First Come First Served

- First Come First Served (FCFS) is the simplest non-preemptive scheduling algorithm.
- None of the scheduling optimization criteria is applied in FCFS
- FCFS is not a good choice for interactive processes

Example

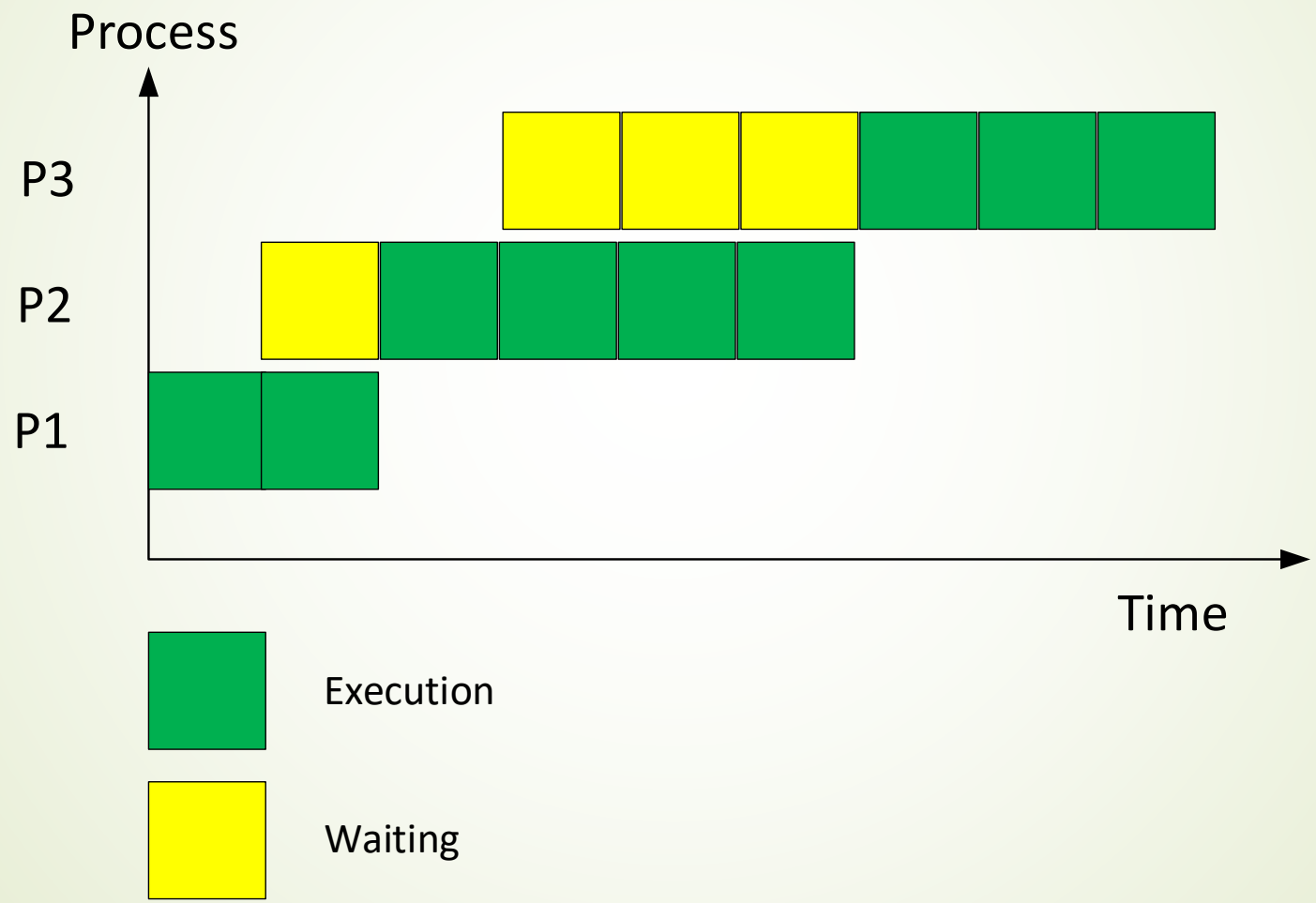
FIRST COME FIRST SERVE

Process	Time needed to execute	Arrival Time
P1	2	0
P2	4	1
P3	3	3

→ [0-2] → 2-0=2
→ [2-6] → 6-1=5
→ [6-9] → 9-3=6

- Average turnaround time = [(2 + 0) + (4 + 1) + (3 + 3)] /3
  = 13/3 = 4.33
- Average waiting time = (0 + 1 + 3)/3 = 4/3 = 1.33
- Throughput = 3 / 9

Gantt Chart Example



Shortest Job First (SJF)

- Shortest Job First scheduling algorithm chooses the next process to execute from the processes currently in the ready queue, based on their execution time
- Shortest Job First minimizes the average turnaround time.
- Shortest Job First is non-preemptive

Example

Process	Time needed to execute	Arrival Time
P1	3	0
P2	4	1
P3	2	2

3-3=0

→ [0-3] => 3-0=3

→ [5-9] => 9-1=8

→ [3-5] => 5-2=3 / 3

8-4=4

3-2=1

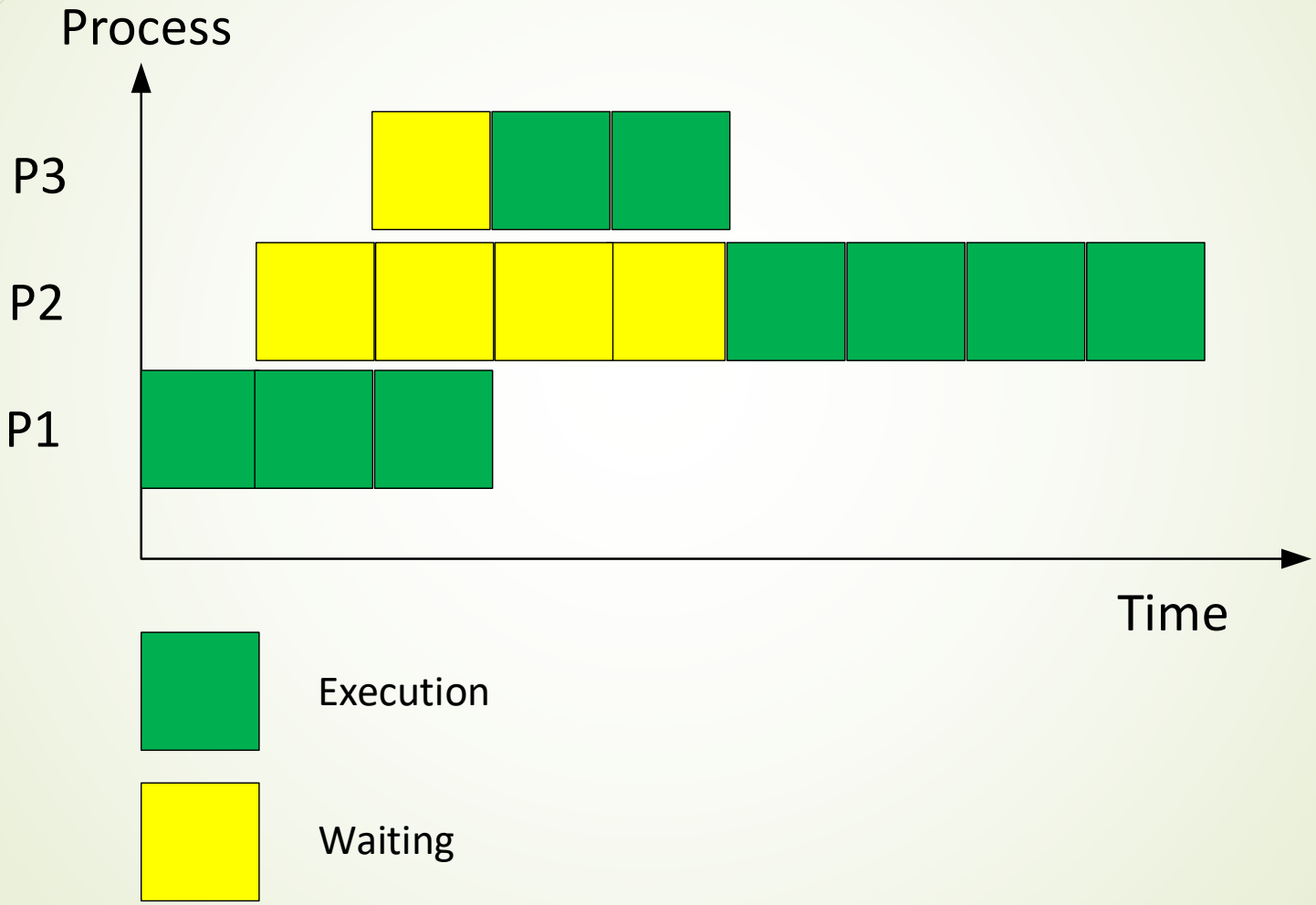
- Processes are execute in P1, P3, P2 order
- Average turnaround time = [(3 + 0) + (4 + 4) + (2 + 1)] /3

waiting time = turnaround time - execute time

= 14/3 = 4.67

- Average waiting time = (0 + 4 + 1)/3 = 5/3 = 1.67
- Throughput = 3 / 9

Gantt Chart Example



Example

- For the following processes, find the average turnaround time using Shortest Job First (SJF), and draw the Gantt chart.

Process	Time needed to execute	Arrival Time
P1	3	0
P2	4	1
P3	3	2
P4	3	4

→[0-3] → 3-0=3

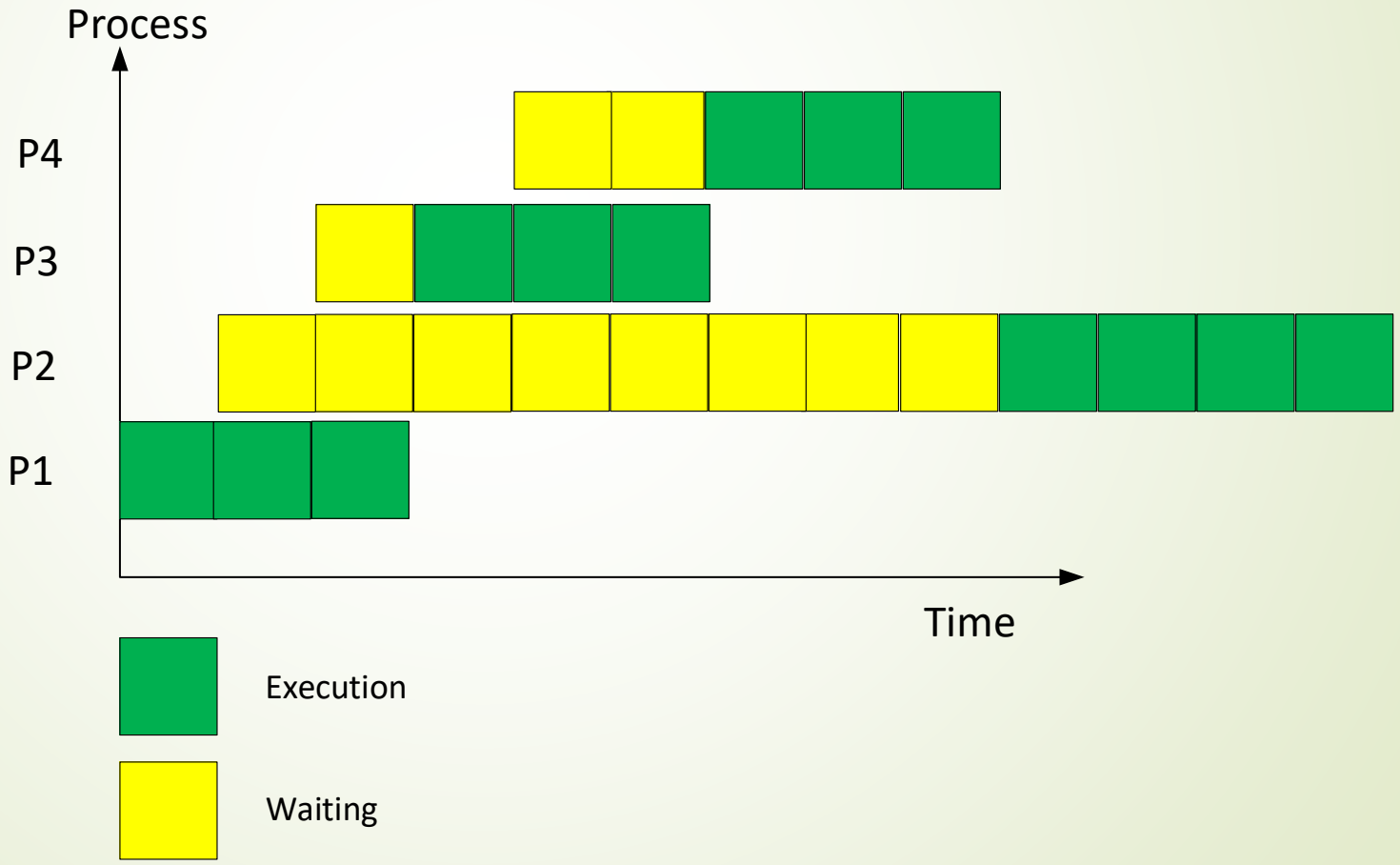
→[9-13] → 13-1=12

→[3-6] → 6-2=4

→[6-9] → 9-4=5

Solution

- Average turnaround time = [(3 + 0) + (4 + 8) + (3 + 1) + (3 + 2)]
  = 24 / 4 = 6



Starvation Problem

- Starvation happens when a process waits for a resource for a long time.
- In scheduling, if a process waits too long (or in the worst case indefinitely) before using CPU, it is said that starvation has happened.

Starvation with SJF Scheduling

- Consider the following situation in scheduling with shortest job first:
- P2 may not have a chance to execute if more short processes arrive!

Process	Execution Time	Arrival Time
P1	3	0
P2	14	1
P3	2	2
P4	4	3
P5	3	4
..	..	..

Solution with Aging

- The process that waits longer, receives higher priority to execute.
- Shortest Job First uses this priority: **1/Execution time**
- Shortest Job First with Aging uses this priority:

{ 1/Execution + 0.1x Waiting time }

The priority in Shortest Job First with Aging is dynamic (changes as the process waits)

Example

- In the following example P2 receives higher priority after waiting for execution.

Process	Execution Time	Arrival Time
P1	2	0
P2	10	1
P3	3	1

[0-2]
[5-15]
[2-5]

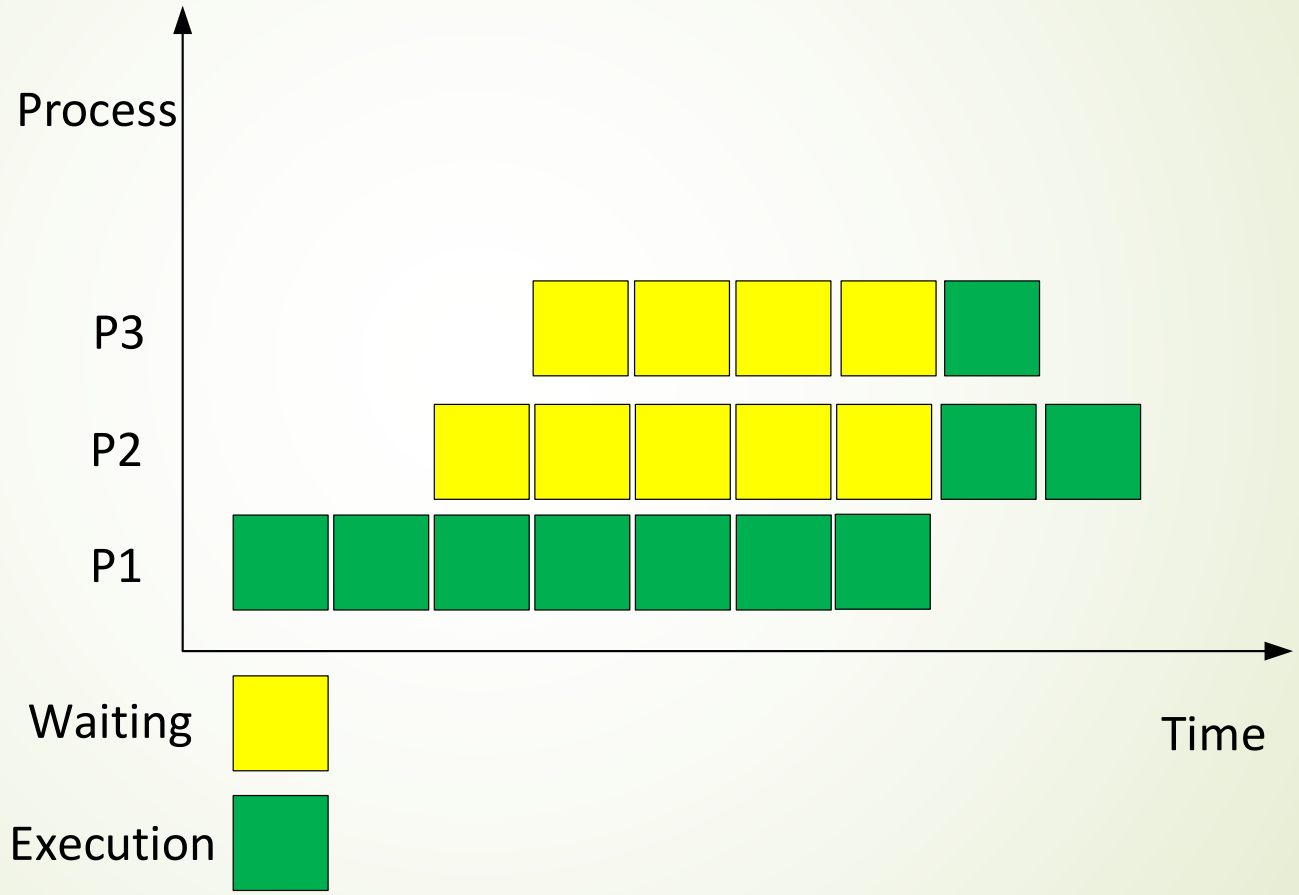
- P1 priority at time 0 = 1/2 + 0.1x0 = 0.5
- P2 priority at time 1 = 1/10 + 0.1x0 = 0.1 (P2 ve P3 gelir)
- P2 priority at time 6 = 1/10 + 0.1x5 = 0.6

Optimizing Turnaround Time

- Non-preemptive scheduling algorithms suffer from long average turnaround time.
- The reason is that when a long process starts, short processes have to wait until it terminates.

- **Problem:** The scheduler does not know if short processes will be submitted in future or not.

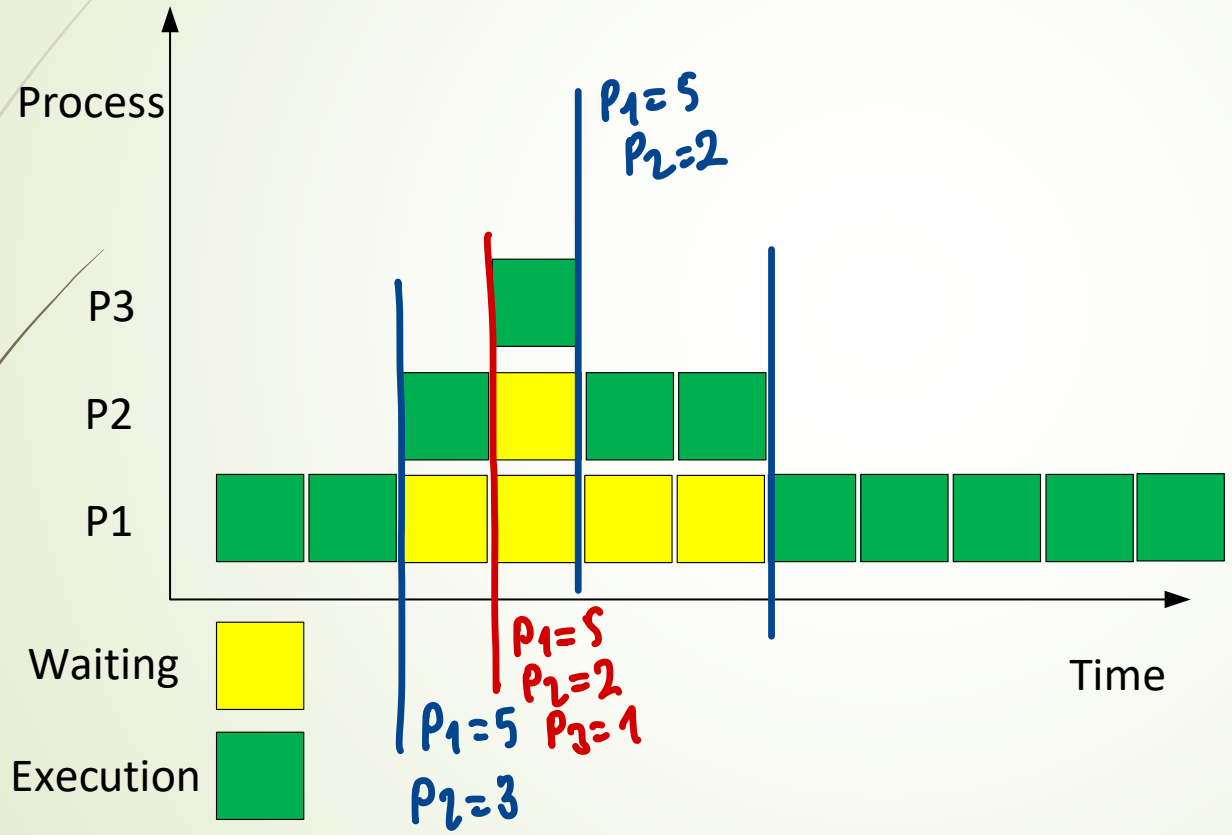
Example



Shortest Remaining Time First

- The preemptive version of shortest job first is, shortest remaining time first
- When a new process is submitted, the execution time of the new process and the remaining execution time of the current process are compared.
- The current process is swapped out if its remaining execution time is longer than the execution time of the new process

Example



Process	Execution Time	Arrival Time
P1	7	0
P2	3	2
P3	1	3

Round Robin

- Round Robin (RR) scheduling assign CPU to each process for one quantum.
- Round Robin is a preemptive scheduling.
- Round Robin does not consider priority for processes.

Example

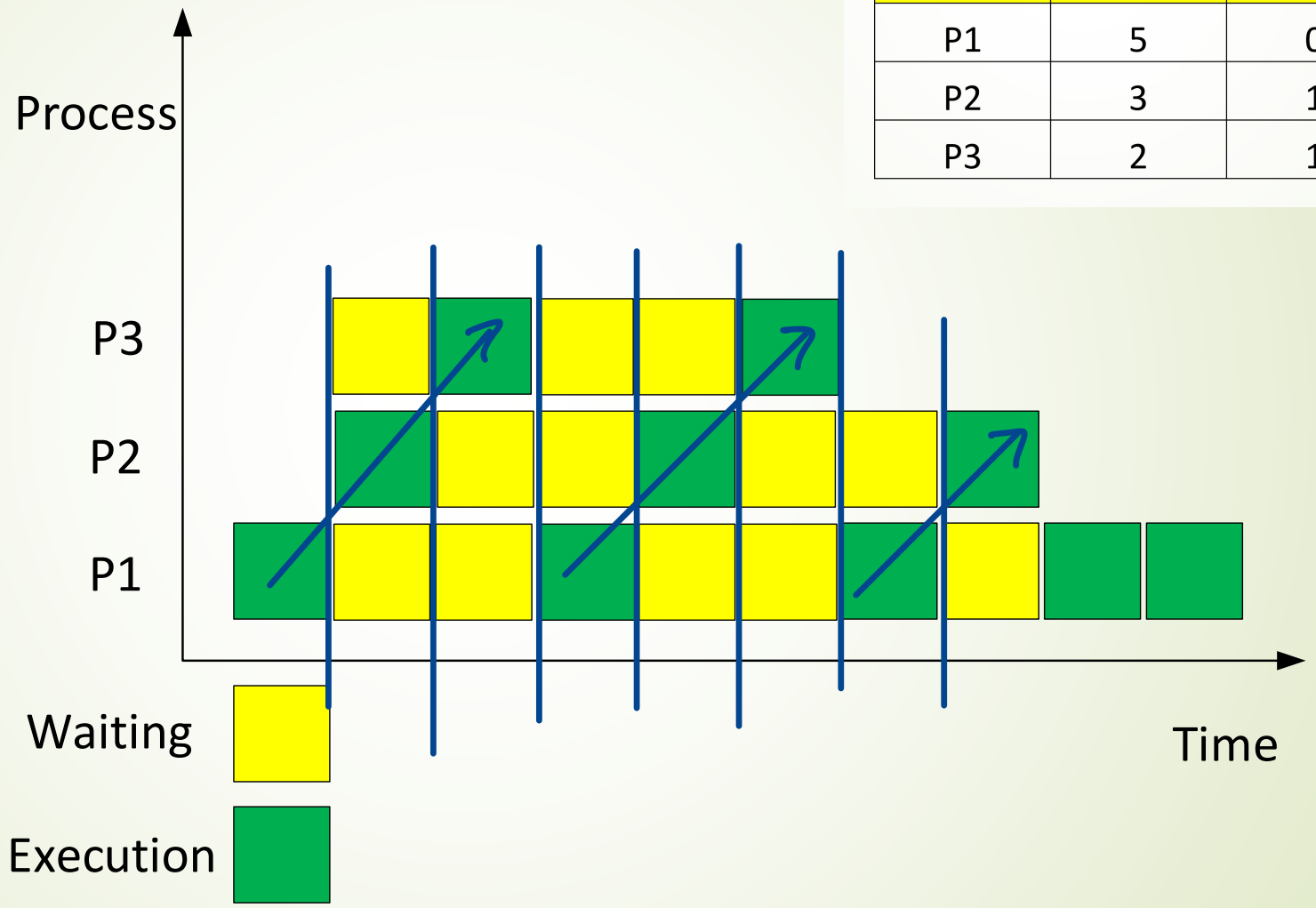
- Assume the following processes arrive at the ready queue as shown:

Process	Execution Time	Arrival Time
P1	5	0
P2	3	1
P3	2	1

- Show how the Round Robin schedules them.

Gantt Chart

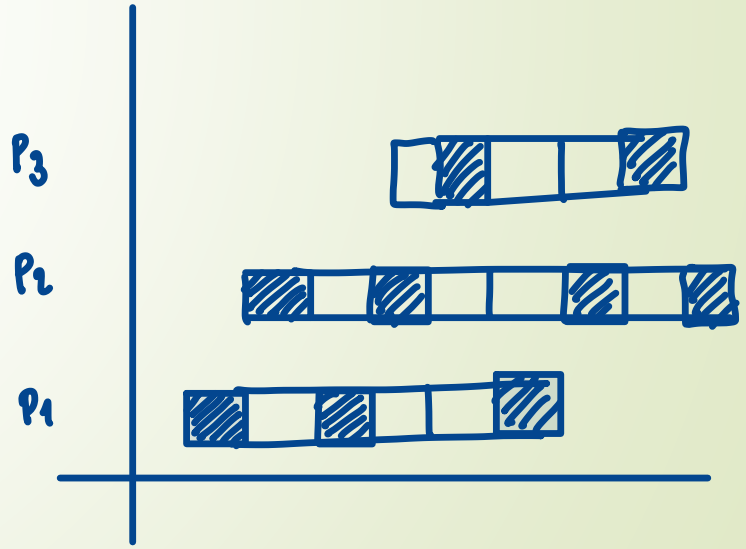
Process	Execution Time	Arrival Time
P1	5	0
P2	3	1
P3	2	1



Example

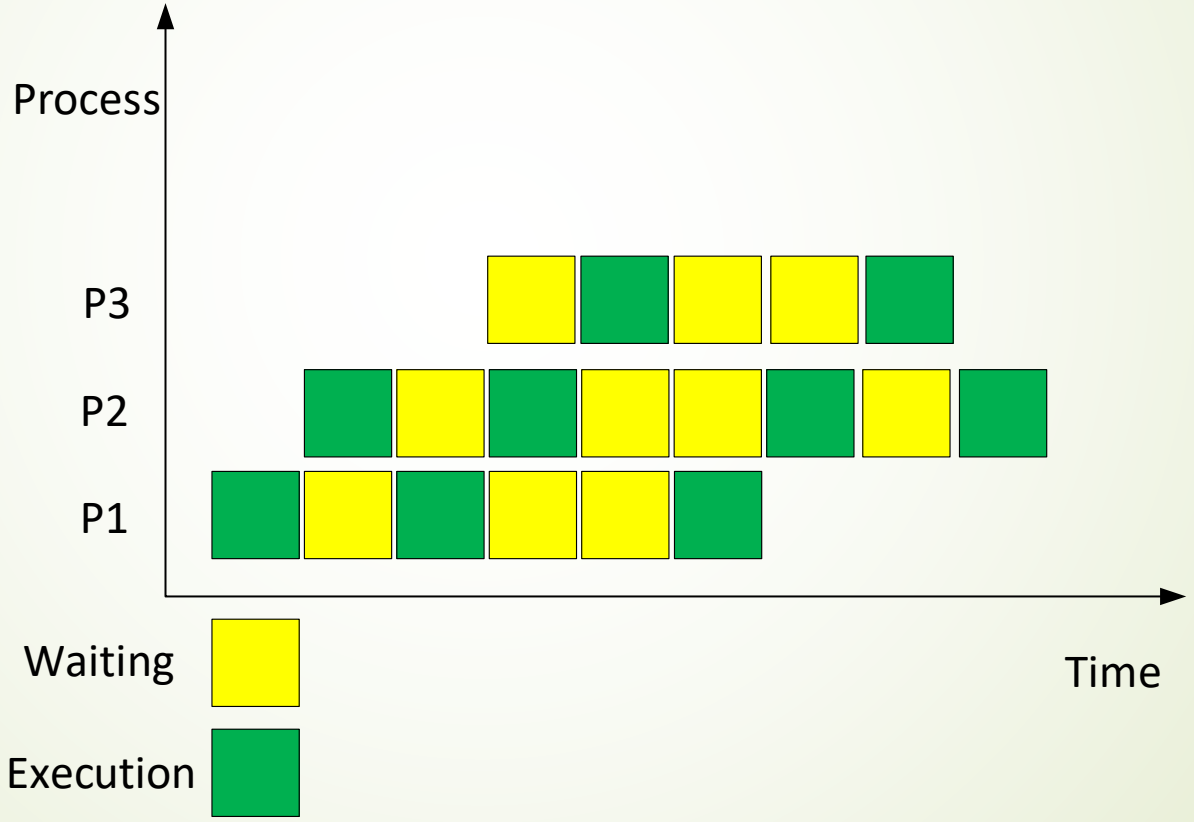
- Assume round robin scheduling is used with the following processes.
- Draw the Gantt chart and find the average turnaround time

Process	Execution Time	Arrival Time
P1	3	0
P2	4	1
P3	2	3



Solution

- Average turnaround time = [(3+3) + (4+4) + (2+3)] /3 =19/3=6.33



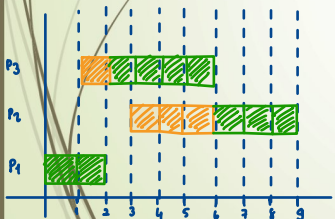
16

Q15

non-preemptive

Consider the following set of processes and assume the shortest job first scheduling algorithm is used. Which Gantt diagram represents the order of execution of these processes?

Choose shortest at that time



Process	Arrival time	Execution Time
P1	0	2
P2	3	3
P3	1	4

Average turnaround time = (2+6+5)/3 = 13/3

Average waiting time = (0+3+1)/3 = 4/3

[0-2] → 2-0 = 2, → 2-2=0

[6-9] → 9-3= 6, → 6-3=3

[2-6] → 6-1= 5, → 5-4=1

waiting time = Turnaround Time - execution Time

$t_{turnaround} = t_{waiting} + t_{execution}$

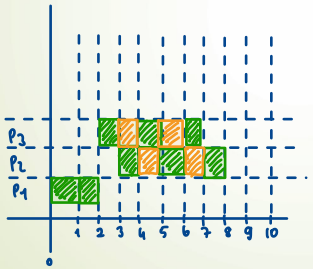
$P_1 = 0+2 = 2$
$P_2 = 2+3 = 5$
$P_3 = 2+3 = 5$
$2+5+5 = 12 / 3 = 4$

18

Q16

- Given the following processes, and assuming that round-robin algorithm is used, what will be the average turnaround time? (consider quantum=1)

a. 3
b. 4
c. 5
d. 6



Process	Arrival time	Execution Time
1) P1	0	2
3) P2	3	3
2) P3	2	3