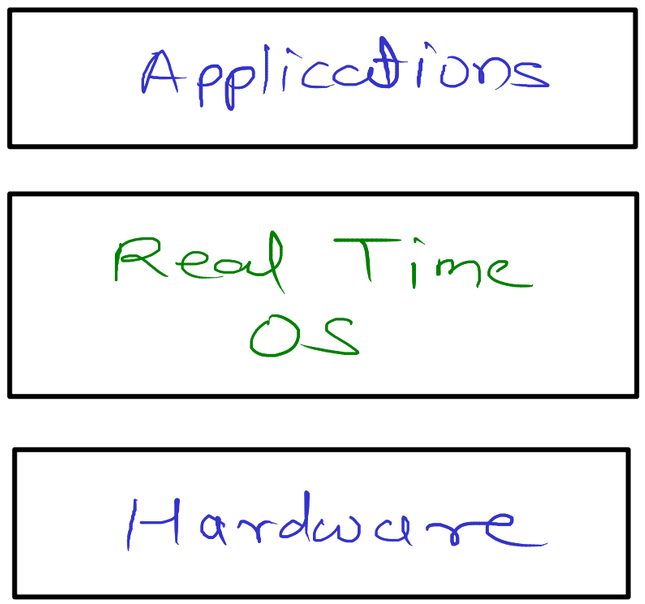
Real Time Operating System



* result should be accurate
* result should be calculated in minimal/deterministic

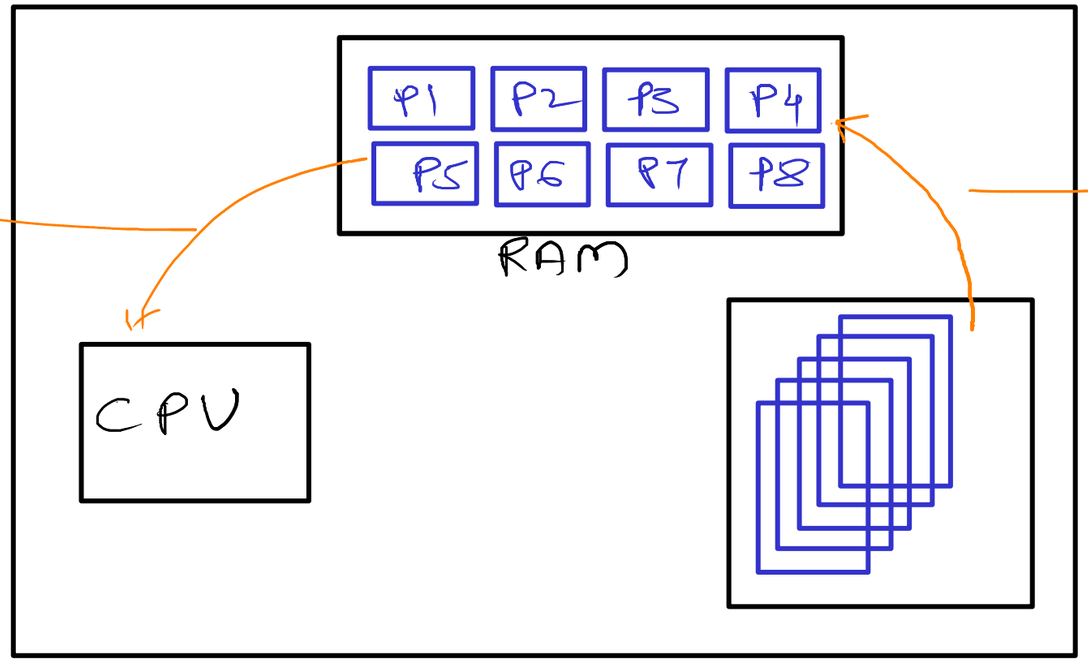
Types of RTOS

1. Hard Real Time OS
2. Soft Real Time OS
3. Firm Real Time OS

Functions of RTOS

1. Hardware Abstraction
2. Task Management
3. Memory Management
4. CPU Scheduling
5. Interrupt Handling

CPU Scheduling



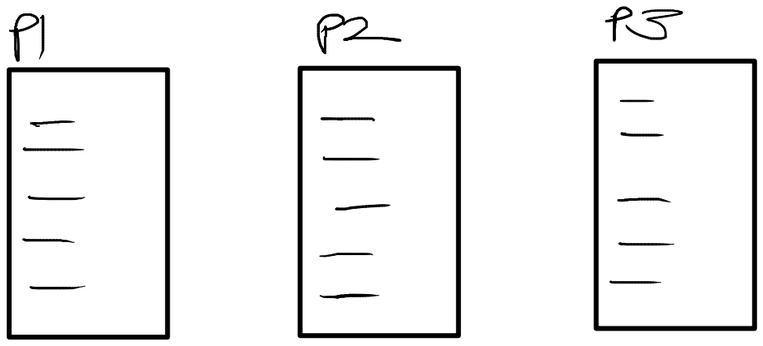
CPU scheduler (select one process from ready queue to execute on CPU)

Job scheduler (select mixture of CPU & IO bound processes)

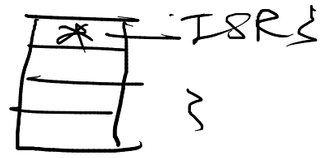
1) Multiprogramming
2) Multitasking (time sharing)

OS data structures
1) Job queue / process list
2) Ready queue ←
3) Waiting queues

Algorithm:
1) FCFS
2) SJF
3) Priority
4) RR
5) Fair Share

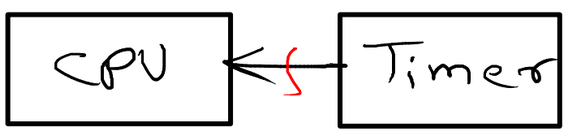


interrupt_handler( ) {
1) Save execution context of current save into its PCB
2) find address of ISR from IVT
3) call ISR
4) pid = cpu_scheduler( );
5) cpu_dispatcher(pid);
}

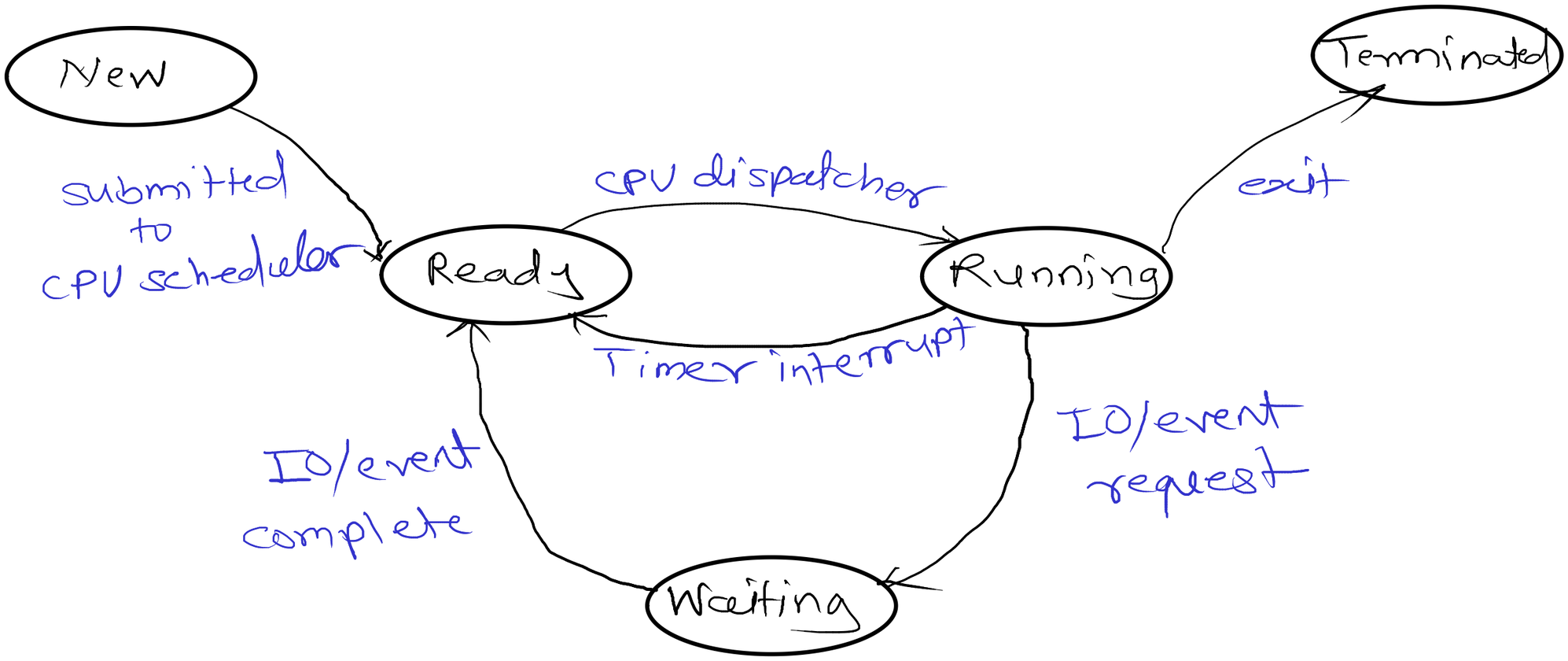


int CPU_scheduler( ) {
if(remaining > 0)
select same process
to be executed next
else
select new process
to be executed next
return pid;

cpu_dispatcher(){
restore execution
context of
selected process
}



Process Life Cycle



Types of Scheduling

(voluntarily)
1) Running – Terminated
2) Running – Waiting
(forcefully)
3) Running – Ready
4) Waiting – Ready

} Non-Preemptive scheduling

} Preemptive scheduling

CPU Scheduling Criterias

1. CPU Utilization (Ideal : Max)

- desktop system — 70%
- server system — 90%

2. Throughput (Ideal : Max)

- Amount of work done in unit time

3. Waiting Time (Ideal : Min)

- waiting for CPU
- total time spent by process into ready queue

4. Response Time (Ideal : Min)

- time from arival of process into ready queue upto first time getting scheduled

5. Turn Around Time (Ideal : Min)

- total time of process spent into memory
- time from creation to termination of process

$$TAT = \underset{\text{waiting time}}{\text{CPU}} + \underset{\text{burst}}{\text{CPU}} + \underset{\text{waiting time}}{\text{IO}} + \underset{\text{burst}}{\text{IO}}$$

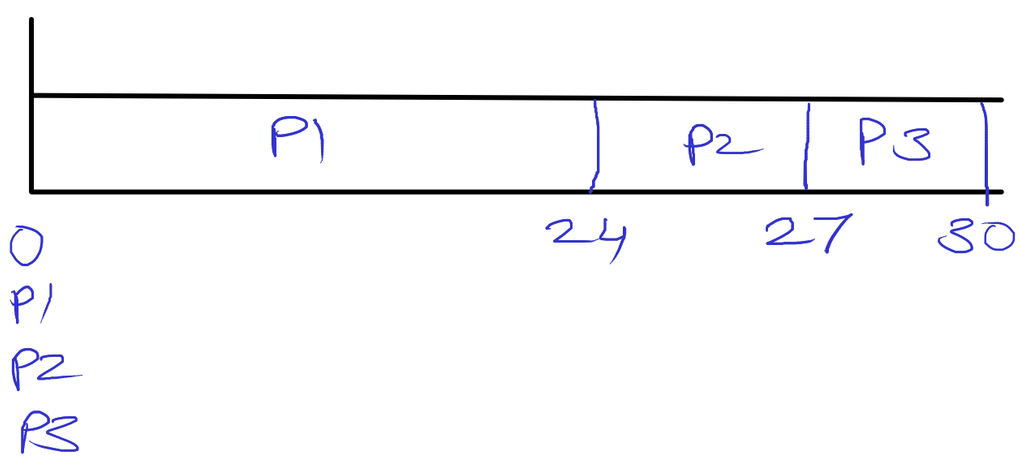
FCFS (First Come First Serve)

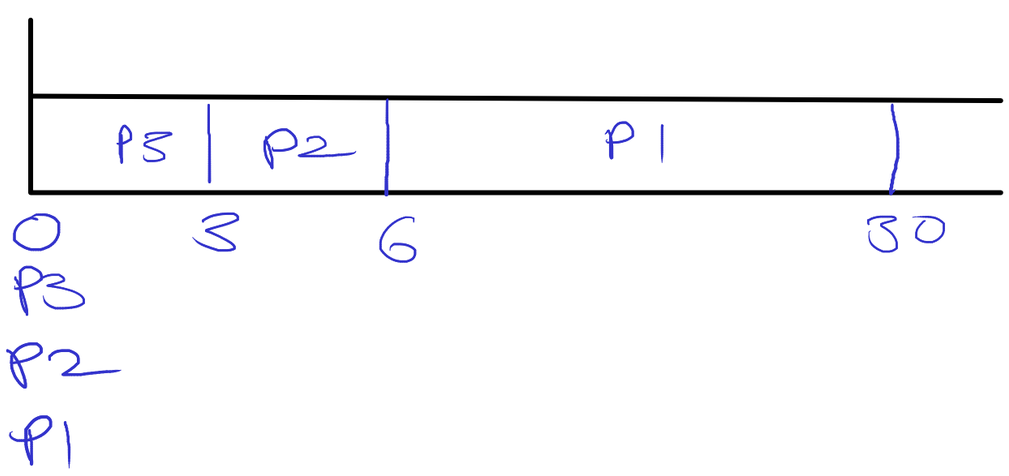
(Non-preemptive)

Process	Arrival	CPU Burst	WT	RT	TAT
P1	0	24	0	0	24
P2	0	3	24	24	27
P3	0	3	27	27	30

Process	Arrival	CPU Burst	WT	RT	TAT
P3	0	3	0	0	3
P2	0	3	3	3	6
P1	0	24	6	6	30

Gantt's chart





Convoy Effect

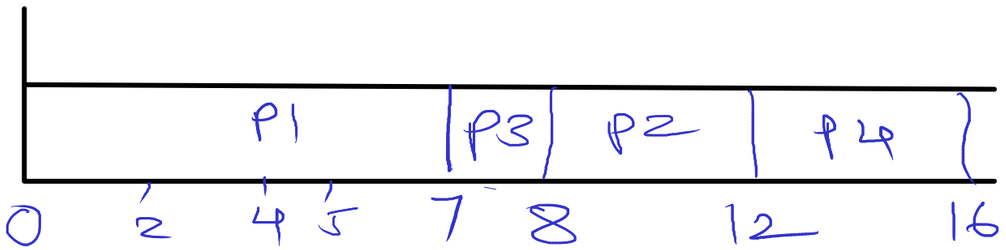
- due to arrival of longer process early, all other processes has to wait for longer time to get CPU access

- we can not control the sequenece of process arival into ready queue

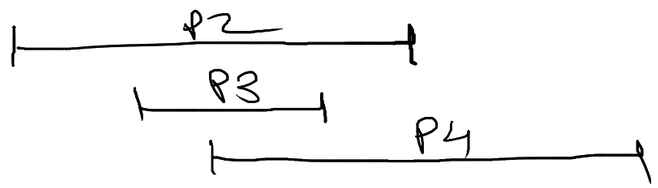
SJF (Shortest Job First)

(Non-Preemptive)

Process	Arrival	CPU Burst	WT	RT	TAT
P1	0	7	0	0	7
P2	2	4	6	6	10
P3	4	1	3	3	4
P4	5	4	7	7	11



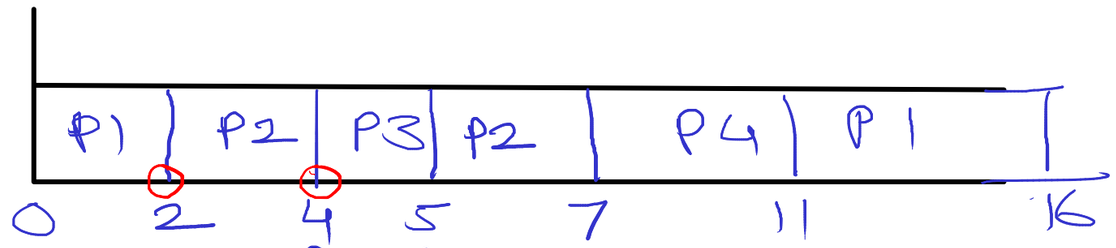
P1 P2 P3 P4



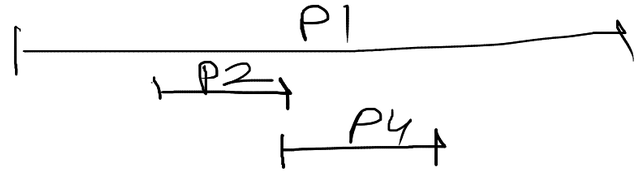
(Preemptive)

(Shortest Remaing Time First)

Process	Arrival	CPU Burst	Remain time	WT	RT	TAT
P1	0	7	5	9	0	16
P2	2	4	2	1	0	5
P3	4	1	0	0	0	1
P4	5	4	4	2	2	6



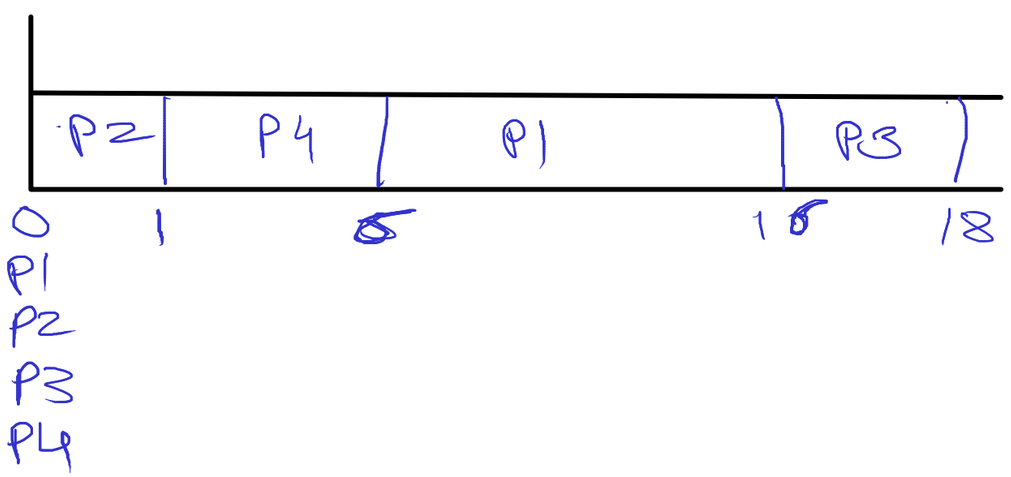
P1 P2 P3 P4



Priority

(Non-Preemptive)

Process	Arrival	CPU Burst	Priority	WT	RT	TAT
P1	0	10	3	6	6	16
P2	0	1	1 (H)	0	0	1
P3	0	2	4 (L)	16	16	18
P4	0	5	2	1	1	6



(Preemptive)

Process	Arrival	CPU Burst	Priority	WT	RT	TAT
P1	0	10	3	6	6	16
P2	1	1	1	0	0	1
P3	3	2	4	13	13	15
P4	0	5	2	1	0	6

4