CS & IT ENGINEERING

OPERATING SYSTEMS

Process Synchronization

Lecture No. 1

By- Dr. Khaleel Khan Sir

TOPICS TO BE COVERED

- Round Robin
- Multi Level Queue Scheduling
- Process Synchronization

Performance of Round Robin

Time - Quantum

Very Small

Efficiency ~ 0 (5%)

Very large

↳ Works like FCFS

Small

→ More Context Switching (overhead)

→ Make the System better responsive

large

→ Less Context switching (Less overhead)

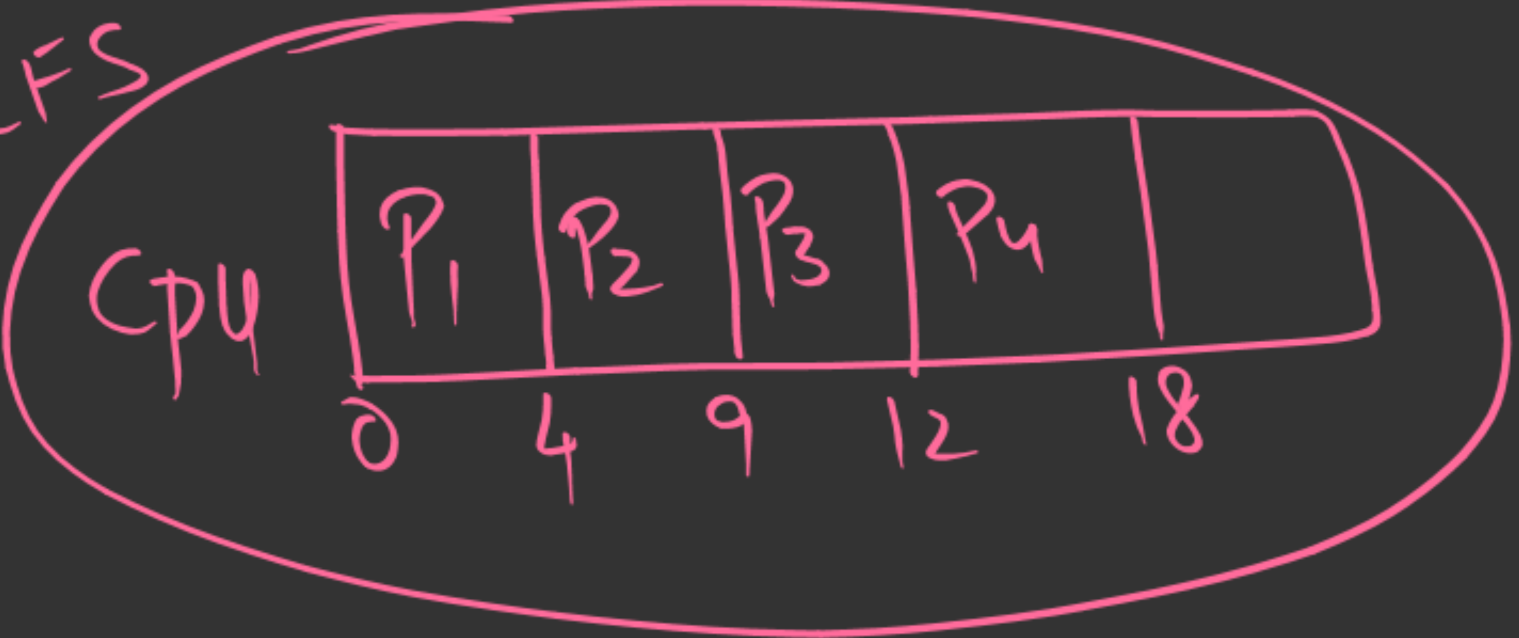
→ poor Interactiveness

responsiveness

TQ = 10

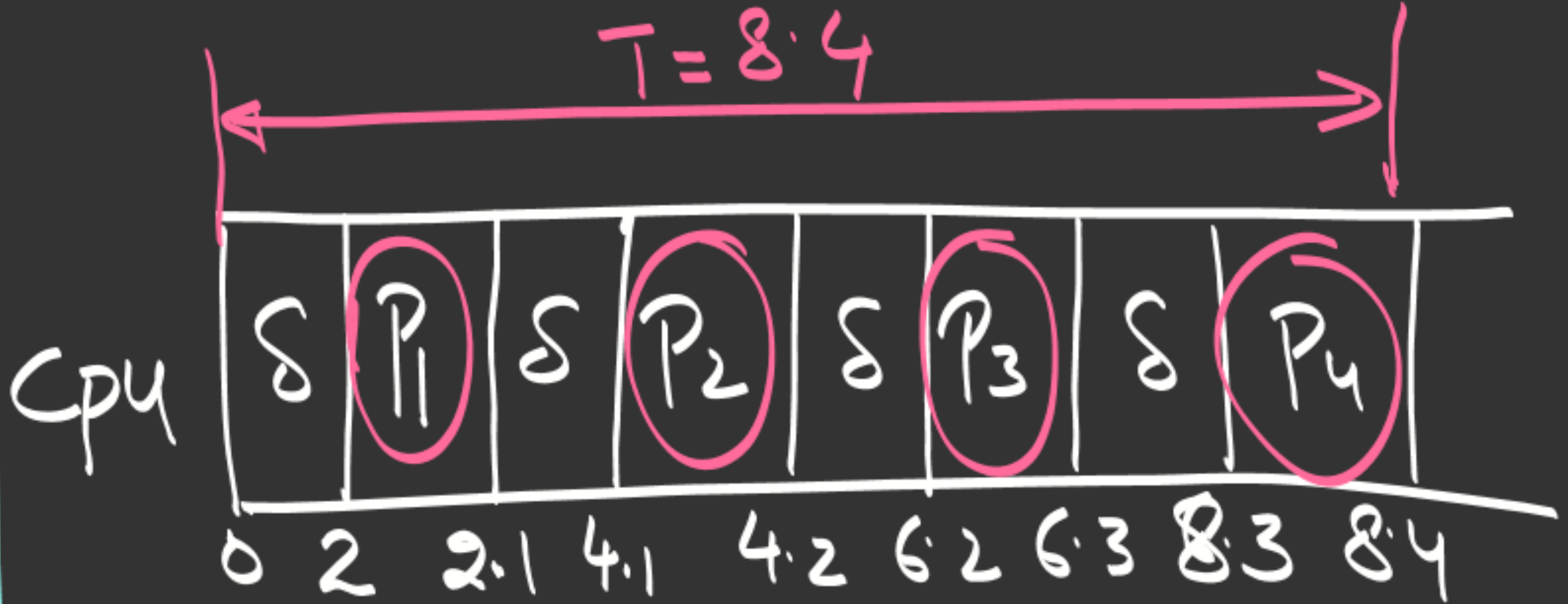
P.No	A.T	B.T
1	0	4
2	0	5
3	0	3
4	2	6

FCFS



TQ = 0.1 (very small)

$\delta = 2$

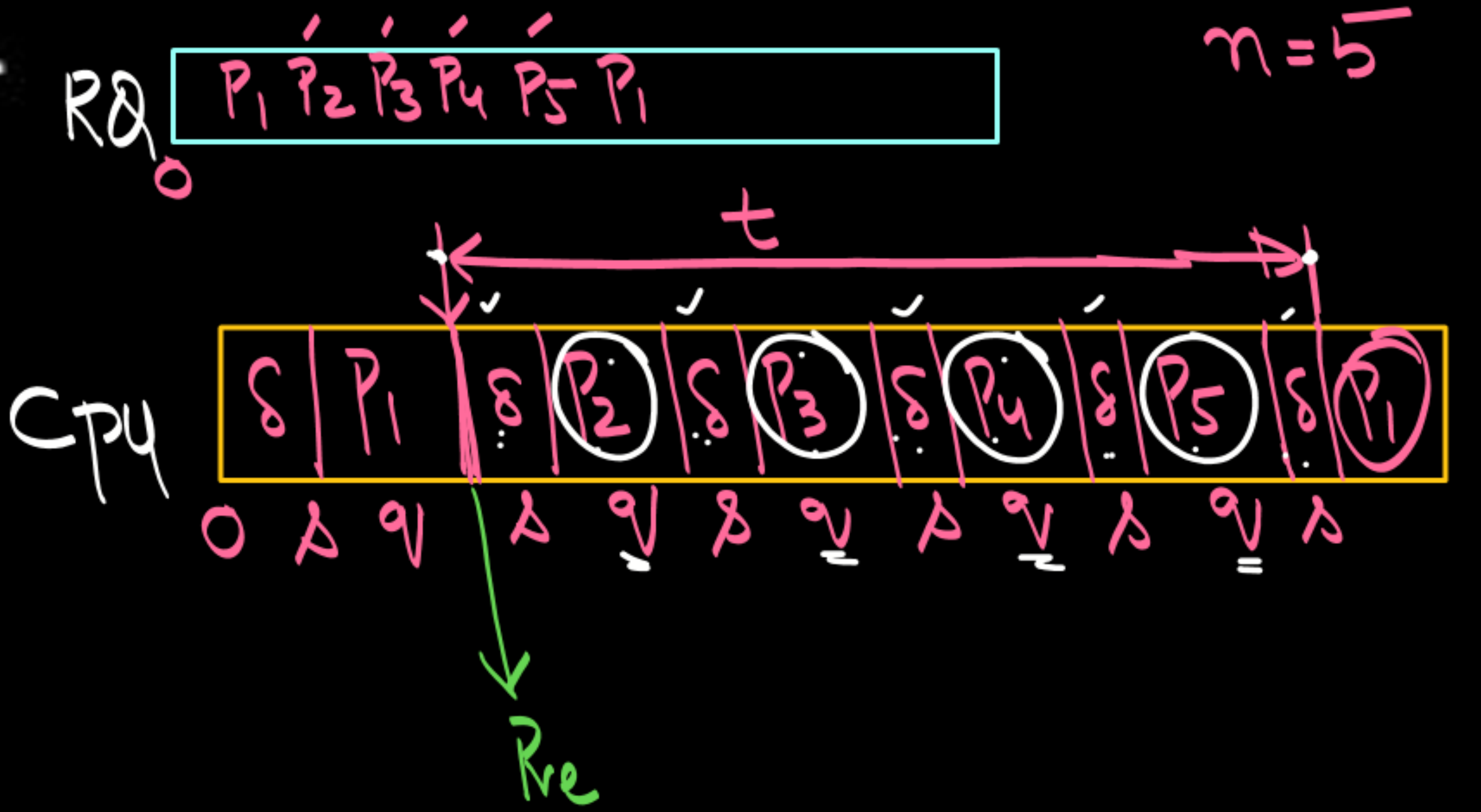


$$\text{CPU efficiency} = \frac{\text{useful work-time}}{\text{Total-time (useful + ovhd)}}$$

$$= \frac{0.4}{8.4} = \frac{4}{84} = \frac{1}{21} \sim \underline{5\%}$$

Q. Consider a System with 'n' Processes arriving at time 0^+ with substantially large Burst Times. The CPU scheduling overhead is 's' seconds, Time Quantum is 'q' seconds. Using Round Robin scheduling, what must be the value of Time Quantum 'q' such that each Process is guaranteed to get its turn at the CPU exactly after 't' seconds in its subsequent run-on CPU.

$n = 5$



$(n * s)$

$$\boxed{(t - ns) = (n-1) \cdot q}$$

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

1) $q = \left(\dfrac{t-ns}{n-1}\right)$: Process will get onto cpu exactly after 't' sec's ;

2) $q < \left(\dfrac{t-ns}{n-1}\right)$: Process will get onto cpu atleast once within 't'

< one/more time >

3) $q > \left(\dfrac{t-ns}{n-1}\right)$: Process will get onto cpu atleast every 't' sec

< Process will min wait for 't' sec to run on cpu >

Q.

Consider the following set of Processes, assumed to have arrived at time 0 . Consider the CPU scheduling algorithms Shortest Job First (SJF) and Round Robin (RR). For *RR* assume that the processes are scheduled in the order P_1, P_2, P_3, P_4.

Processes	P_1	P_2	P_3	P_4
Burst time (in ms)	8	7	2	4

S.J.F

P_3	P_4	P_2	P_1

0 2 6 13 21

$$\text{Av. TAT} = \frac{21+13+2+6}{4} = \frac{42}{4} = 10.5$$

If the time quantum for *RR* is 4 ms, then the absolute value of the difference between the average turnaround times (in *ms*) of *SJF* and *RR* (round off to 2 decimal places) is

$$\text{Av. TAT (RR)} = 15.75$$

Q. Consider four Processes P, Q, R, and S scheduled on a CPU as per Round Robin Algorithm with a Time Quantum of 4 units. The Processes arrive in the order P, Q, R, S, all at time $t = 0$. There is exactly one context switch from S to Q, exactly one context switch from R to Q, and exactly two context switches from Q to R. There is no context switch from S to P. Switching to a ready process after the termination of another process is also considered a context switch. Which one of the following is NOT the values of PQRS.

G1

(A) $P = 4, Q = 10, R = 6, S = 2$

(B) $P = 2, Q = 9, R = 5, S = 1$

(C) $P = 4, Q = 12, R = 5, S = 4$

(D) $P = 3, Q = 7, R = 7, S = 3$

(2-ways)

Brute Force

Applying appropriate strategy

Q. Consider Processes $P_1 \& P_2$ arriving in the ready queue at time 0 with following properties.

H/W

i) P1 needs a total of 12 units of CPU time and 20 units of I/O time. After every 3 units of CPU time $P1$ spends 5 units on I/O.

P1: <3; 5; 3; 5; 3; 5; 3; 5>

ii) $P2$ needs a total of 15 units of CPU time and no I/O. P2 arrives just after P1.

Compute the Completion times of $P_1 \& P_2$ using the following scheduling techniques:

1. SRTF
2. Round Robin with Time Quanta = 4 units

RQ $\boxed{P_1 \; P_2}$

Q. Which of the following statements (s) is/are correct in the context of CPU Scheduling?

MSQ : BCD

(A) The goal is to only maximize CPU utilization and minimize throughput

(B) Turnaround time includes waiting time

device (H/W)

Interrupt

(C) Implementing preemptive scheduling needs hardware support.

(D) Round-robin policy can be used even when the CPU time required by each of the processes is not known Apriority.

Multi-Level Queue Scheduling

200 ✓

1000

STS

RQ → Single Ready 'Q'

R.R

CPU

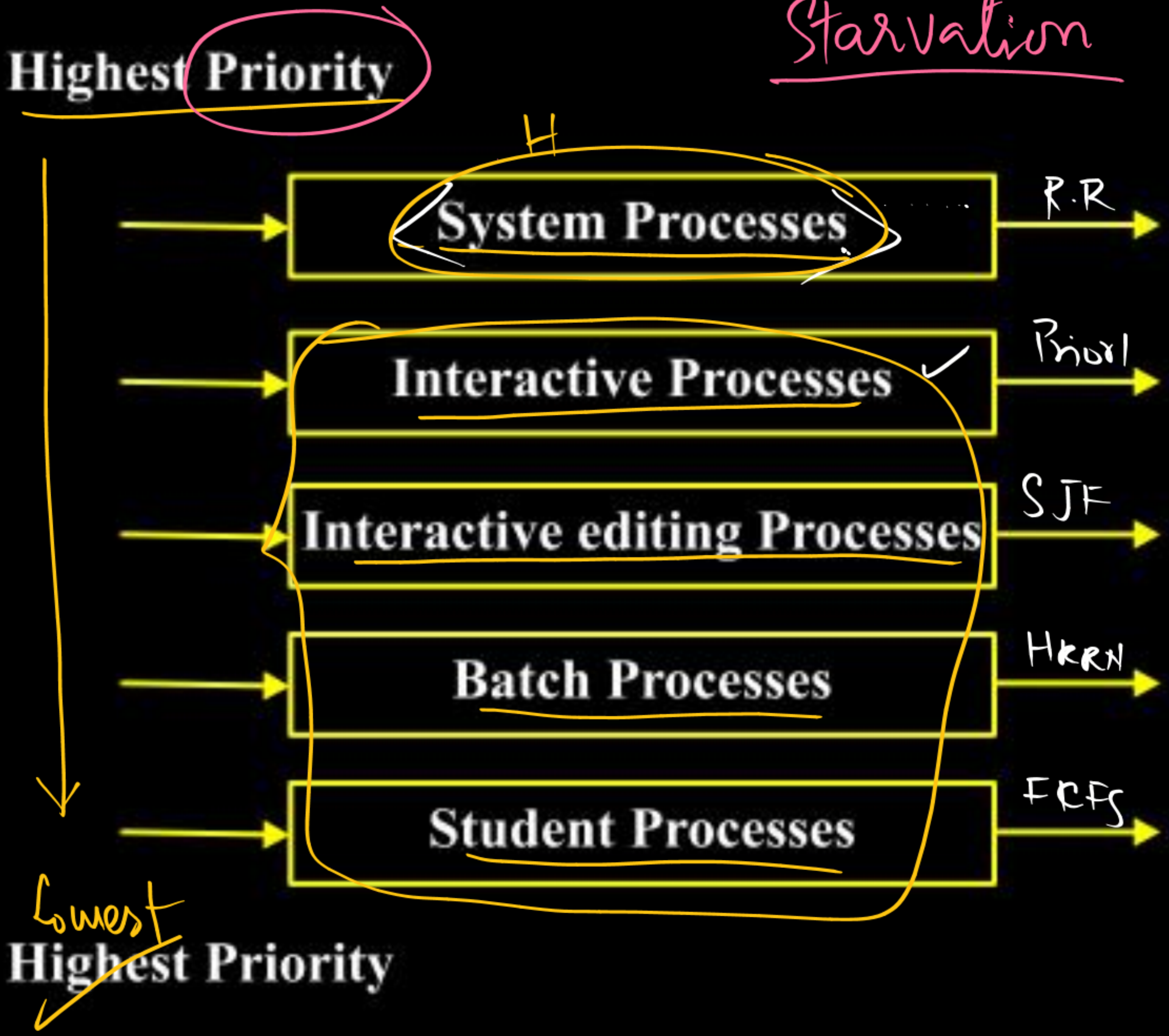
Types of Process
- OS
- User
  - Interactive (Foreground)
  - Background

(i) Searching Time

(ii) All processes will use Single Scheduling Technique

(Multiple - Queues)

Starvation

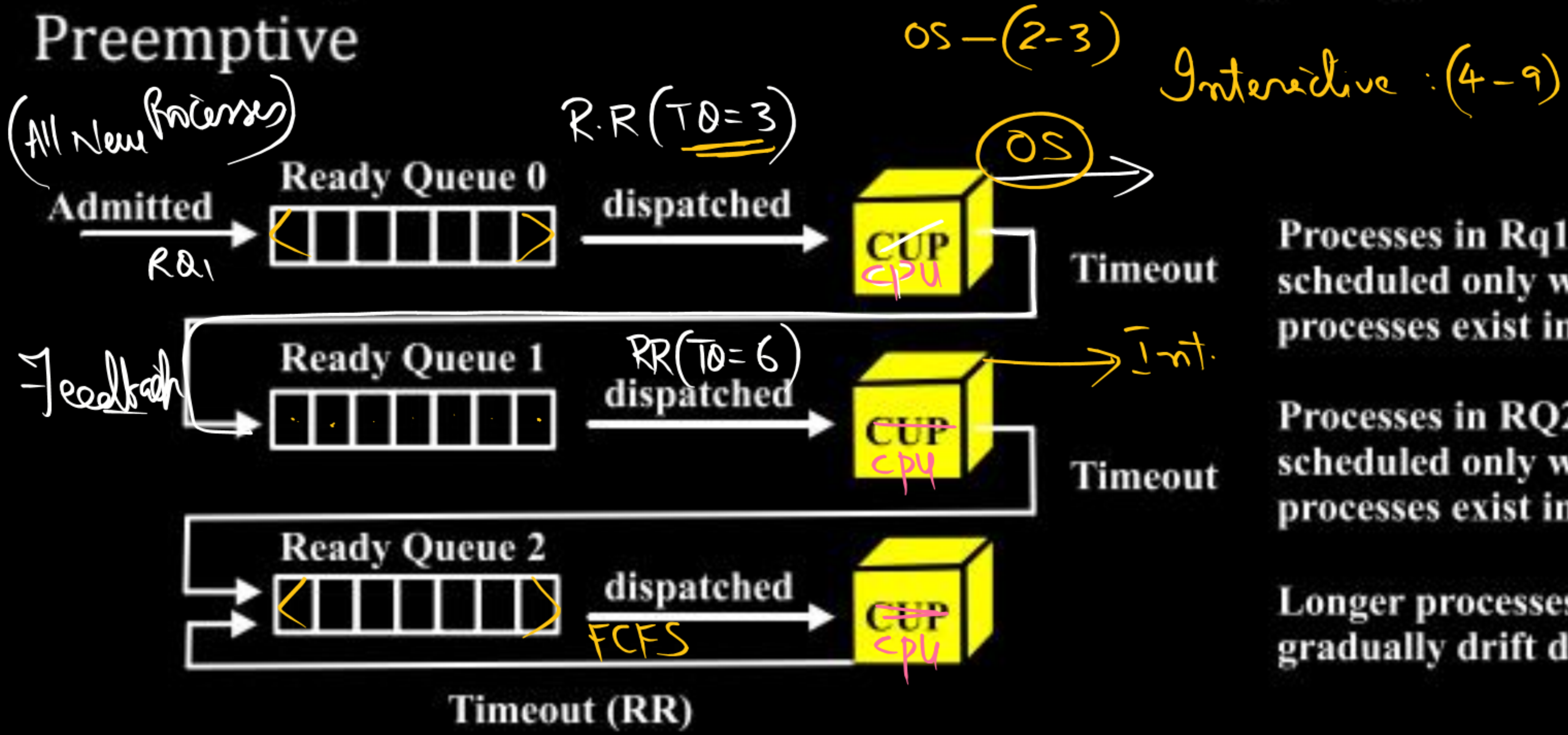


Multi-Level Feedback Q

CPU

Multilevel feedback Queue Scheduling

- Another way to put a preference on short-lived processes
  - Penalize processes that have been running longer
- Preemptive



Processes in Rq1 are scheduled only when no processes exist in Rq0

Processes in RQ2 are scheduled only when no processes exist in Rq1

Longer processes gradually drift downward

Q. Consider a system which has CPU bound process, which require the burst time of 80 seconds, the multilevel feedback queue scheduling algorithm is used and the First queue time quantum is '4' seconds and in each level it is incremented by '10' seconds. Then how many times the process will be interrupted, and on which queue the process will terminate the execution?

1993-2022

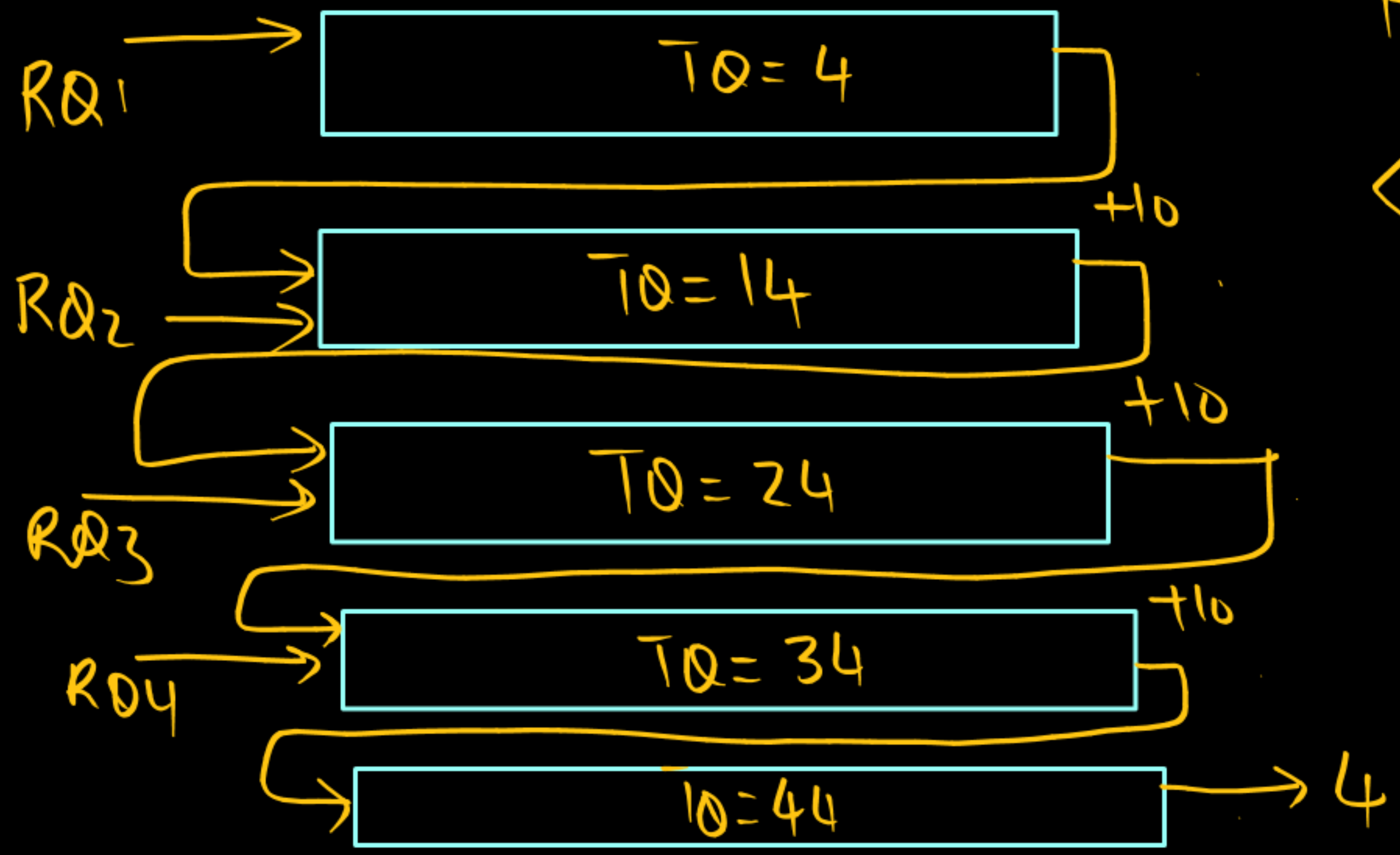
B.T = 80 ms

A. 2,5

B. 3,5

C. 4,5 ✓

D. None



4 + 14 + 24 + 34

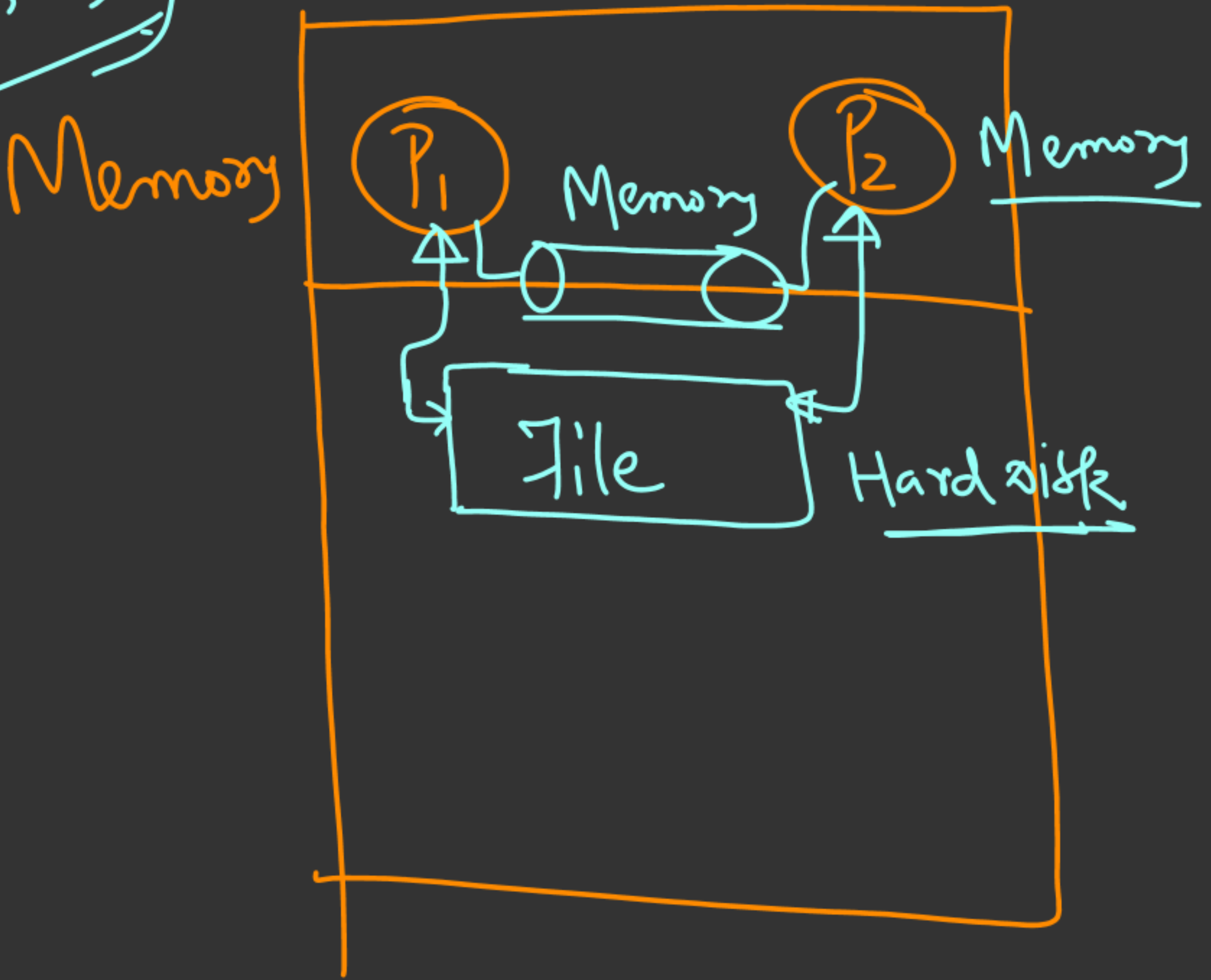
(76)

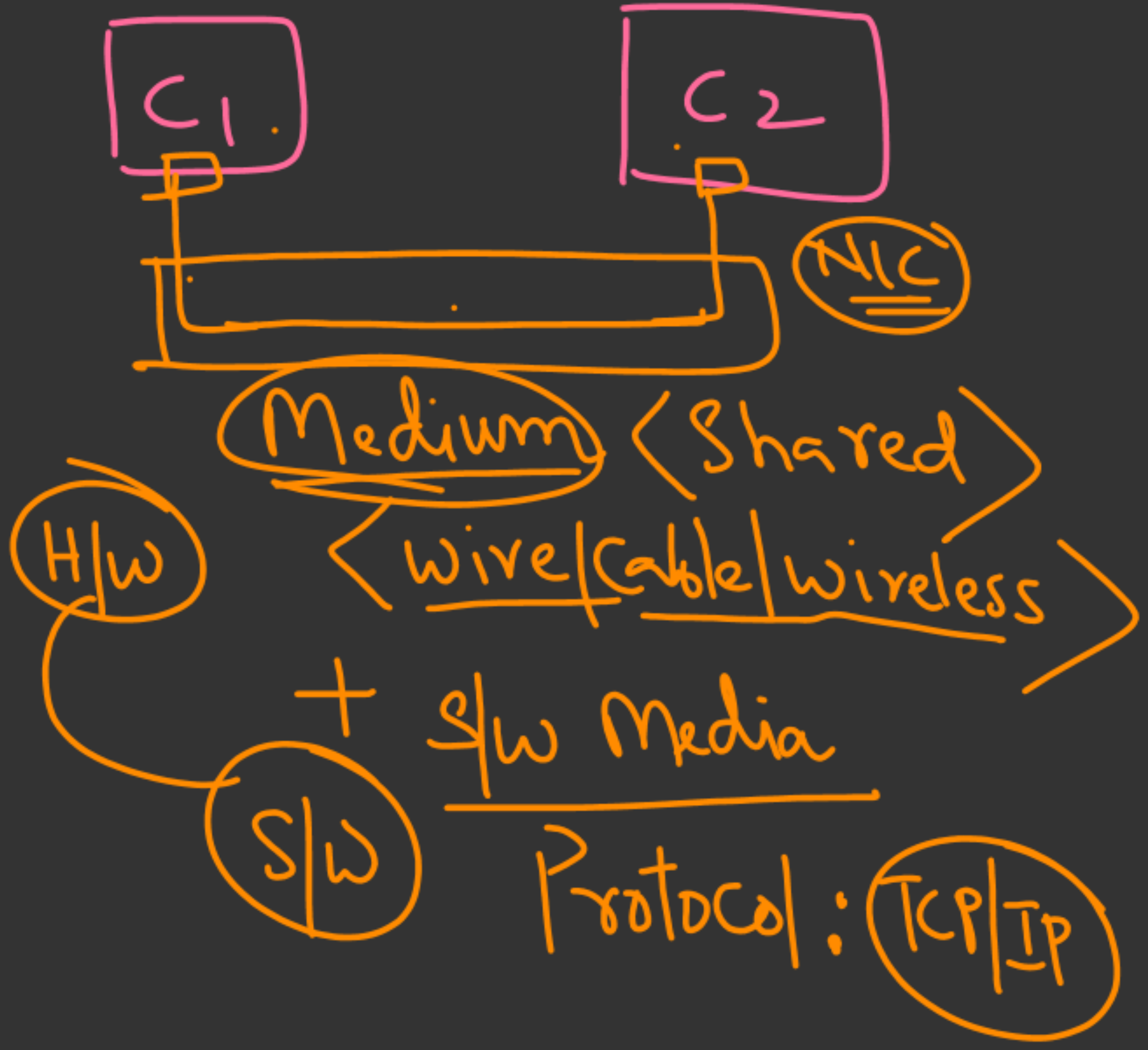
5

** I.P.C & Process Synchronization
Process coordination

<Inter Process Communication>

IPC Mechanisms

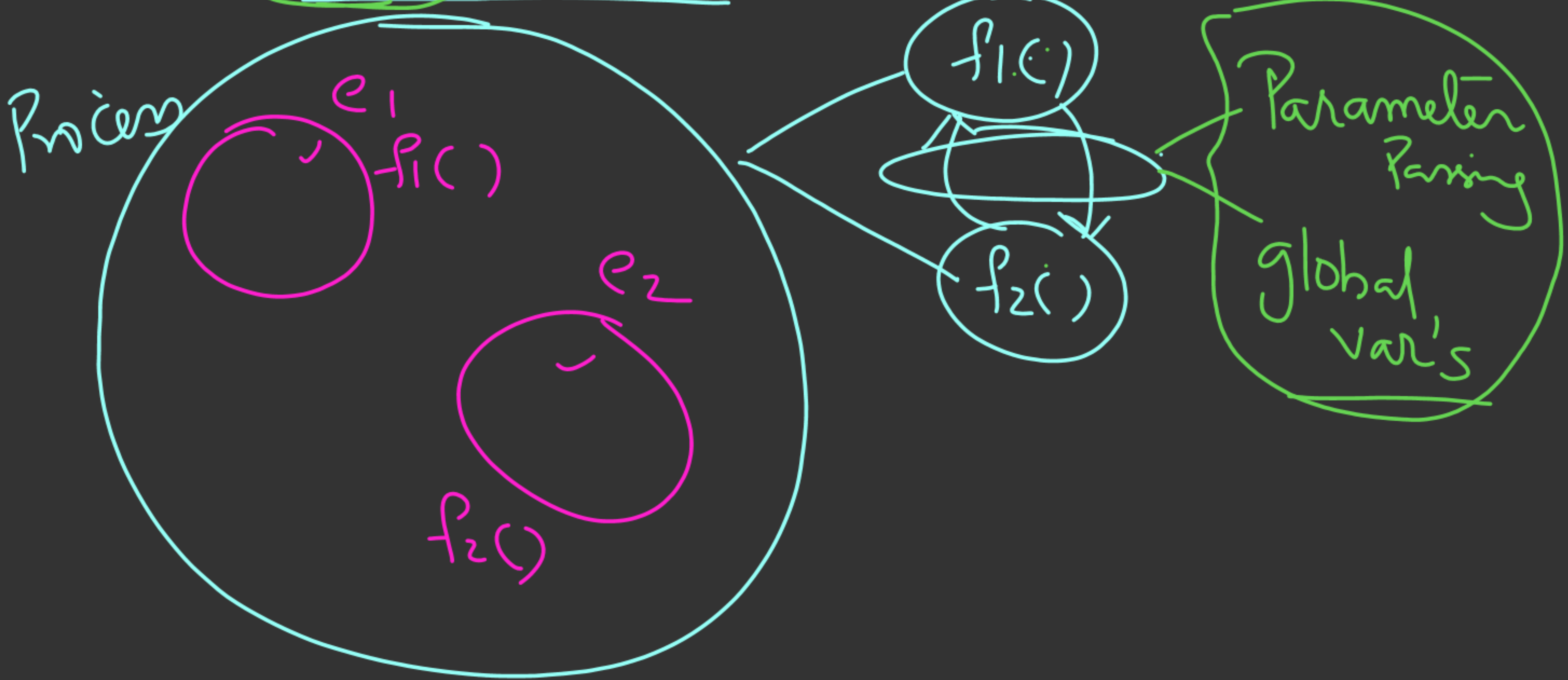




IPC

Intra - Process

int x;




THANK YOU
GW
SOLDIERS !