR2**: P1, P2, P3
B. **FCFS**: P1, P3, P2 **RR2**: P1, P3, P2
C. **FCFS**: P1, P2, P3 **RR2**: P1, P3, P2
D. **FCFS**: P1, P3, P2 **RR2**: P1, P2, P3

GATE-2013

A scheduling algorithm assigns priority proportional to the waiting time of a process. Every process starts with zero (the lowest priority). The scheduler re-evaluates the process priorities every T time units and decides the next process to schedule. Which one of the following is **TRUE** if the processes have no I/O operations and all arrive at time zero?

A. This algorithm is equivalent to the first-come-first-serve algorithm.
B. This algorithm is equivalent to the round-robin algorithm.
C. This algorithm is equivalent to the shortest-job-first algorithm.
D. This algorithm is equivalent to the shortest-remaining-time-first algorithm.

GATE-2014

Consider the following set of processes that need to be scheduled on a single CPU. All the times are given in milliseconds.

Process Name	**Arrival Time**	**Execution Time**
A	0	6
B	3	2
C	5	4
D	7	6
E	10	3

Using the *shortest remaining time first* scheduling algorithm, the average process turnaround time (in msec) is ______________________.

GATE-2014

Three processes A, B and C each execute a loop of 100 iterations. In each iteration of the loop, a process performs a single computation that requires t_c CPU milliseconds and then initiates a single I/O operation that lasts for t_{io} milliseconds. It is assumed that the computer where the processes execute has sufficient number of I/O devices and the OS of the computer assigns different I/O devices to each process. Also, the scheduling overhead of the OS is negligible. The processes have the following characteristics:

Process id	t_c	t_{io}
A	100 ms	500 ms
B	350 ms	500 ms
C	200 ms	500 ms

The processes A, B, and C are started at times 0, 5 and 10 milliseconds respectively, in a pure time sharing system (round robin scheduling) that uses a time slice of 50 milliseconds. The time in milliseconds at which process C would ***complete*** its first I/O operation is ____________.

GATE-2014

An operating system uses *shortest remaining time first* scheduling algorithm for pre-emptive scheduling of processes. Consider the following set of processes with their arrival times and CPU burst times (in milliseconds):

Process	Arrival Time	Burst Time
$P1$	0	12
$P2$	2	4
$P3$	3	6
$P4$	8	5

The average waiting time (in milliseconds) of the processes is ______.

GATE-2015

Consider a uniprocessor system executing three tasks T_1, T_2 and T_3 each of which is composed of an infinite sequence of jobs (or instances) which arrive periodically at intervals of 3, 7 and 20 milliseconds, respectively. The priority of each task is the inverse of its period, and the available tasks are scheduled in order of priority, which is the highest priority task scheduled first. Each instance of T_1, T_2 and T_3 requires an execution time of 1, 2 and 4 milliseconds, respectively. Given that all tasks initially arrive at the beginning of the 1^{st} millisecond and task preemptions are allowed, the first instance of T_3 completes its execution at the end of____________________milliseconds.

GATE-2015

The maximum number of processes that can be in $Ready$ state for a computer system with n CPUs is :

A. n
B. n^2
C. 2^n
D. Independent of n

GATE-2015

For the processes listed in the following table, which of the following scheduling schemes will give the lowest average turnaround time?

Process	Arrival Time	Process Time
A	0	3
B	1	6
C	4	4
D	6	2

A. First Come First Serve
B. Non-preemprive Shortest job first
C. Shortest Remaining Time
D. Round Robin with Quantum value two

GATE-2016

Consider an arbitrary set of CPU-bound processes with unequal CPU burst lengths submitted at the same time to a computer system. Which one of the following process scheduling algorithms would minimize the average waiting time in the ready queue?

A. Shortest remaining time first
B. Round-robin with the time quantum less than the shortest CPU burst
C. Uniform random
D. Highest priority first with priority proportional to CPU burst length

GATE-2016

Consider the following processes, with the arrival time and the length of the CPU burst given in milliseconds. The scheduling algorithm used is preemptive shortest remaining-time first.

Process	Arrival Time	Burst Time
P_1	0	10
P_2	3	6
P_3	7	1
P_4	8	3

The average turn around time of these processes is ___________ milliseconds.

GATE-2017

Consider the following CPU processes with arrival times (in milliseconds) and length of CPU bursts (in milliseconds) as given below:

Process	Arrival Time	Burst Time
P1	0	7
P2	3	3
P3	5	5
P4	6	2

If the pre-emptive shortest remaining time first scheduling algorithm is used to schedule the processes, then the average waiting time across all processes is ______________ milliseconds.

GATE-2017

Consider the set of process with arrival time (in milliseonds) , CPU burst time (in millisecods) and priority (0 is the highest priority) shown below . None of the process have I/O burst time

Process	Arival Time	Burst Time	Priority
P1	0	11	2
P2	5	28	0
P3	12	2	3
P4	2	10	1
P5	9	16	4

The average waiting time (in milli seconds) of all the process using premtive priority scheduling algorithm is _______

GATE-2019

Consider the following four processes with arrival times (in milliseconds) and their length of CPU bursts (in milliseconds) as shown below:

Process	P1	P2	P3	P4
Arrival time	0	1	3	4
CPU burst time	3	1	3	Z

These processes are run on a single processor using preemptive Shortest Remaining Time First scheduling algorithm. If the average waiting time of the processes is 1 millisecond, then the value of Z is________.

GATE-2020

Consider the following statements about process state transitions for a system using preemptive scheduling.

I. A running process can move to ready state.
II. A ready process can move to running state.
III. A blocked process can move to running state.
IV. A blocked process can move to ready state.

Which of the above statements are TRUE?

A. I, II, and III only
B. II and III only
C. I, II, and IV only
D. I, II, III and IV only

GATE-2021

Which of the following statement(s) is/are correct in the context of CPU scheduling?

A. Turnaround time includes waiting time

B. The goal is to only maximize CPU utilization and minimize throughput

C. Round-robin policy can be used even when the CPU time required by each of the processes is not known apriori

D. Implementing preemptive scheduling needs hardware support

Happy Learning.!