unacademy

11:00 1.5 hours

2:00 PM 2 hours

Tom. Same time 2 classes

Sat. ?

# Operating System Revision: Part 1

Special class

Vishvadeep Gothi • June 8, 2023

unacademy

# Operating System: Basics & Process Management

By: Vishvadeep Gothi

unacademy

# OS

- Software abstracting hardware
- Interface between user and hardware
- Set of utilities to simplify application development/execution
- Control program
- Acts like a government

unacademy

# OS

| Chapter Number | Chapter Name |
|---|---|
| 1 | Introduction |
| 2 | Process Management |
| 3 | CPU Scheduling |
| 4 | Process Synchronization |
| 5 | Deadlock |
| 6 | Memory Management & Virtual Memory |
| 7 | File System |
| 8 | Disk Scheduling |

unacademy

# Services of OS

- User Interface
- Program Execution
- I/O Operation
- File-System Manipulation
- Communication (Inter-process Communication)
- Error Detection
- Resource Allocation
- Accounting
- Protection & Security

unacademy

# Goals of OS

- Convenience (User-friendly)
- Efficiency
- Portability
- Reliability
- Scalability
- Robustness

unacademy

# Types of OS

1. Uniprogramming OS
2. Multiprogramming OS → Non-preemptive, Preemptive
3. Multitasking OS (Time Sharing) → R.R.
4. Multiprocessing OS → multiple CPU
5. Multiuser OS → only linux, Unix
6. Real Time OS → Hard R.O.S., Soft R.O.S.
7. Embedded OS
8. Handheld Device OS

unacademy

# System Call

A system call is a way for programs to interact with the operating system

unacademy

# Parts of OS


G.U.I.
C.L.I.
Batch interface

unacademy

# Dual Mode of Operation

**2 modes:**

User Mode (mode bit = 1)

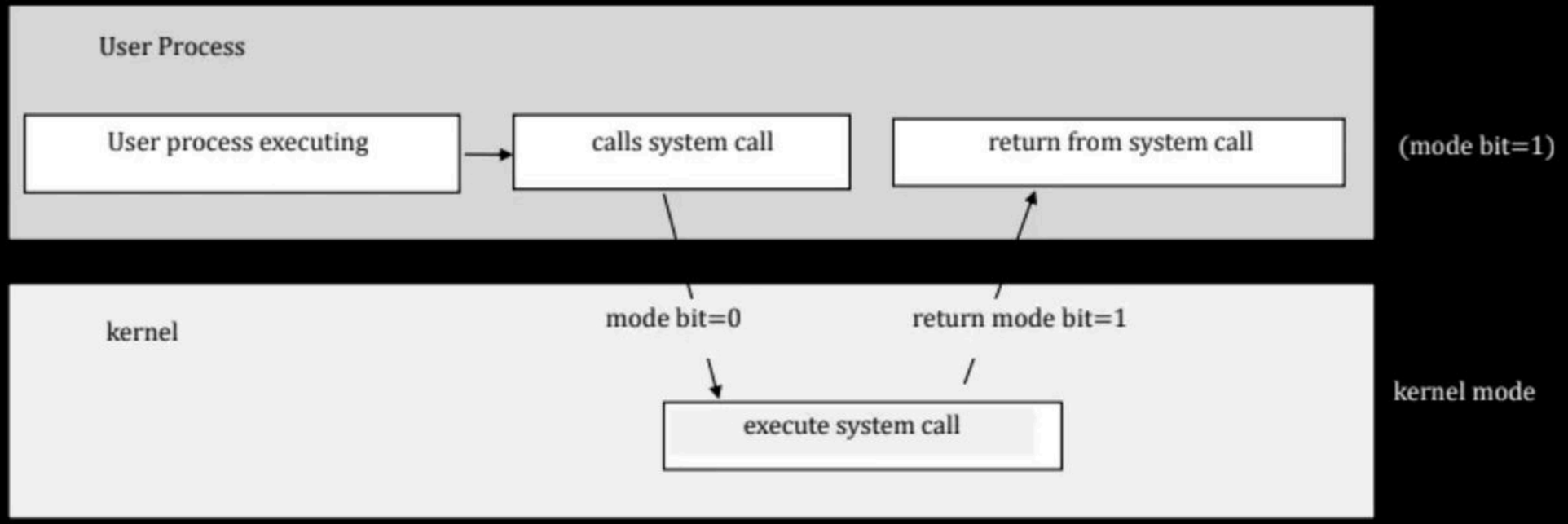
Kernel/System/Supervisor/Privileged Mode (mode bit = 0)

# Dual Mode of Operation

**2 modes:**

User Mode (mode bit = 1)

Kernel/System/Supervisor/Privileged Mode (mode bit = 0)



unacademy

# Process

↳ Prog. + Run-time activities

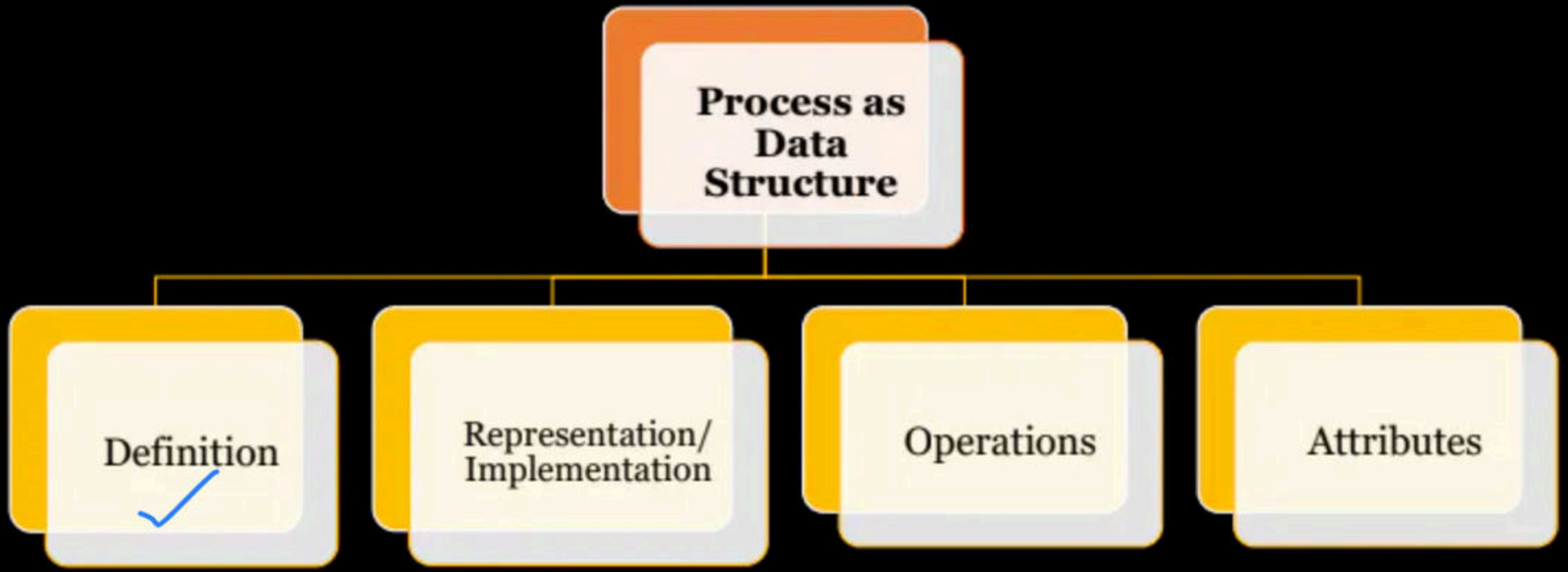
**Process:**

- Program under execution
- An instance of a program
- Schedulable/Dispatchable unit (CPU)
- Unit of execution (CPU)
- Locus of control (OS)

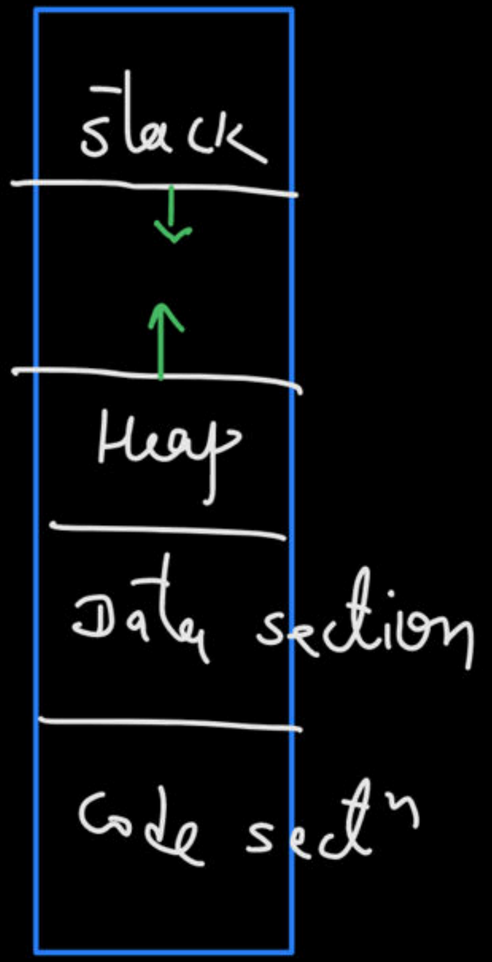
# Process

- **Process:**
  - Program under execution
  - An instance of a program
  - Schedulable/Dispatchable unit (CPU)
  - Unit of execution (CPU)
  - Locus of control (OS)

# Process as Data Structure



# Representation of a Process

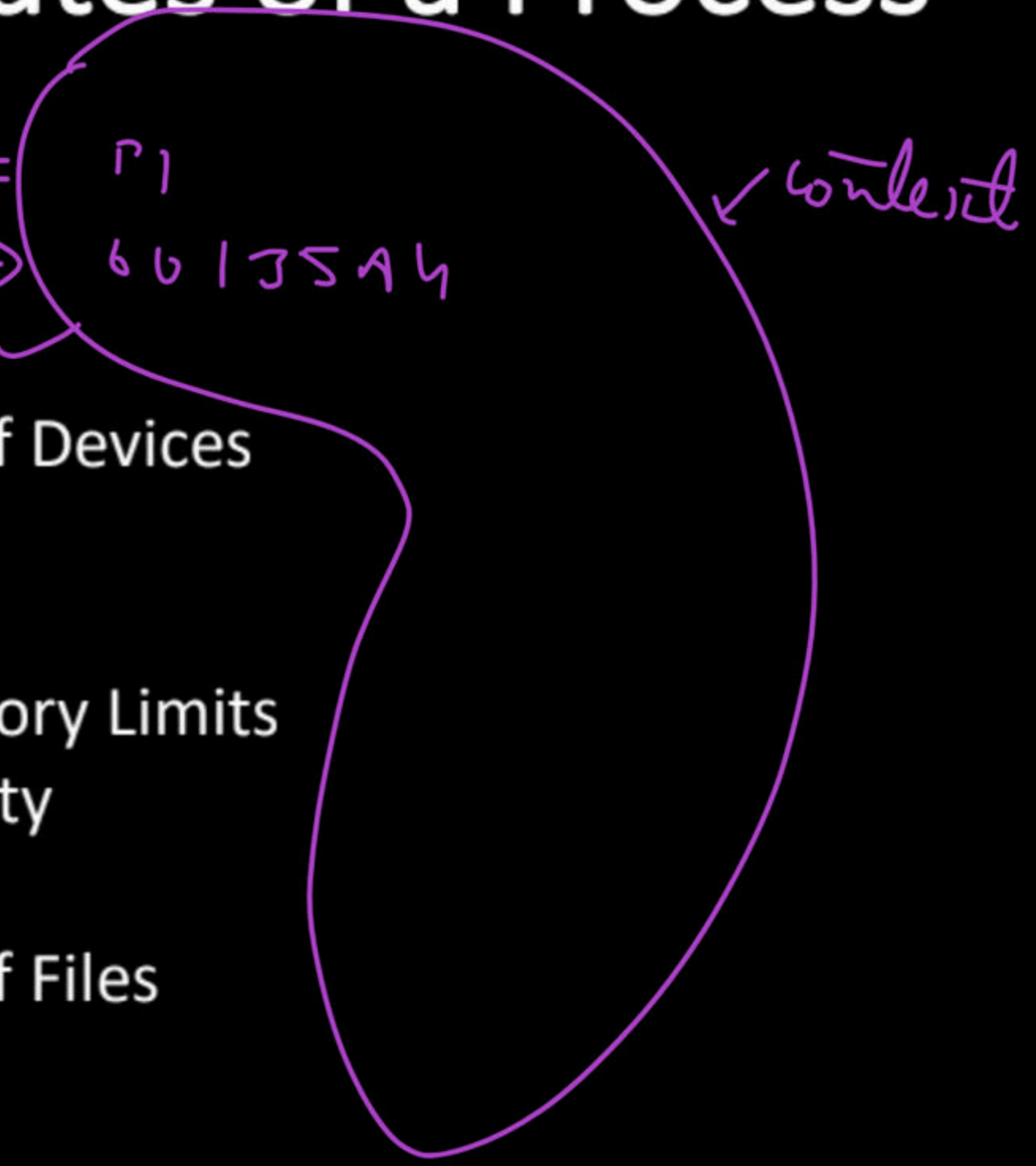


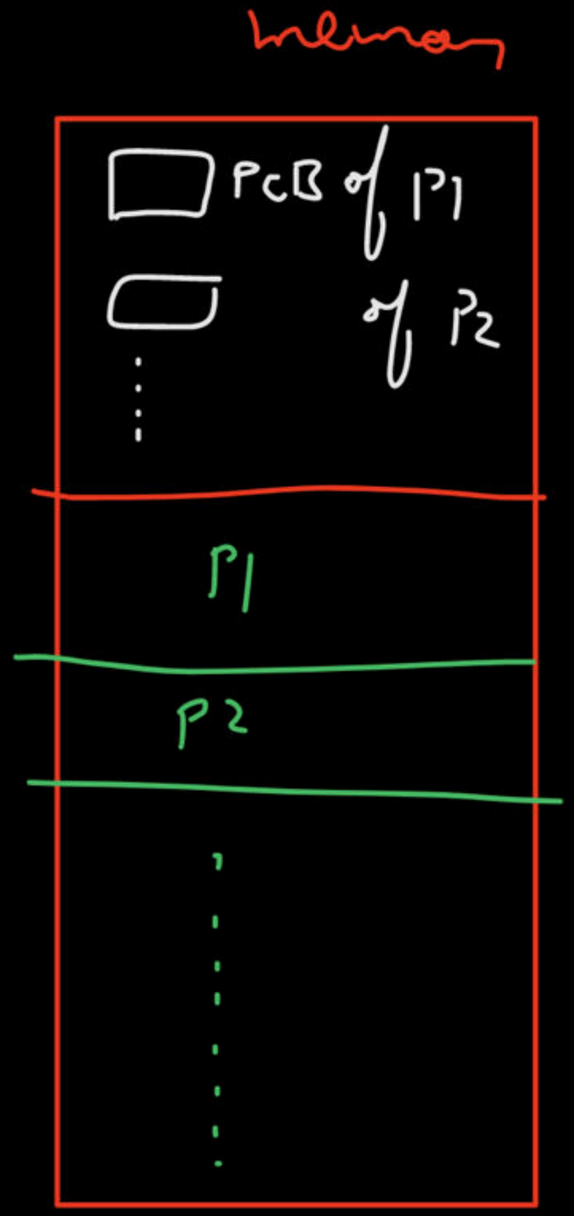
# Operations on a Process

- Create (Resource Allocation)
- Schedule, Run
- Wait/Block
- Suspend, Resume
- Terminate (Resource Deallocation)

# Attributes of a Process

- PID
- PC
- GPR
- List of Devices
- Type
- Size
- Memory Limits
- Priority
- State
- List of Files



# PCB (Process Control Block)

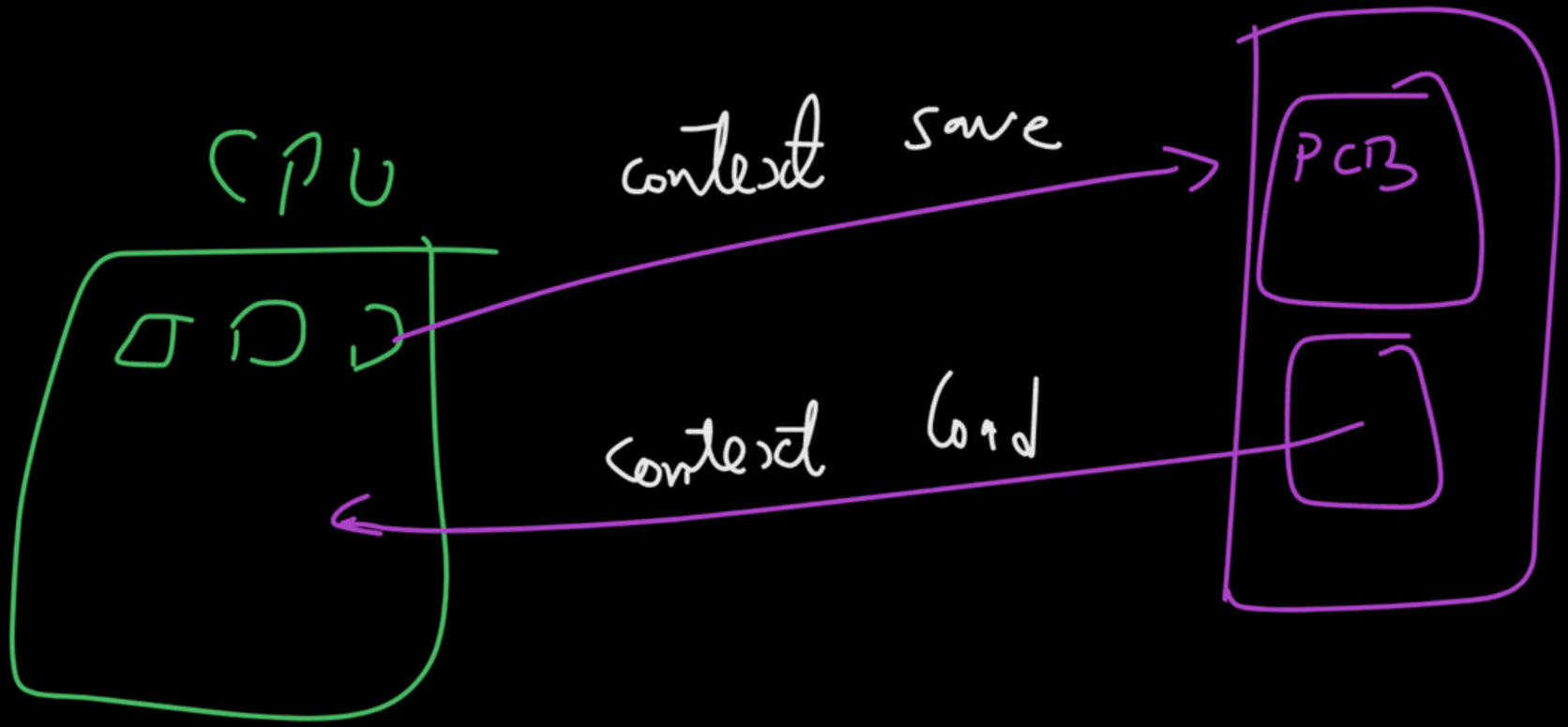
- Also known as processor descriptor

# Context

The content of PCB of a process are collectively know as ‘Context’ of that process

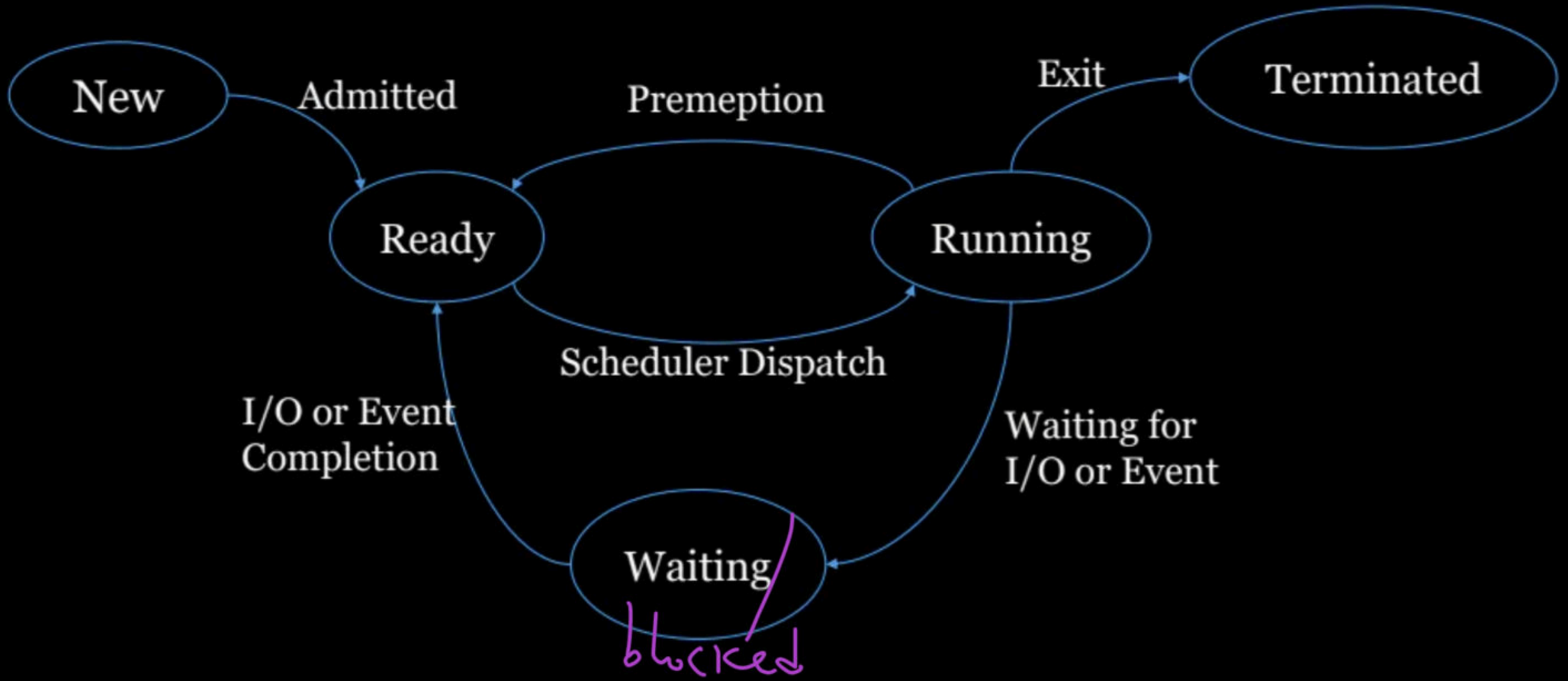
# Context Switch

dispatcher does



# Process States

# Process States



# Process States

**New:**

All installed processes are in known to be in new state

**Ready:**

All processes which are waiting to run on CPU are known to be in ready state

# Process States

**Running:**
A process which is running on CPU has its state as running

**Terminated:**
A completed process has its state as terminated

**Blocked:**
All processes which are waiting for any IO or event

# Process States Transitions

**New To Ready:** When process is admitted by OS

**Ready to Running:** When a process is dispatched to CPU

**Running to Terminated:** When a process is completed

**Running to Blocked:** When a process goes for IO or event

**Running to Ready:** When a process is preempted

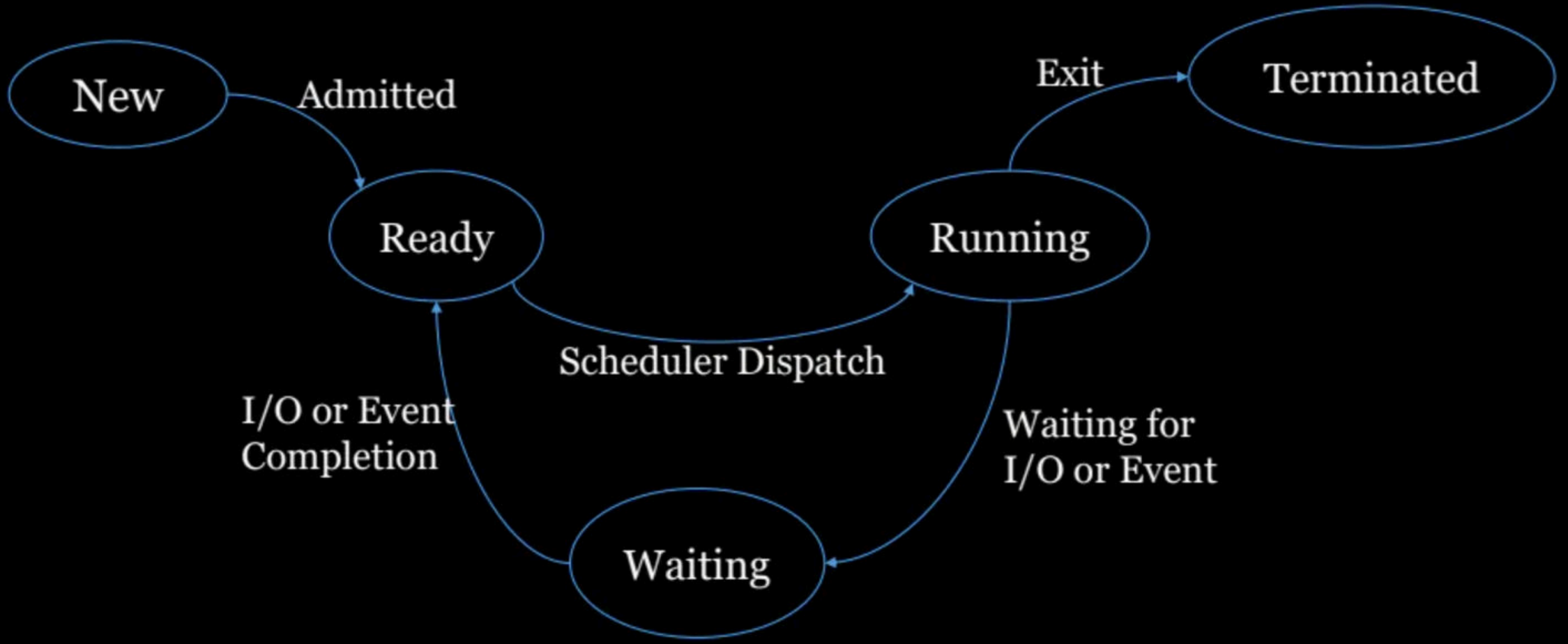
**Blocked to Ready:** When a process completes IO or event

# Process States Transitions

2 Transitions are voluntary:

- Running to Terminated
- Running to Blocked

# Process States: Non-preemptive



# CPU vs IO Bound Process

CPU extensive

**CPU Bound:** If the process is intensive in terms of CPU operations

**IO Bound:** If the process is intensive in terms of IO operations

I/O extensive

# Scheduling Queues

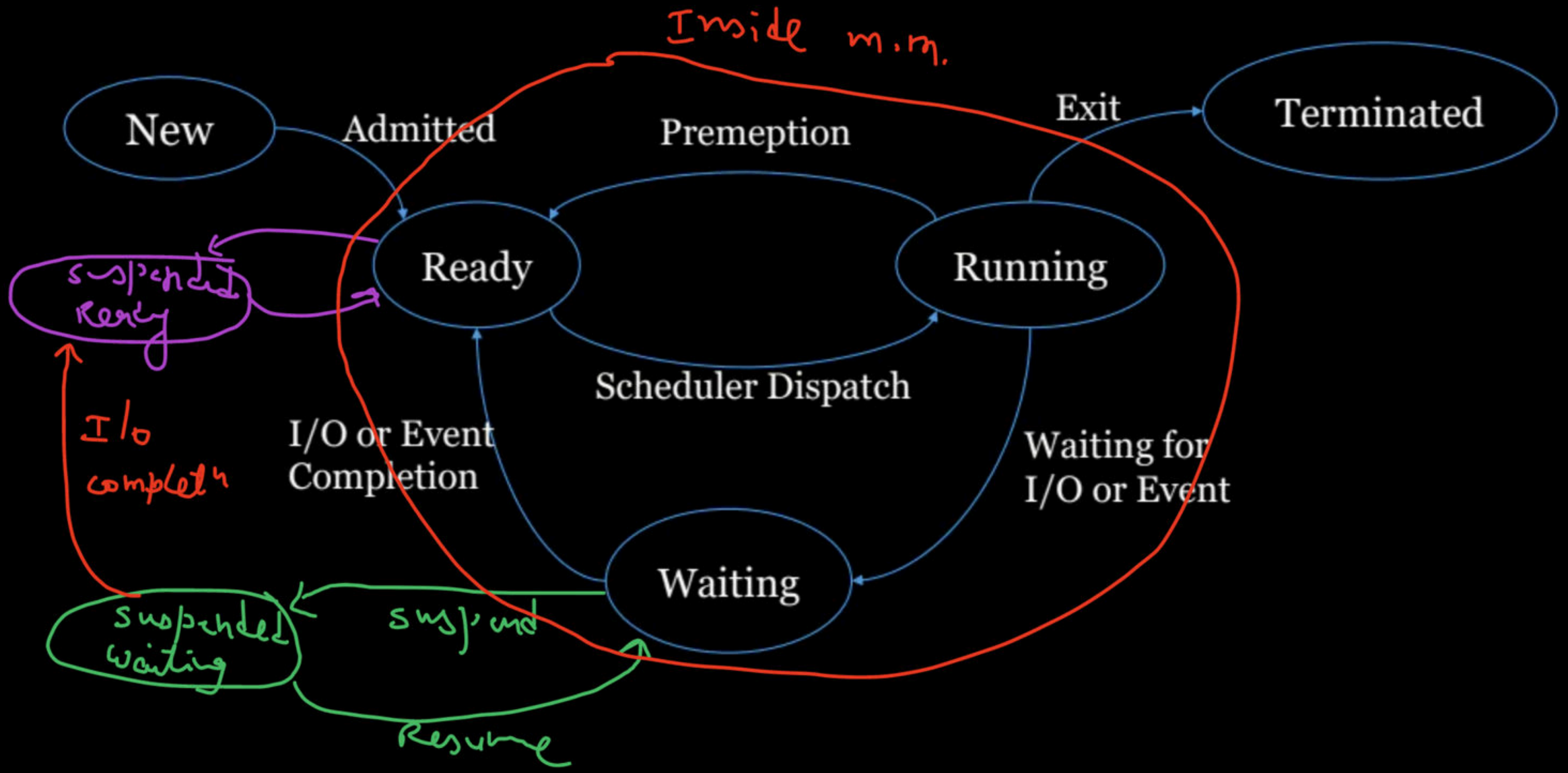
- Job Queue → new state → single
- Ready Queue → Ready state → single
- Device Queue
  - ↳ blocked state → one for each device

# Scheduling Queues

# Types of Schedulers

1. Long-Term Scheduler (Job)
2. Short-Term Scheduler (CPU)
3. Mid-Term Scheduler (Medium-term) ⟶ swapping

# Updated Process States



# CPU Scheduling

**Function:**

Make a selection of a process to run next on CPU.
among all ready processes.

# CPU Scheduling

**Function:**

Make a selection

**Goal**

- Minimize Wait time and Turn-around time
- Maximize CPU utilization (Throughput)
- Fairness

SJF, SRTF

→ no starvation

(FCFS, RR, HRRN)

- minimize avg. response time
- maximize interactivness

} → RR

# Scheduling Times

1. Arrival Time (AT)
2. Burst Time (BT)
3. Completion Time(CT)
4. Turnaround Time (TAT) = CT − AT
5. Waiting Time (WT) = TAT − BT

# Scheduling Times

6. Response Time (RT):
7. Deadline (D):
8. Scheduling Length (L):
9. Throughput:

# CPU Scheduling: Types

CPU Scheduling:

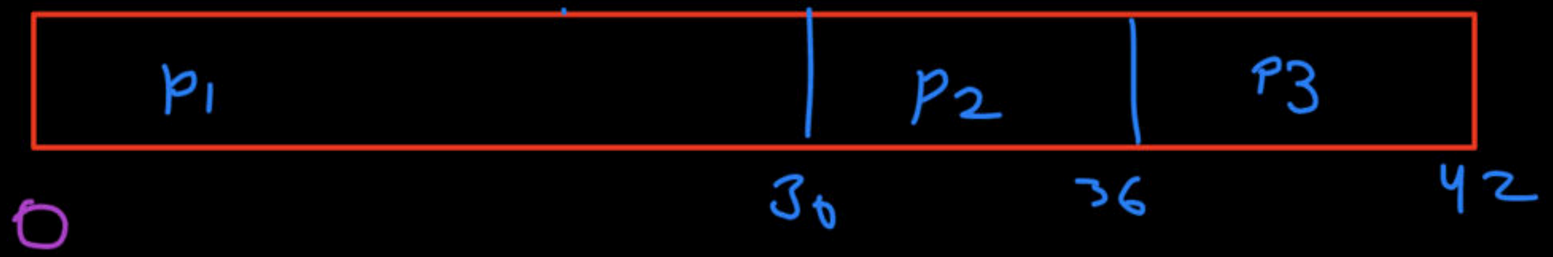
1. Preemptive
2. Non-preemptive

# FCFS (First Come First Serve)

- Criteria: min AT first
  - Tie-breaker: min. process ID first
- Type: Non-preemptive

# FCFS (First Come First Serve)

| Process | AT | Burst Time |
|---|---|---|
| P1 | 0 | 30 |
| P2 | 6 | 6 |
| P3 | 0 | 6 |

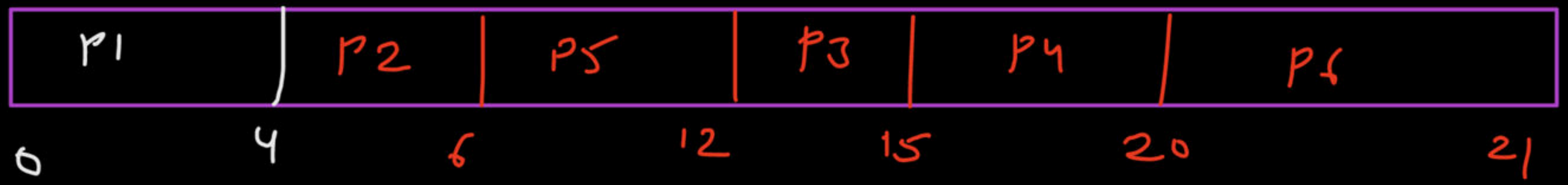


# FCFS (First Come First Serve)

| Process | Arrival Time | Burst Time | Completion Time | Turnaround Time | Waiting Time |
|---|---|---|---|---|---|
| P1 | 0 | 30 | 30 | 30 | 0 |
| P2 | 0 | 6 | 36 | 36 | 30 |
| P3 | 0 | 6 | 42 | 42 | 36 |

# FCFS (First Come First Serve)

| Process | Arrival Time | Burst Time | CT | TAT | WT |
|---|---|---|---|---|---|
| P1 | 0 | 4 | 4 | 4 | 0 |
| P2 | 1 | 2 | 6 | 5 | 3 |
| P3 | 4 | 3 | 15 | 11 | 8 |
| P4 | 5 | 5 | 20 | 15 | 10 |
| P5 | 2 | 6 | 12 | 10 | 4 |
| P6 | 9 | 1 | 21 | 12 | 11 |

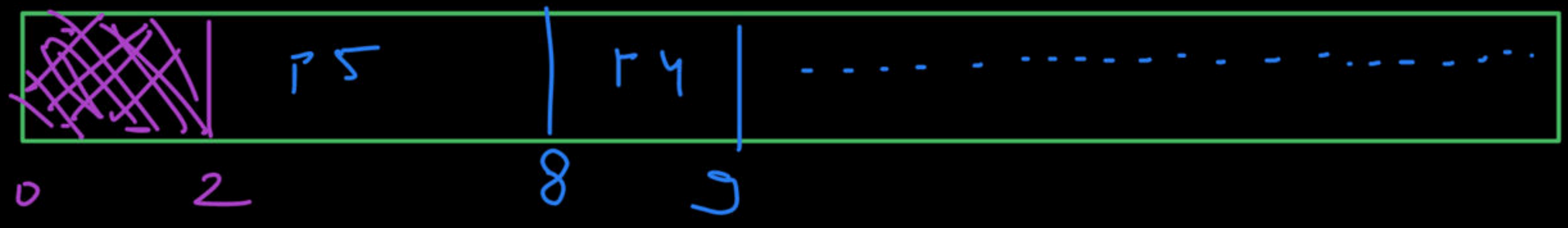


# FCFS (First Come First Serve)

| Process | Arrival Time | Burst Time | Completion Time | Turnaround Time | Waiting Time |
|---|---|---|---|---|---|
| P1 | 0 | 4 | | | |
| P2 | 1 | 2 | | | |
| P3 | 2 | 3 | | | |
| P4 | 3 | 5 | | | |
| P5 | 4 | 6 | | | |
| P6 | 5 | 1 | | | |

# FCFS (First Come First Serve)

| Process | Arrival Time | Burst Time |
|---|---|---|
| P1 | 4 | 4 |
| P2 | 8 | 2 |
| P3 | 6 | 3 |
| P4 | 3 | 1 |
| P5 | 2 | 6 |
| P6 | 7 | 7 |



# FCFS (First Come First Serve)

| Process | Arrival Time | Burst Time | Completion Time | Turnaround Time | Waiting Time |
|---|---|---|---|---|---|
| P1 | 4 | 4 | | | |
| P2 | 8 | 2 | | | |
| P3 | 6 | 3 | | | |
| P4 | 5 | 3 | | | |
| P5 | 2 | 1 | | | |
| P6 | 7 | 7 | | | |

# Convoy Effect

# FCFS (First Come First Serve)

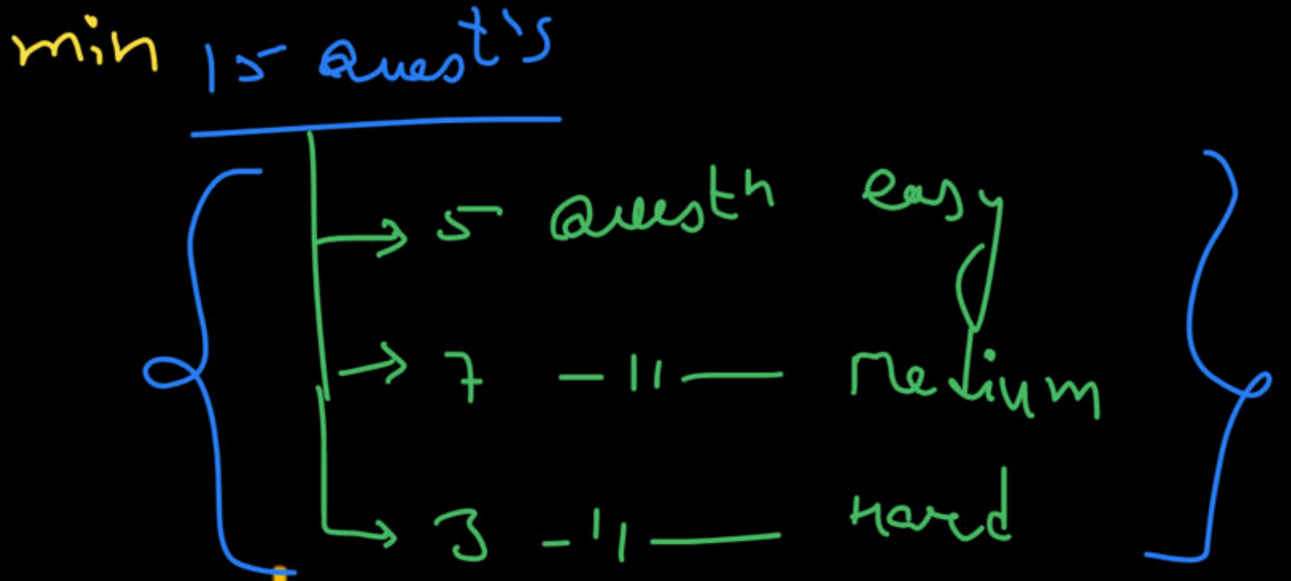
Advantages:

1. Easy to implement
2. No complex logic
3. No starvation / fairness

Disadvantages:

1. No option of Preemption
2. Convoy effect makes the system slow

unacademy




# Happy Learning.!