# CS & IT ENGINEERING

## Operating Systems

## Process Management

**Lecture No. 2**

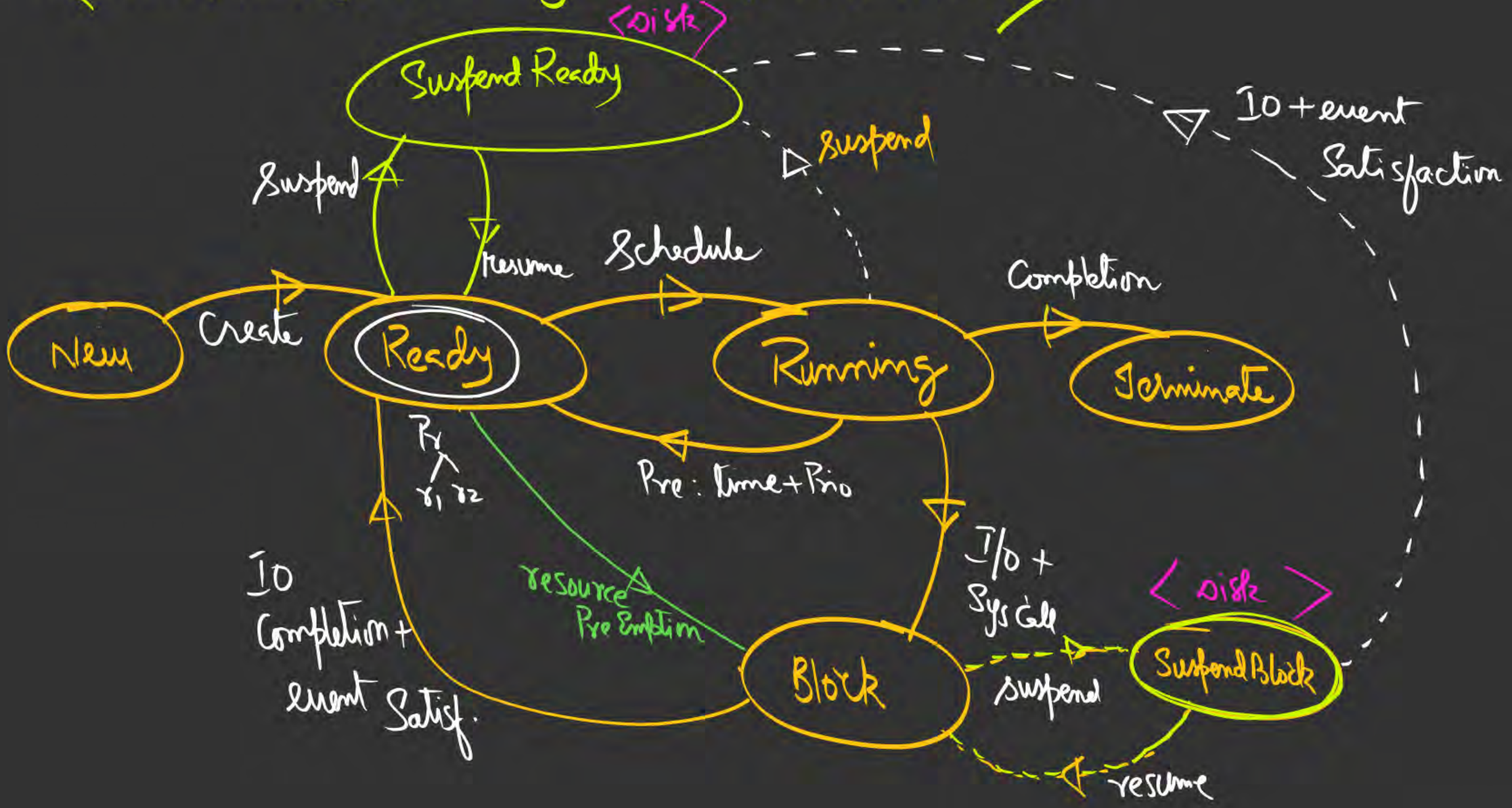
**By- Dr. Khaleel Khan sir**


- Process State Diagram
- Schedulers & Dispatcher
- Context Switching

Process Suspension

# Process State Transition Diagram

< New ; Ready ; Running ; Block ; Terminate >



→ Processes are suspended from memory onto disk for improving degrading Performance;

→ one can suspend Processes from Ready + Running + Block States;

→ When a Process in Ready State is PreEmpted of one/more resources, then it will make a Transition to Block State;

Q1) Processes can get Blocked from Running & Ready State(s);

Q2)

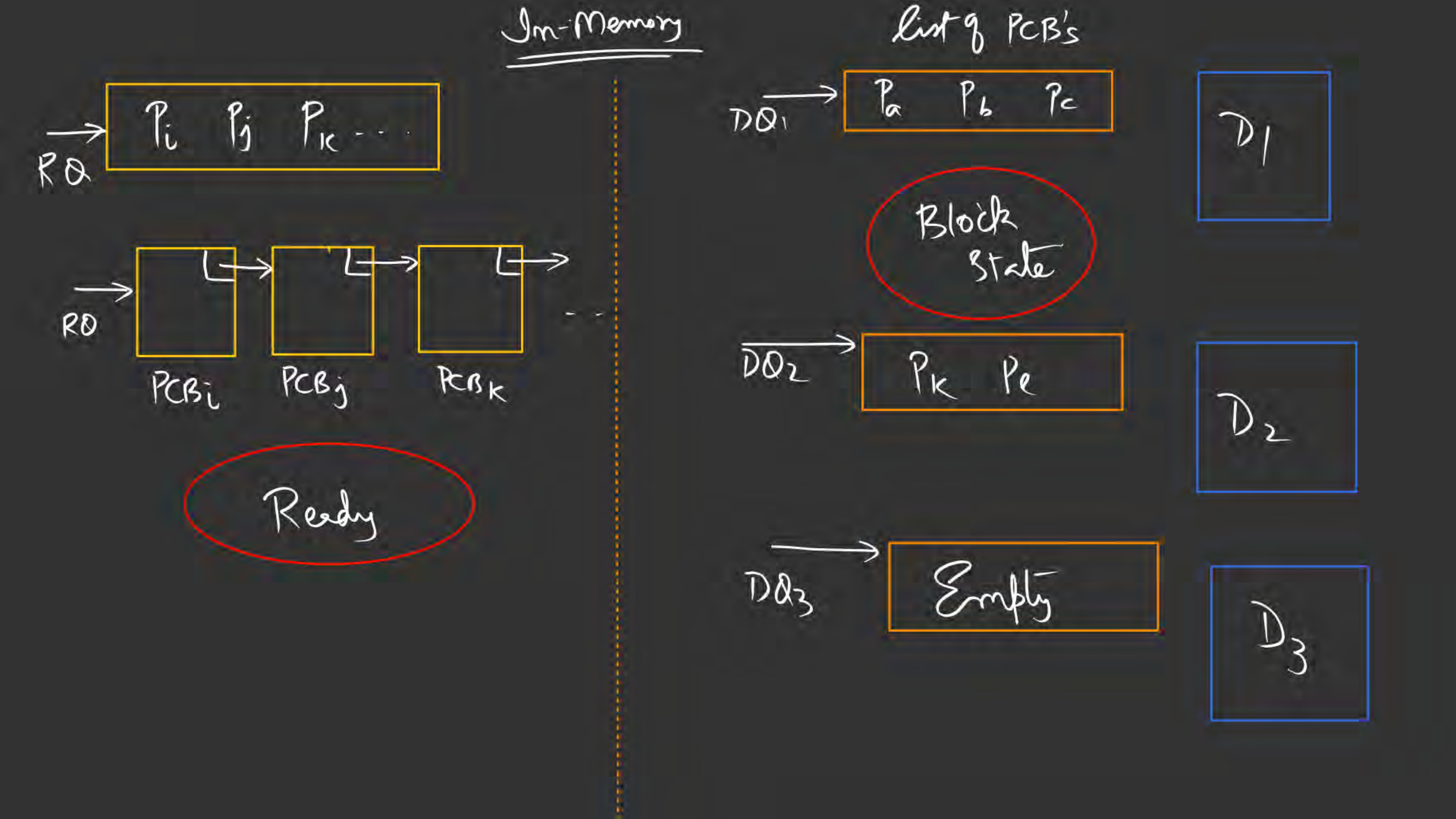
| | Event | Transition | |
|---|---|---|---|
| a) | Timerout | Running → Ready | ✓ |
| b) | I/O | Running → Block | ✓ |
| c) | Sys-Call Completion | Block → Running | ✗ |
| d) | Scheduler dispatch | Ready → Block | ✗ |

# Schedulers & Dispatcher :

## Scheduling Queues:

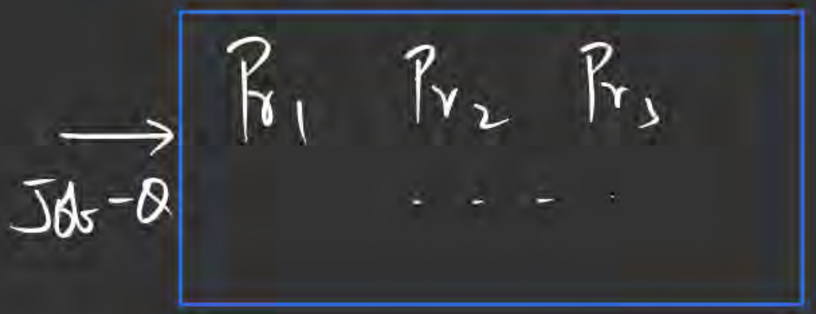
→ Every M. Pr. O.S maintains Two Types of Queues;

- In Memory
  - Ready - Q
    - < list of PCB's of Ready Processes >
  - Block - Q / Device - Q
    - < list of PCB's of Block Processes >
- on - Disk
  - Jobs - Q / Input - Q < Contains Programs to be Loaded in Memory >
  - Suspend - Q < Holds Suspended Processes >


In-Memory
List of PCB's
RQ
Pi Pj Pk ...
PCBi
PCBj
PCBk
Ready
DQ1
Pa Pb Pc
D1
Block State
DQ2
Pk Pe
D2
DQ3
Empty
D3

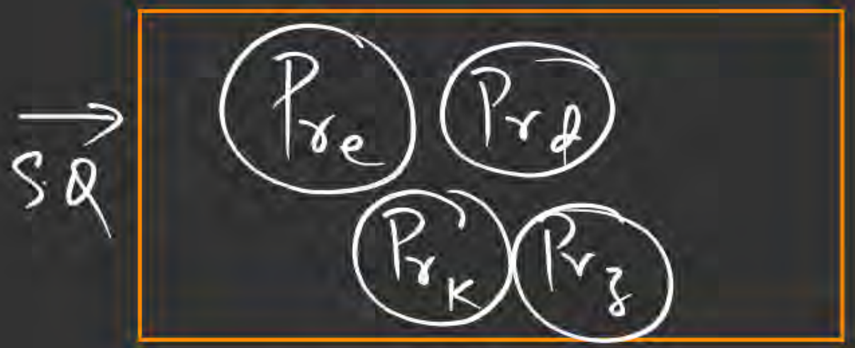
# On-disk Q's

## New State

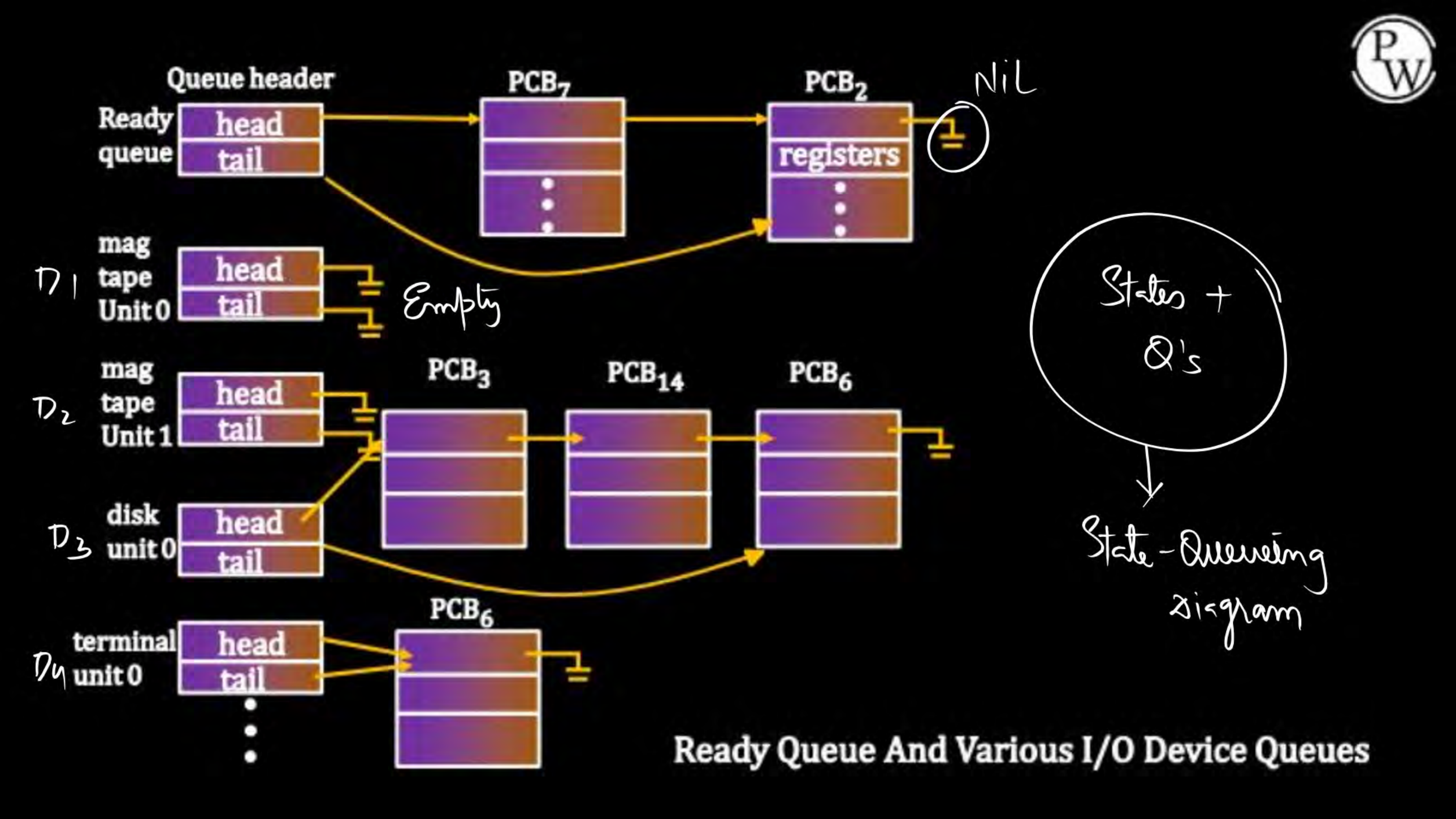



< Programs that are waiting to get Loaded in Memory for execution >

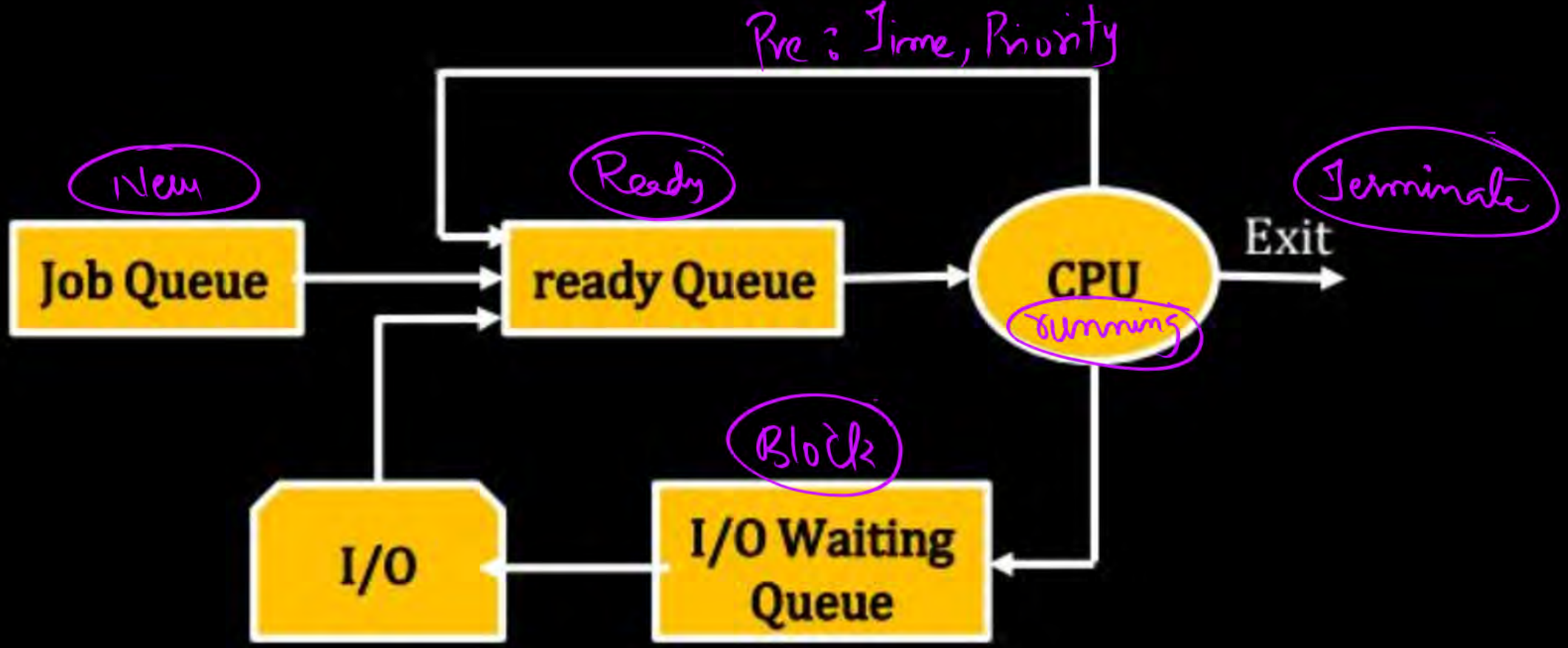
## Suspend State



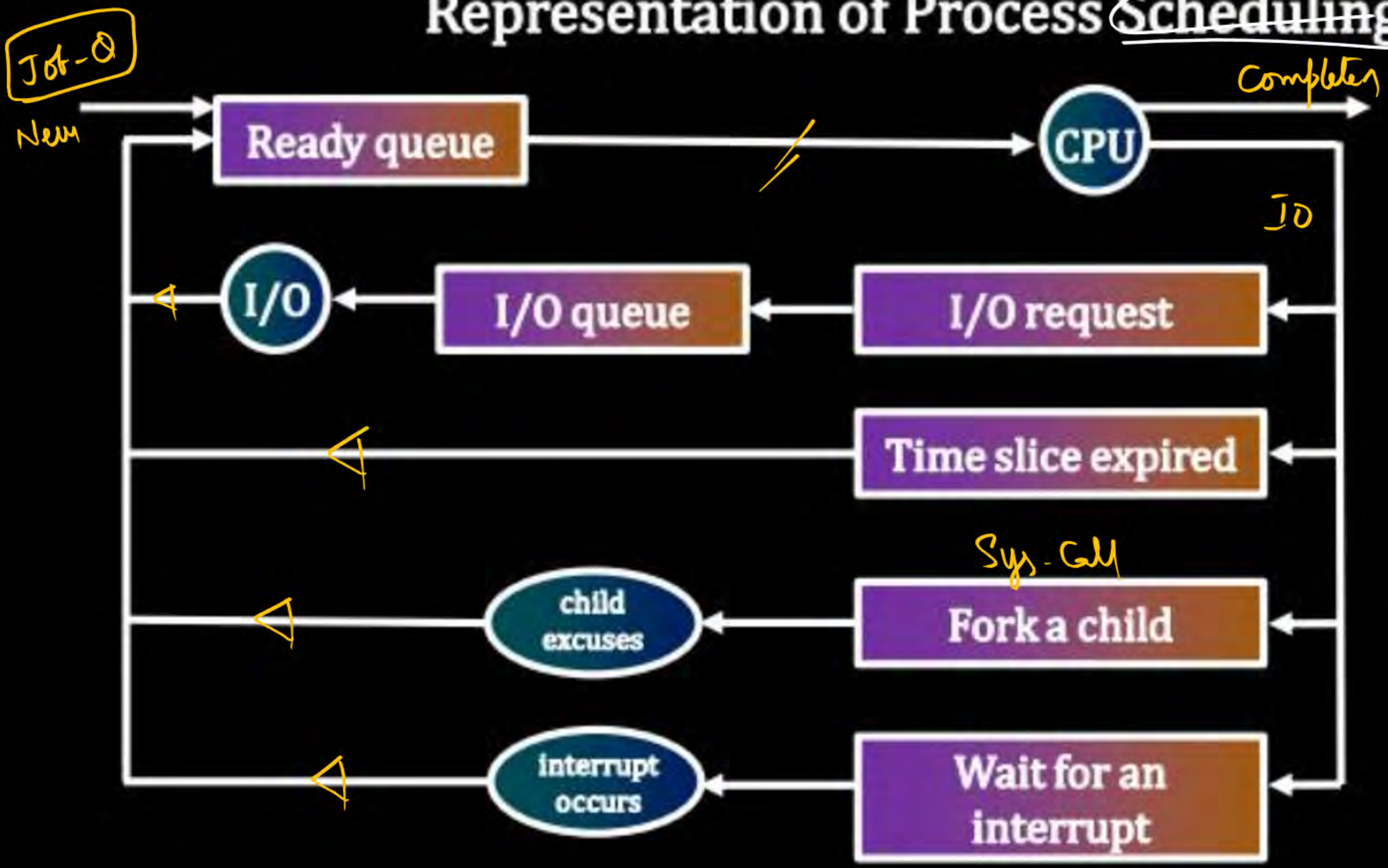

< Contains Suspended Processes from Memory >



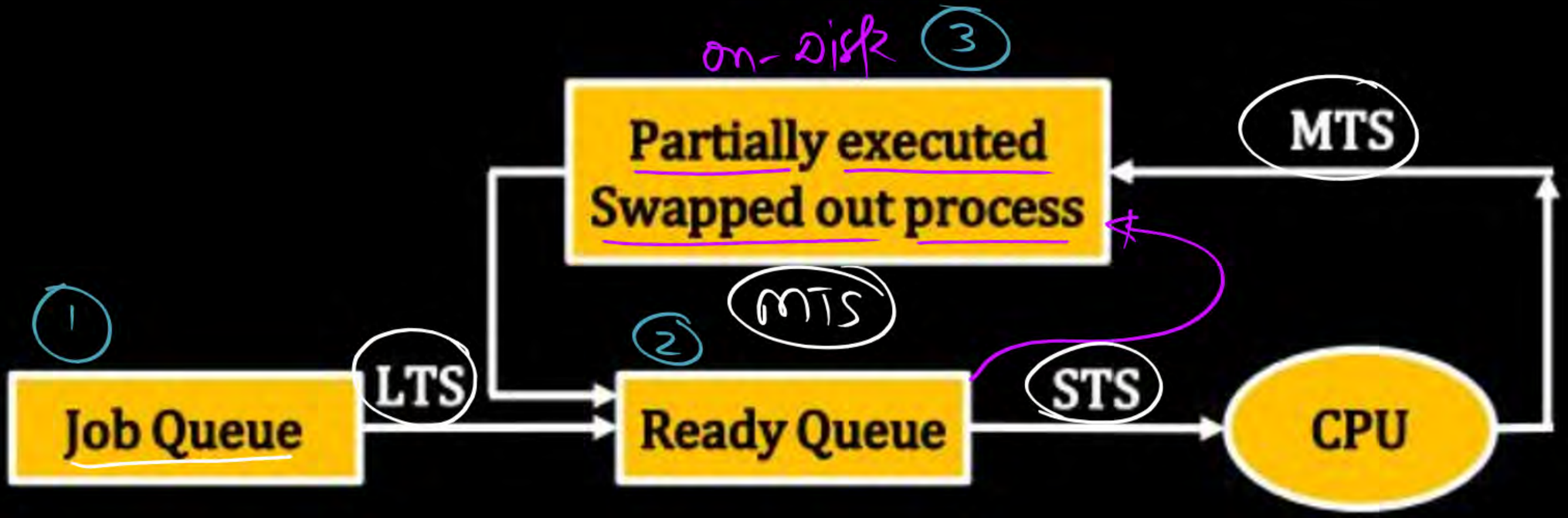

Ready Queue And Various I/O Device Queues


Pre : Time, Priority
New
Job Queue
Ready
ready Queue
CPU
Running
Exit
Terminate
Block
I/O
I/O Waiting Queue

# Representation of Process Scheduling



# Suspend-Q


on-disk
3
Partially executed
Swapped out process
MTS
MTS
1
2
LTS
Job Queue
Ready Queue
STS
CPU

# Schedulers:

→ Every M.Pr. O.S maintains atleast 3 - Schedulers, that make decisions on different Q's;

Schedule ~ to make a decision

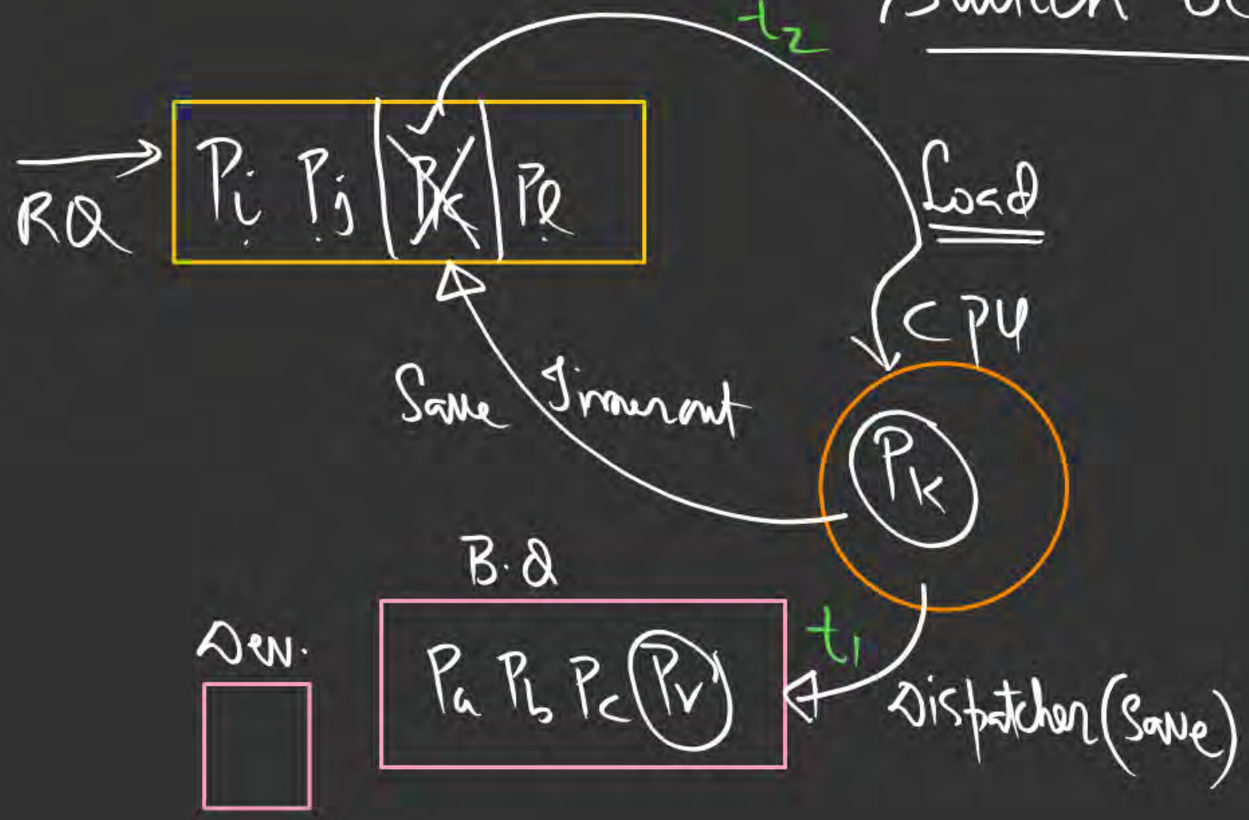
1 Long Term Scheduler (LTS): decides which Program(s) should get loaded in Memory; operates on Job-Q;

2 Short-Term Scheduler (STS): operates on Ready 'Q' to decide which ready Process should run next onto CPU;
<CPU-Scheduler>

3. Medium Term Scheduler (MTS): operates on Suspend 'Q' for carrying out the operations of Process Suspension & resumption

**Dispatcher**: Carries out the activity of **Context-switching**;

**Context-switching (CS)**: is an activity of Saving and loading the PCB's of Processes onto cpu and into **Block Q / Ready-Q**, whenever a **Process Switch occurs**;




The Total Time Needed to Save & Load the PCB's is known as **Context-switch Time** ($\delta$) also known as Scheduling overhead

$$\delta = t_1 + t_2$$

Q1) The degree of M.Pr is controlled by ____LTS____.

↓

No. of Processes registered with O.S

a) LTS ✓
b) STS
c) MTS
d) Dispatcher.

**Q.1**

Consider a System having 'n' CPUs ($n \geq 1$) and 'k' Processes ($k > n$).

Calculate lower bound and upper bound of the number of Processes that can be in the Ready, Running and Block states

| | L.B | U.B |
|---|---|---|
| Ready | 0 | k |
| Running | 0 | n |
| Block | 0 | k |


n = 1

K = 3


**Q.2** 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 IV only

B. I, II, III and IV

C. I, II and III only

D. II and III only

Q.3

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

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

B. Turnaround time includes waiting time

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.

**Q.4** A Processor needs Software Interrupt to

→ Sys-Call : to avail access O.S services

A. Test the Interrupt System of the Processor.

B. Implement Co-Routines.

C. Obtain system services which need execution of privileged instructions.

D. Return from subroutine.

Q.5

A CPU has two Modes-Privileged and Non-Privileged. In order to change the mode from Privileged to Non - Privileged.

A. A Hardware Interrupt is needed.

B. A Software Interrupt is needed.

C. A Privileged Instruction (which does not generate an interrupt) is needed.

D. A Non - Privileged Instruction (which does not generate an interrupt) is needed.

Q.6

System Calls are usually invoked by using:

A. A Software Interrupt

B. Polling

C. An Indirect jump

D. A Privileged Instruction.

Q.7

A part of the system S/W, which under all circumstances must reside in the Main Memory, is:

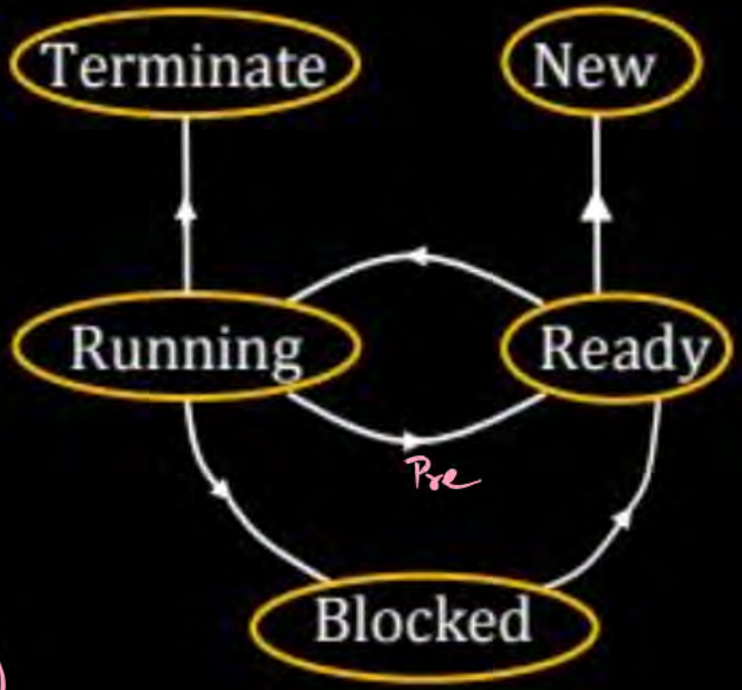
A. Text Editor

B. Assembler

C. Linker

D. Loader

**Q.8** The Process state Transition diagram given below is representative of




Is it U.Pr → O.S
M.Pr → O.S
Pr
Non-Pr

A ✗ A Batch O.S. (U.Pr)

B ✓ An O.S. with a preemptive scheduler

C ✗ An O.S with a non-preemptive scheduler

D ✗ A Uniprogrammed O.S.

**Q.9**

In the following process state transition diagram for a uniprocessor system, assume that there are always some processes in the ready state:

Now consider the following statements:

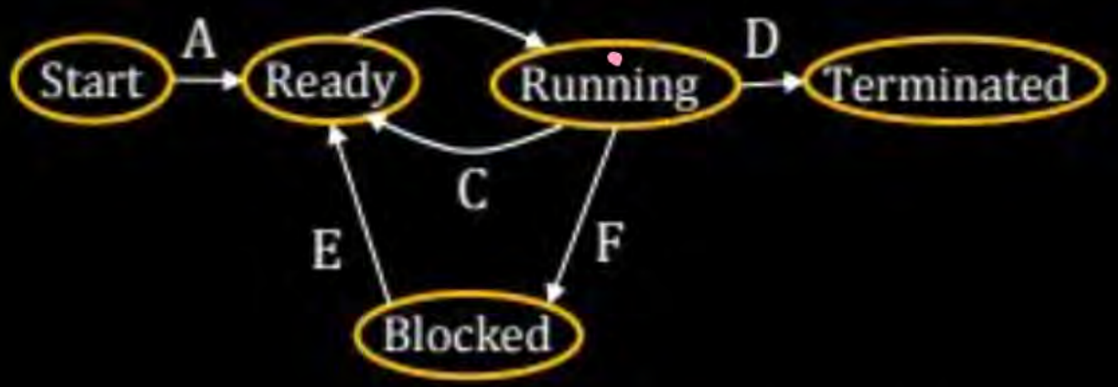
(I) If a process makes a transition D, it would result in another process making transition A immediately.

(II) A process P2 in blocked state can make transition E while another process P1 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

**Q.10** Which combination of the following feature will suffice to characterize an OS as a multi-programmed OS?

(a) More than one program may be loaded into main memory at the same time for execution.

(b) If a program waits for certain events such as I/O, another program is immediately scheduled for execution,

(c) If the execution of program terminates, another program is immediately scheduled for execution

(A) a

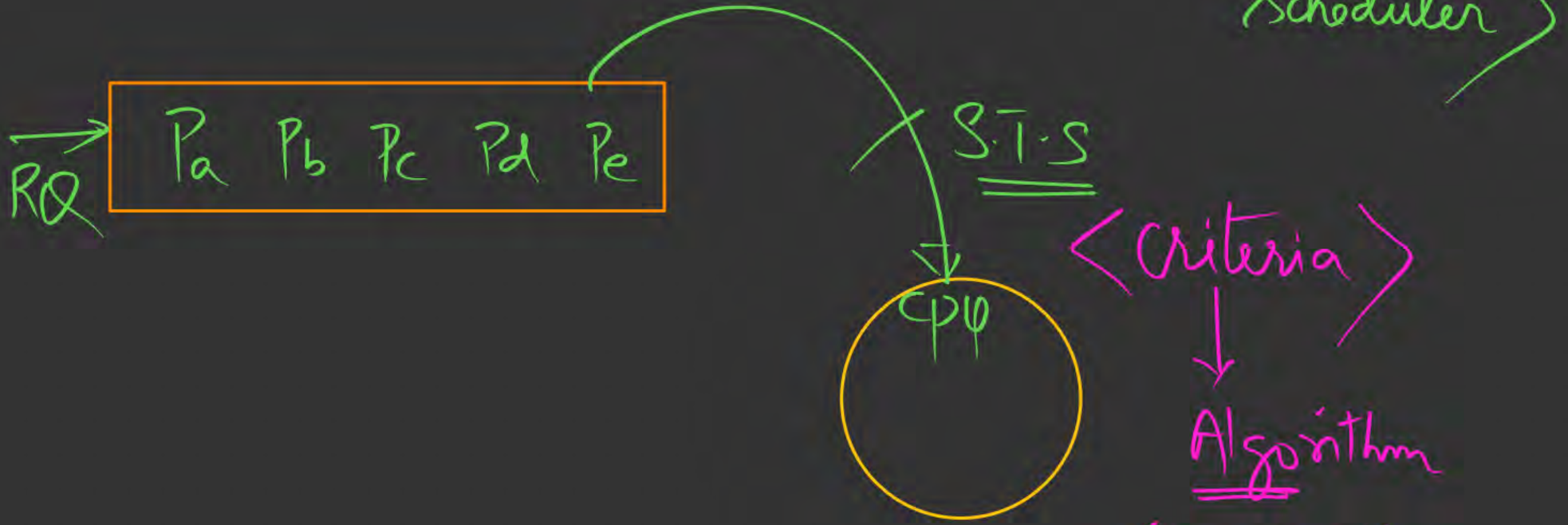
(B) a and b

(C) a and c

(D) a, b and c

# 2) ⊛ CPU - Scheduling [90%]

< Design & Implementation of Short Term Scheduler >




< Criteria >

↓

Algorithm

< CPU Schdlng Technique >


THANK YOU
GW
SOLDIERS !