Operating Systems

Chapter 5: CPU Scheduling

Dr. Ahmed Hagag

Scientific Computing Department,
Faculty of Computers and Artificial Intelligence
Benha University

2019

Chapter 5: CPU Scheduling

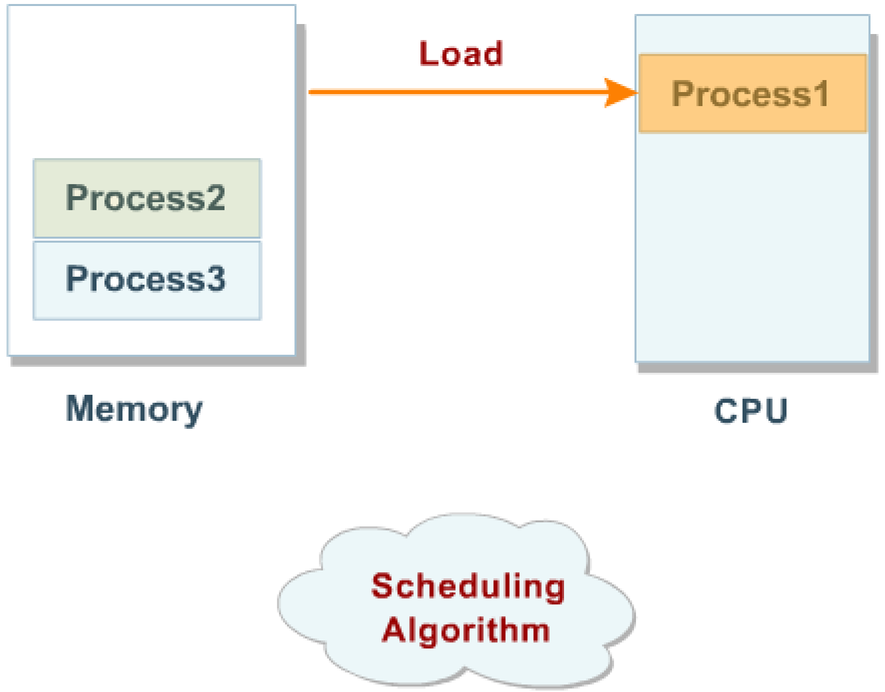
- Basic Concepts
- Scheduling Criteria
- Scheduling Algorithms

Basic Concepts (1/11)

- **CPU scheduling** is the central in multi-programming system.
- Maximum CPU utilization obtained with multiprogramming (**prevent CPU from being idle**).
- Processes residing in the main memory is selected by the Scheduler that is:
  - Concerned with deciding a policy about which process is to be selected.
  - Process selection based on a **scheduling algorithm**.

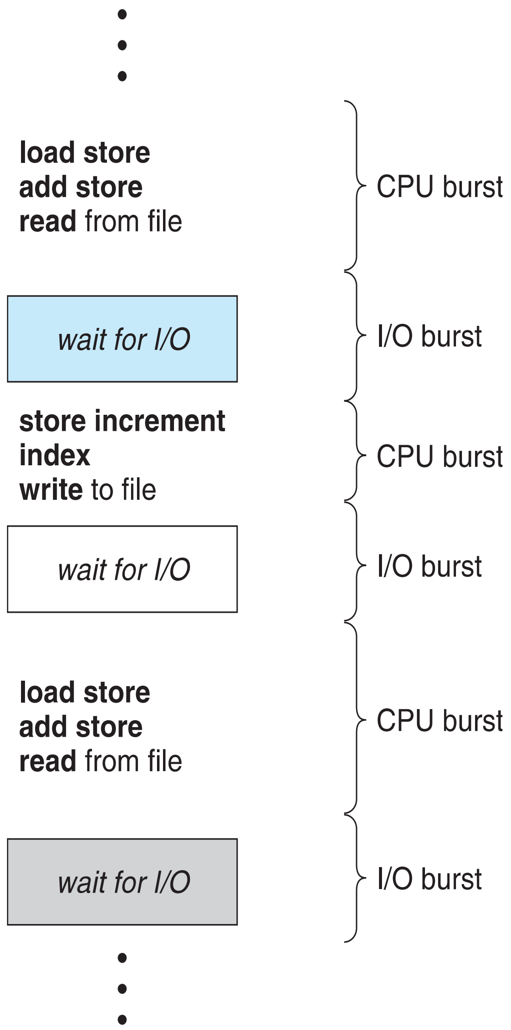
Basic Concepts (2/11)

Basic Concepts (2/11)



Basic Concepts (3/11)

- Process execution consists of a **cycle** of CPU execution and I/O wait.
- Processes alternate between these two states. Process execution begins with a **CPU burst**. That is followed by an **I/O burst**, which is followed by another CPU burst, then another I/O burst, and so on.
- CPU bursts vary greatly from process to process and from computer to computer.



Basic Concepts (4/11)

Schedulers

- **Long-term scheduler** chooses some of them to go to memory (ready queue).
- Then, **short-term scheduler** (or CPU scheduler) chooses from ready queue a job to run on CPU.
- **Medium-term scheduler** may move (swap) some partially-executed jobs from memory to disk (to enhance performance).

Basic Concepts (5/11)

CPU Scheduler

- Whenever the CPU becomes idle, the operating system must select one of the processes in the ready queue to be executed. The selection process is carried out by the **short-term scheduler**, or **CPU scheduler**.

Basic Concepts (6/11)

CPU scheduling decisions may take place when a process:

1. Switches from running to waiting state
2. Switches from running to ready state
3. Switches from waiting to ready
4. Terminates

Basic Concepts (7/11)

Scheduling can be

- **Non-preemptive**
  - Once a process is allocated the CPU, it does not leave until terminate.
- **Preemptive**
  - OS can force (preempt) a process from CPU at anytime.
    - ✓ Say, to allocate CPU to another higher-priority process.

Basic Concepts (8/11)

Non-preemptive and Preemptive

Which is harder to implement? and why?

Basic Concepts (9/11)

Non-preemptive and Preemptive

- **Preemptive is harder**: Need to maintain consistency of data shared between processes, and more importantly, kernel data structures (e.g., I/O queues).

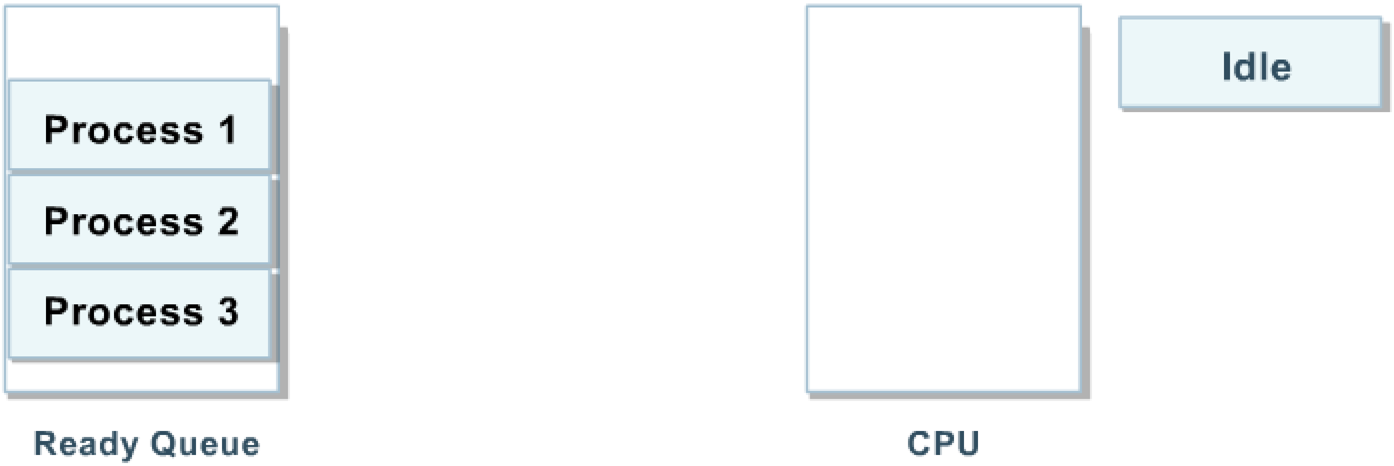
Basic Concepts (10/11)

Dispatcher

- Dispatcher module gives control of the CPU to the process selected by the short-term scheduler; this involves:
  - ➢ Switching context.
  - ➢ Switching to user mode.
  - ➢ Jumping to the proper location in the user program to restart that program.
- **Dispatch latency** – time it takes for the dispatcher to stop one process and start another running.

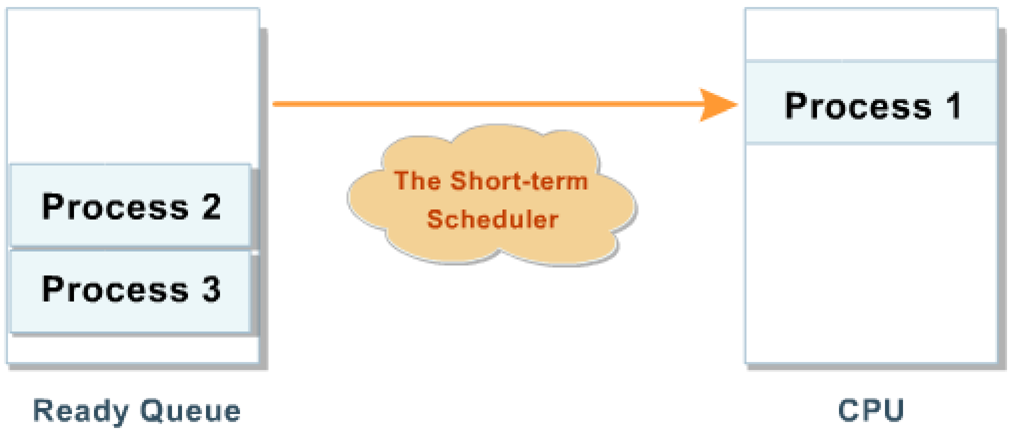
Basic Concepts (11/11)

Dispatcher



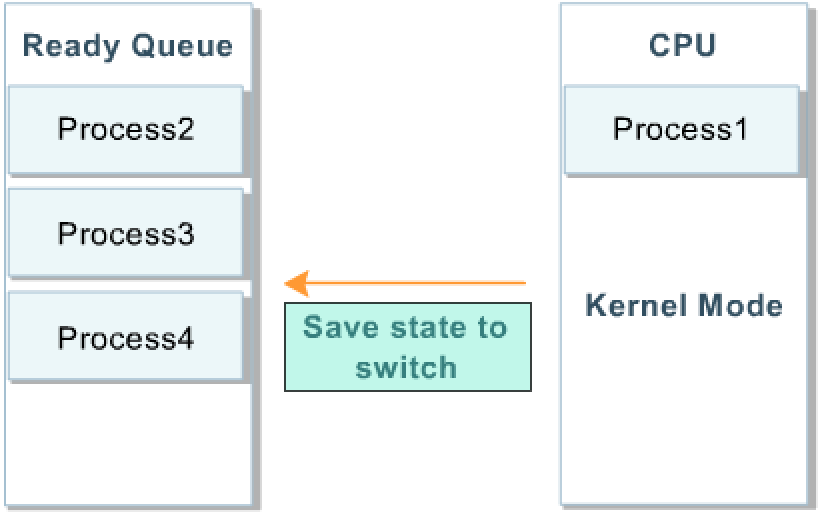
Basic Concepts (11/11)

Dispatcher



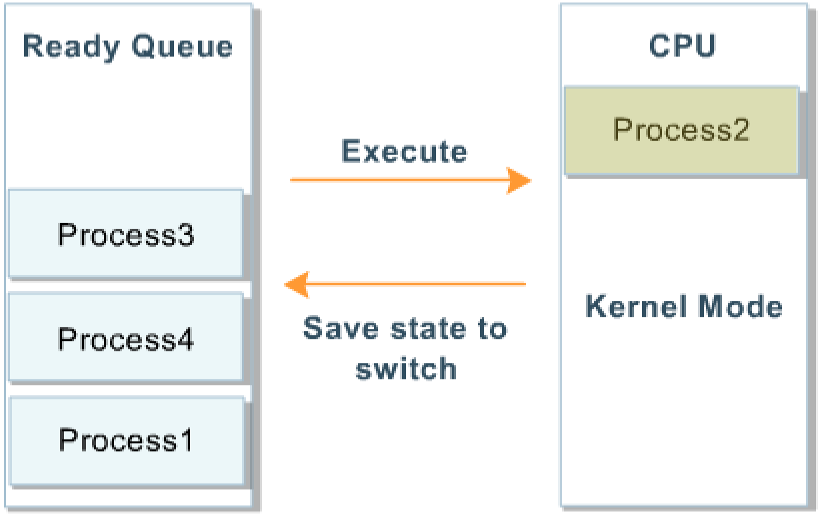
Basic Concepts (11/11)

Dispatcher



Basic Concepts (11/11)

Dispatcher



Scheduling Criteria (1/2)

- **CPU utilization** – keep the CPU as busy as possible.
- **Throughput** – #of processes that complete their execution per time unit.
- **Turnaround time** – amount of time to execute a particular process. (time from submission to termination)
- **Waiting time** – amount of time a process has been waiting in the ready queue.
- **Response time** – amount of time it takes from when a request was submitted until the first response is produced, not output.

Scheduling Criteria (2/2)

Scheduling Algorithm Optimization Criteria

- **Max** CPU utilization.
- **Max** throughput.
- **Min** turnaround time.
- **Min** waiting time.
- **Min** response time.

Scheduling Algorithms (1/30)

- There are many different CPU-scheduling algorithms:
  1. First Come, First Served (FCFS).
  2. Shortest Job First (SJF).
     - Preemptive SJF.
     - Non-Preemptive SJF.
  3. Priority.
  4. Round Robin.
  5. Multilevel queues.

1. First-Come, First-Served (FCFS) Scheduling

Process	Burst Time
P_1	24
P_2	3
P_3	3

- Suppose that the processes arrive in the order: P_1 , P_2 , P_3
  The **Gantt Chart** for the schedule is:

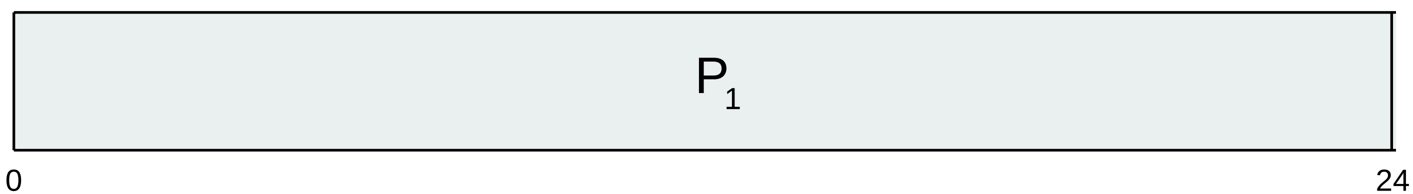
- **Note:** A process may have many CPU bursts, but in the following examples we show only one for simplicity.

Scheduling Algorithms (3/30)

1. First-Come, First-Served (FCFS) Scheduling

Process	Burst Time
P_1	24
P_2	3
P_3	3

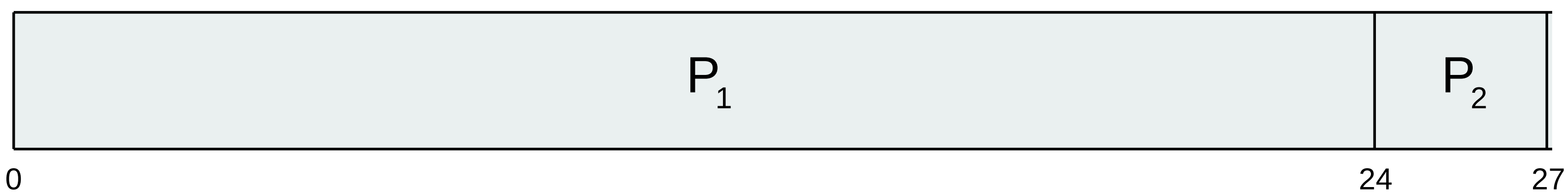
- Suppose that the processes arrive in the order: P_1 , P_2 , P_3
  The **Gantt Chart** for the schedule is:



1. First-Come, First-Served (FCFS) Scheduling

Process	Burst Time
P_1	24
P_2	3
P_3	3

- Suppose that the processes arrive in the order: P_1 , P_2 , P_3
  The **Gantt Chart** for the schedule is:



1. First-Come, First-Served (FCFS) Scheduling

Process	Burst Time
P_1	24
P_2	3
P_3	3

- Suppose that the processes arrive in the order: P_1 , P_2 , P_3
  The **Gantt Chart** for the schedule is:

Scheduling Algorithms (4/30)

1. First-Come, First-Served (FCFS) Scheduling

Process	Burst Time
P_1	24
P_2	3
P_3	3

- Suppose that the processes arrive in the order: P_1 , P_2 , P_3
  The **Gantt Chart** for the schedule is:

Waiting Time		
P_1	P_2	P_3

Scheduling Algorithms (4/30)

1. First-Come, First-Served (FCFS) Scheduling

Process	Burst Time
P_1	24
P_2	3
P_3	3

- Suppose that the processes arrive in the order: P_1 , P_2 , P_3
  The **Gantt Chart** for the schedule is:

Waiting Time		
P_1	P_2	P_3
0	24	27

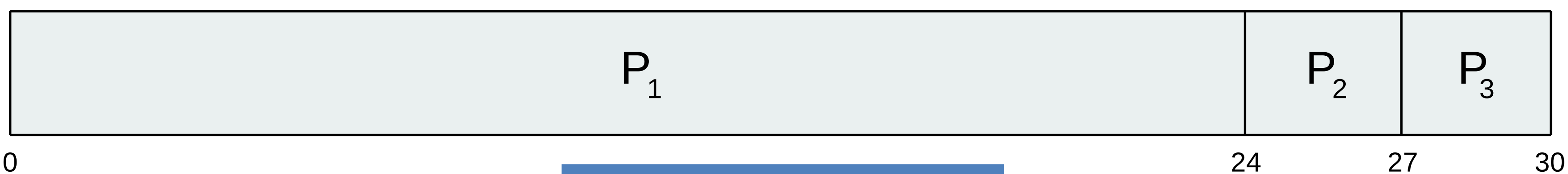
Average waiting time: (0 + 24 + 27)/3 = 17

Scheduling Algorithms (5/30)

1. First-Come, First-Served (FCFS) Scheduling

Process	Burst Time
P_1	24
P_2	3
P_3	3

- Suppose that the processes arrive in the order: P_1 , P_2 , P_3
  The **Gantt Chart** for the schedule is:



Turnaround Time		
P_1	P_2	P_3

1. First-Come, First-Served (FCFS) Scheduling

Process	Burst Time
P_1	24
P_2	3
P_3	3

- Suppose that the processes arrive in the order: P_1 , P_2 , P_3
  The **Gantt Chart** for the schedule is:

P_1	P_2	P_3
0 – 24	24 – 27	27 – 30

0 24 27 30

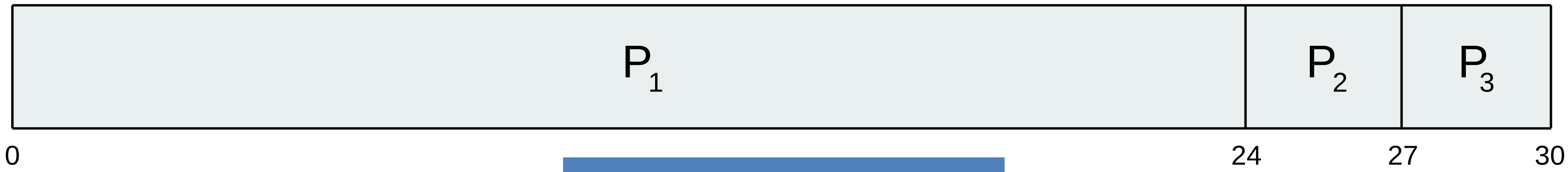
Turnaround Time		
P_1	P_2	P_3
24	27	30

Average turnaround time: (24 + 27 + 30)/3 = 27

1. First-Come, First-Served (FCFS) Scheduling

Process	Burst Time
P_1	24
P_2	3
P_3	3

- Suppose that the processes arrive in the order: P_1 , P_2 , P_3
  The **Gantt Chart** for the schedule is:

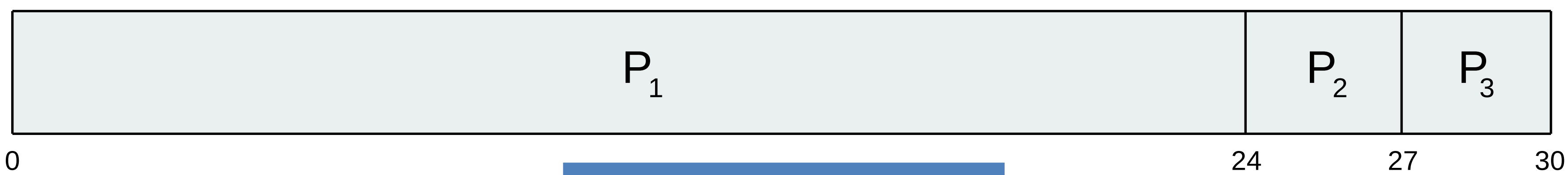


Response Time		
P_1	P_2	P_3

1. First-Come, First-Served (FCFS) Scheduling

Process	Burst Time
P_1	24
P_2	3
P_3	3

- Suppose that the processes arrive in the order: P_1 , P_2 , P_3
  The **Gantt Chart** for the schedule is:



Response Time		
P_1	P_2	P_3
0	24	27

Average response time: (0 + 24 + 27)/3 = 17

1. First-Come, First-Served (FCFS) Scheduling

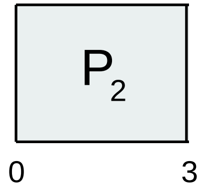
Process	Burst Time
P_1	24
P_2	3
P_3	3

- Suppose that the processes arrive in the **order**: P_2 , P_3 , P_1
  The **Gantt Chart** for the schedule is:

1. First-Come, First-Served (FCFS) Scheduling

Process	Burst Time
P_1	24
P_2	3
P_3	3

- Suppose that the processes arrive in the **order**: P_2 , P_3 , P_1
  The **Gantt Chart** for the schedule is:



1. First-Come, First-Served (FCFS) Scheduling

Process	Burst Time
P_1	24
P_2	3
P_3	3

- Suppose that the processes arrive in the **order**: P_2 , P_3 , P_1
  The **Gantt Chart** for the schedule is:

P_2	P_3
0 – 3	3 – 6

1. First-Come, First-Served (FCFS) Scheduling

Process	Burst Time
P_1	24
P_2	3
P_3	3

- Suppose that the processes arrive in the **order**: $\boldsymbol{P_2 , P_3 , P_1}$
  The **Gantt Chart** for the schedule is:

1. First-Come, First-Served (FCFS) Scheduling

Process	Burst Time
P_1	24
P_2	3
P_3	3

- Suppose that the processes arrive in the **order**: $\boldsymbol{P_2 , P_3 , P_1}$
  The **Gantt Chart** for the schedule is:

Waiting Time		
$\boldsymbol{P_1}$	$\boldsymbol{P_2}$	$\boldsymbol{P_3}$

1. First-Come, First-Served (FCFS) Scheduling

Process	Burst Time
P_1	24
P_2	3
P_3	3

- Suppose that the processes arrive in the **order**: P_2 , P_3 , P_1
  The **Gantt Chart** for the schedule is:

Waiting Time		
P_1	P_2	P_3
6	0	3

Average waiting time: (6 + 0 + 3)/3 = 3

1. First-Come, First-Served (FCFS) Scheduling

Process	Burst Time
P_1	24
P_2	3
P_3	3

- Suppose that the processes arrive in the **order**: P_2 , P_3 , P_1
  The **Gantt Chart** for the schedule is:

Turnaround Time		
P_1	P_2	P_3

1. First-Come, First-Served (FCFS) Scheduling

Process	Burst Time
P_1	24
P_2	3
P_3	3

- Suppose that the processes arrive in the **order**: P_2 , P_3 , P_1
  The **Gantt Chart** for the schedule is:

Turnaround Time		
P_1	P_2	P_3
30	3	6

Average turnaround time: (30 + 3 + 6)/3 = 13

1. First-Come, First-Served (FCFS) Scheduling

Process	Burst Time
P_1	24
P_2	3
P_3	3

- Suppose that the processes arrive in the **order**: $\boldsymbol{P_2, P_3, P_1}$
  The **Gantt Chart** for the schedule is:

Response Time		
$\boldsymbol{P_1}$	$\boldsymbol{P_2}$	$\boldsymbol{P_3}$

1. First-Come, First-Served (FCFS) Scheduling

Process	Burst Time
P_1	24
P_2	3
P_3	3

- Suppose that the processes arrive in the **order**: $\boldsymbol{P_2 , P_3 , P_1}$
  The **Gantt Chart** for the schedule is:

Response Time		
P_1	P_2	P_3
6	0	3

Average response time: (6 + 0 + 3)/3 = 3

1. First-Come, First-Served (FCFS) Scheduling

- FCFS is fair in the formal sense or human sense of fairness.
- but it is unfair in the sense that long jobs take priority over short jobs and unimportant jobs make important jobs wait.
- One of the major drawbacks of this scheme is that the waiting time and the average turnaround time is often quite long.

2. Shortest-Job-First (SJF) Scheduling

- Associate with each process the length of its next CPU burst.
  - Use these lengths to schedule the process with the shortest time.
- SJF is optimal – gives minimum average waiting time for a given set of processes.
  - The difficulty is knowing the length of the next CPU request.
  - Could ask the user.

2.1 Shortest-Job-First (SJF) Scheduling

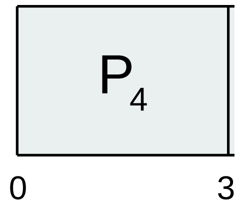
Process	Burst Time
P_1	6
P_2	8
P_3	7
P_4	3

- SJF scheduling chart

2.1 Shortest-Job-First (SJF) Scheduling

Process	Burst Time
P_1	6
P_2	8
P_3	7
P_4	3

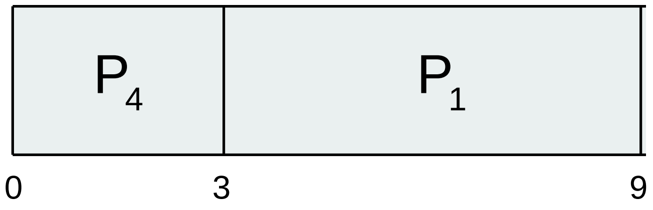
- SJF scheduling chart



2.1 Shortest-Job-First (SJF) Scheduling

Process	Burst Time
P_1	6
P_2	8
P_3	7
P_4	3

- SJF scheduling chart



2.1 Shortest-Job-First (SJF) Scheduling

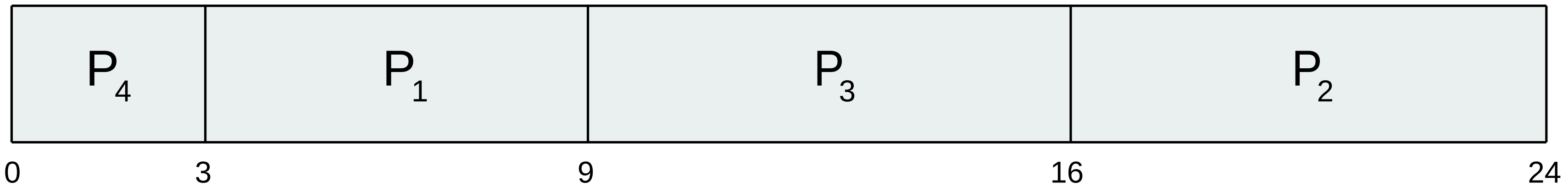
Process	Burst Time
P_1	6
P_2	8
P_3	7
P_4	3

- SJF scheduling chart

2.1 Shortest-Job-First (SJF) Scheduling

Process	Burst Time
P_1	6
P_2	8
P_3	7
P_4	3

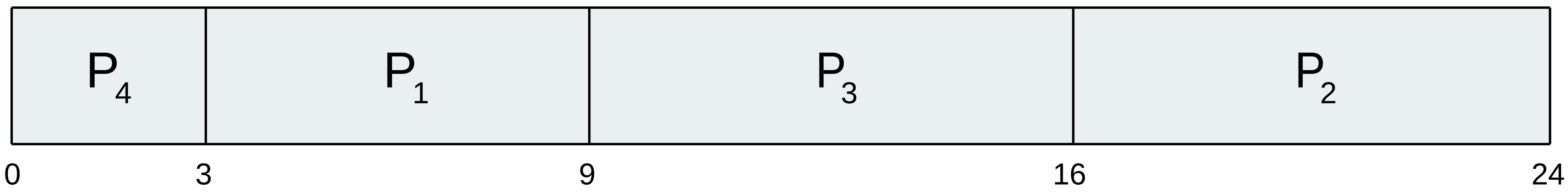
- SJF scheduling chart



2.1 Shortest-Job-First (SJF) Scheduling

Process	Burst Time
P_1	6
P_2	8
P_3	7
P_4	3

- SJF scheduling chart

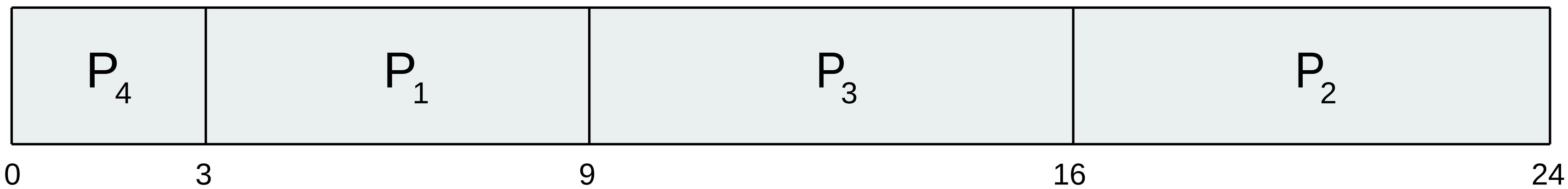


Waiting Time			
P_1	P_2	P_3	P_4

2.1 Shortest-Job-First (SJF) Scheduling

Process	Burst Time
P_1	6
P_2	8
P_3	7
P_4	3

- SJF scheduling chart



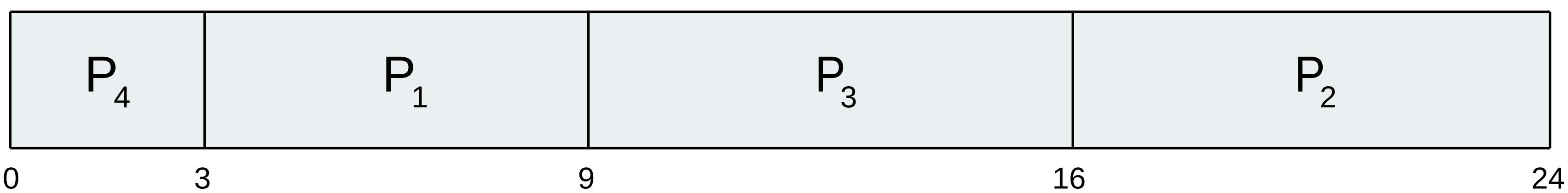
Waiting Time			
P_1	P_2	P_3	P_4
3	16	9	0

Average waiting time: (3 + 16 + 9 + 0)/4 = 7

2.1 Shortest-Job-First (SJF) Scheduling

Process	Burst Time
P_1	6
P_2	8
P_3	7
P_4	3

- SJF scheduling chart

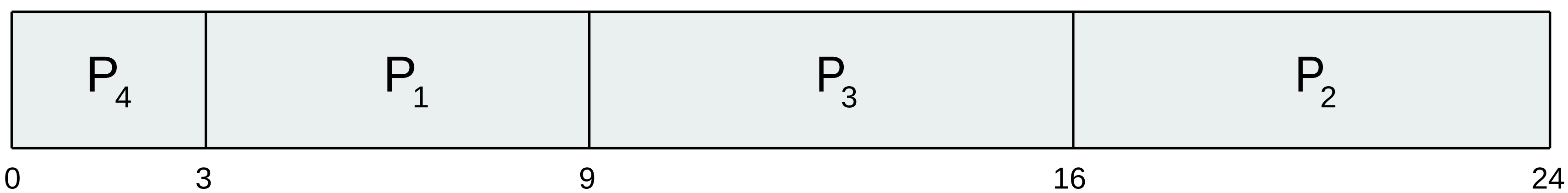


Turnaround Time			
P_1	P_2	P_3	P_4

2.1 Shortest-Job-First (SJF) Scheduling

Process	Burst Time
P_1	6
P_2	8
P_3	7
P_4	3

- SJF scheduling chart



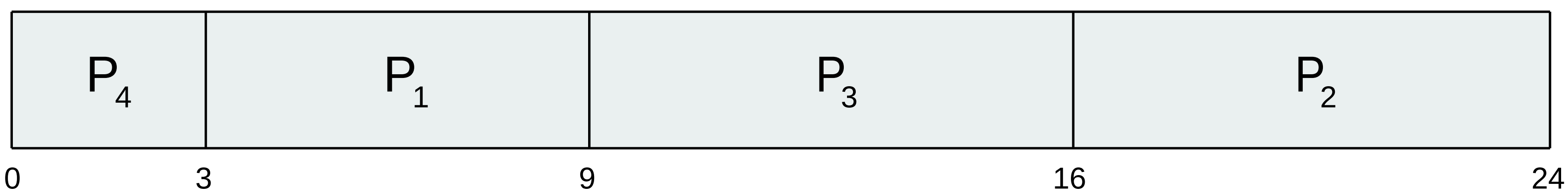
Turnaround Time			
P_1	P_2	P_3	P_4
9	24	16	3

Average Turnaround time: (9 + 24 + 16 + 3)/4 = 13

2.1 Shortest-Job-First (SJF) Scheduling

Process	Burst Time
P_1	6
P_2	8
P_3	7
P_4	3

- SJF scheduling chart

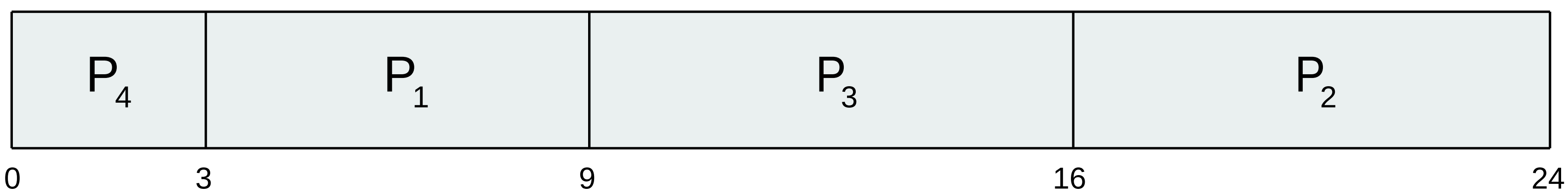


Response Time			
P_1	P_2	P_3	P_4

2.1 Shortest-Job-First (SJF) Scheduling

Process	Burst Time
P_1	6
P_2	8
P_3	7
P_4	3

- SJF scheduling chart



Response Time			
P_1	P_2	P_3	P_4
3	16	9	0

Average Response time: (3 + 16 + 9 + 0)/4 = 7

2.1 Shortest-Job-First (SJF) (Non-Preemptive SJF)

- Now we add the concepts of varying arrival times and preemption to the analysis

Process	***Arrival* Time**	Burst Time
P_1	0	1
P_2	1	4
P_3	2	7
P_4	3	5

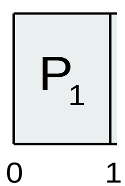
- *Non-Preemptive* SJF Gantt Chart

2.1 Shortest-Job-First (SJF) (Non-Preemptive SJF)

- Now we add the concepts of varying arrival times and preemption to the analysis

Process	*Arrival* Time	Burst Time
P_1	0	1
P_2	1	4
P_3	2	7
P_4	3	5

- *Non-Preemptive* SJF Gantt Chart

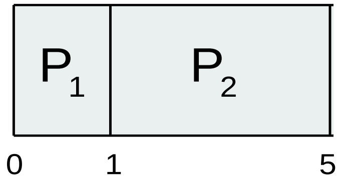


2.1 Shortest-Job-First (SJF) (Non-Preemptive SJF)

- Now we add the concepts of varying arrival times and preemption to the analysis

Process	*Arrival* Time	Burst Time
P_1	0	1
P_2	1	4
P_3	2	7
P_4	3	5

- *Non-Preemptive* SJF Gantt Chart

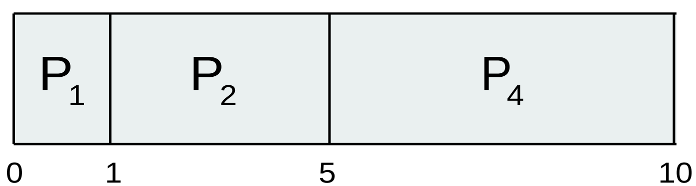


2.1 Shortest-Job-First (SJF) (Non-Preemptive SJF)

- Now we add the concepts of varying arrival times and preemption to the analysis

Process	*Arrival* Time	Burst Time
P_1	0	1
P_2	1	4
P_3	2	7
P_4	3	5

- *Non-Preemptive* SJF Gantt Chart



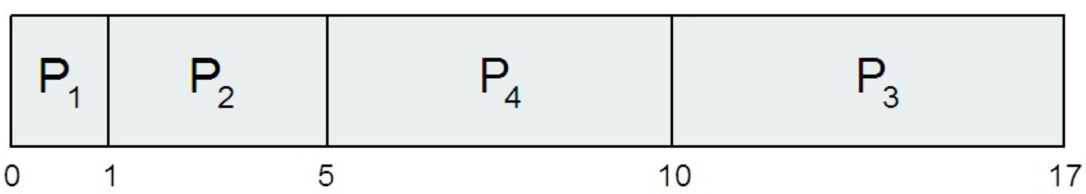
Scheduling Algorithms (20/30)

2.1 Shortest-Job-First (SJF) (Non-Preemptive SJF)

- Now we add the concepts of varying arrival times and preemption to the analysis

Process	*Arrival* Time	Burst Time
P_1	0	1
P_2	1	4
P_3	2	7
P_4	3	5

- *Non-Preemptive* SJF Gantt Chart

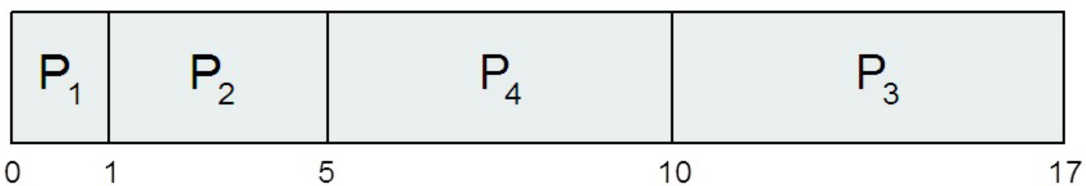


2.1 Shortest-Job-First (SJF) (Non-Preemptive SJF)

- Now we add the concepts of varying arrival times and preemption to the analysis

Process	*Arrival* Time	Burst Time
P_1	0	1
P_2	1	4
P_3	2	7
P_4	3	5

- *Non-Preemptive* SJF Gantt Chart



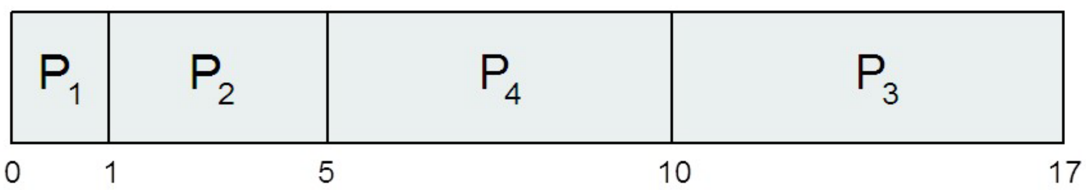
Waiting Time			
P_1	P_2	P_3	P_4

2.1 Shortest-Job-First (SJF) (Non-Preemptive SJF)

- Now we add the concepts of varying arrival times and preemption to the analysis

Process	*Arrival* Time	Burst Time
P_1	0	1
P_2	1	4
P_3	2	7
P_4	3	5

- *Non-Preemptive* SJF Gantt Chart



Waiting Time			
P_1	P_2	P_3	P_4
$(0-0)$	$(1-1)$	$(10-2)$	$(5-3)$

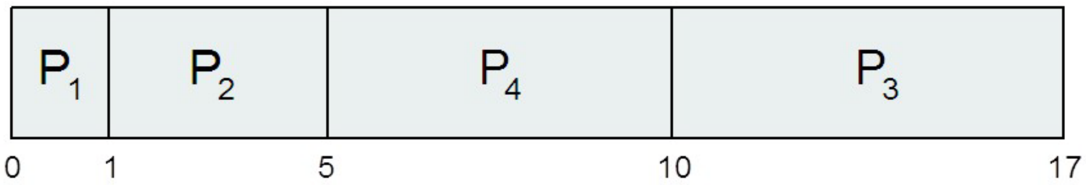
Average waiting time: $(0 + 0 + 8 + 2)/4 = 2.5$ msec

2.1 Shortest-Job-First (SJF) (Non-Preemptive SJF)

- Now we add the concepts of varying arrival times and preemption to the analysis

Process	*Arrival* Time	Burst Time
P_1	0	1
P_2	1	4
P_3	2	7
P_4	3	5

- *Non-Preemptive* SJF Gantt Chart



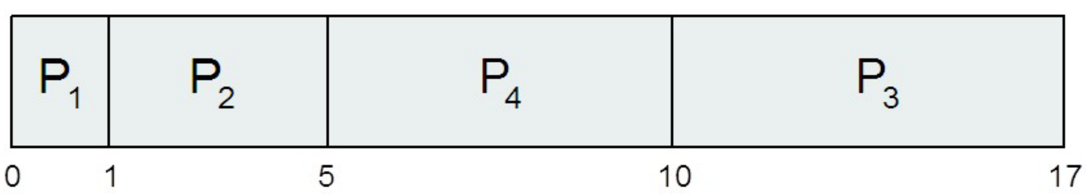
Turnaround Time			
P_1	P_2	P_3	P_4

2.1 Shortest-Job-First (SJF) (Non-Preemptive SJF)

- Now we add the concepts of varying arrival times and preemption to the analysis

Process	*Arrival* Time	Burst Time
P_1	0	1
P_2	1	4
P_3	2	7
P_4	3	5

- *Non-Preemptive* SJF Gantt Chart



Turnaround Time			
P_1	P_2	P_3	P_4
$(1-0)$	$(5-1)$	$(17-2)$	$(10-3)$

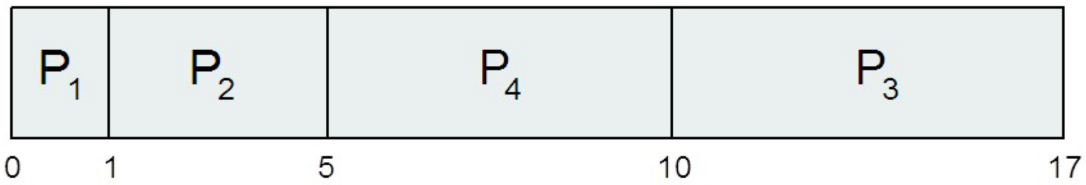
Average Turnaround time: (1 + 4 + 15 + 7)/4 = 6.75 msec

2.1 Shortest-Job-First (SJF) (Non-Preemptive SJF)

- Now we add the concepts of varying arrival times and preemption to the analysis

Process	*Arrival* Time	Burst Time
P_1	0	1
P_2	1	4
P_3	2	7
P_4	3	5

- *Non-Preemptive* SJF Gantt Chart



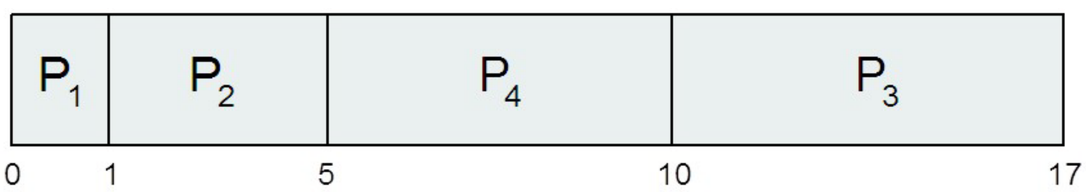
Response Time			
P_1	P_2	P_3	P_4

2.1 Shortest-Job-First (SJF) (Non-Preemptive SJF)

- Now we add the concepts of varying arrival times and preemption to the analysis

Process	*Arrival* Time	Burst Time
P_1	0	1
P_2	1	4
P_3	2	7
P_4	3	5

- *Non-Preemptive* SJF Gantt Chart



Response Time			
P_1	P_2	P_3	P_4
$(0-0)$	$(1-1)$	$(10-2)$	$(5-3)$

Average Response time: $(0 + 0 + 8 + 2)/4 = 2.5$ msec

Determining Length of Next CPU Burst

- Can only estimate the length – should be similar to the previous one
  - Then pick process with shortest predicted next CPU burst

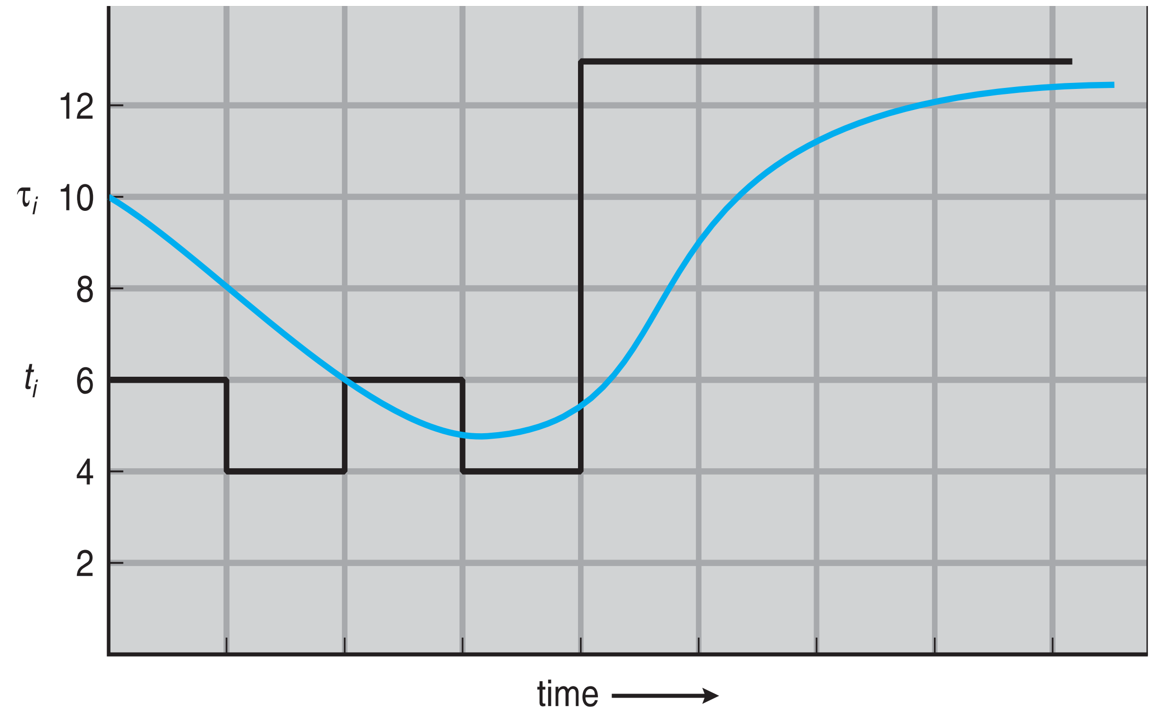
- Can be done by using the length of previous CPU bursts, using exponential averaging

$$
\begin{aligned}
&1.\ t_n = \text{actual length of } n^{th} \text{ CPU burst} \\
&2.\ \tau_{n+1} = \text{predicted value for the next CPU burst} \\
&3.\ \alpha, 0 \le \alpha \le 1 \\
&4.\ \text{Define: } \tau_{n=1} = \alpha\, t_n + (1-\alpha)\tau_n .
\end{aligned}
$$

- Commonly, α set to ½
- Preemptive version called **shortest-remaining-time-first**

Scheduling Algorithms (25/30)

Prediction of the Length of the Next CPU Burst



CPU burst (t_i)		6	4	6	4	13	13	13	...
"guess" (τ_i)	10	8	6	6	5	9	11	12	...

2.2 Shortest-remaining-time-first (Preemptive SJF)

- Now we add the concepts of varying arrival times and preemption to the analysis

Process	***Arrival* Time**	Burst Time
P_1	0	8
P_2	1	4
P_3	2	9
P_4	3	5

- ***Preemptive* SJF** Gantt Chart

2.2 Shortest-remaining-time-first (Preemptive SJF)

- Now we add the concepts of varying arrival times and preemption to the analysis

Process	***Arrival* Time**	Burst Time
P_1	**0**	8
P_2	1	4
P_3	2	9
P_4	3	5

0 ms

- ***Preemptive*** SJF Gantt Chart

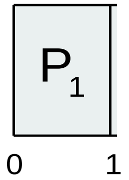
2.2 Shortest-remaining-time-first (Preemptive SJF)

- Now we add the concepts of varying arrival times and preemption to the analysis

Process	***Arrival* Time**	Burst Time
P_1	**0**	8
P_2	1	4
P_3	2	9
P_4	3	5

0 ms

- ***Preemptive*** SJF Gantt Chart



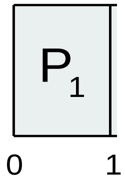
2.2 Shortest-remaining-time-first (Preemptive SJF)

- Now we add the concepts of varying arrival times and preemption to the analysis

Process	***Arrival* Time**	Burst Time
P_1	0	~~8~~ 7
P_2	**1**	4
P_3	2	9
P_4	3	5

1 ms

- ***Preemptive*** SJF Gantt Chart



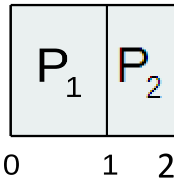
2.2 Shortest-remaining-time-first (Preemptive SJF)

- Now we add the concepts of varying arrival times and preemption to the analysis

Process	***Arrival* Time**	Burst Time
P_1	0	~~8~~ 7
P_2	1	~~4~~ 3
P_3	**2**	9
P_4	3	5

2 ms

- ***Preemptive*** SJF Gantt Chart



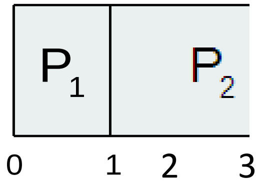
2.2 Shortest-remaining-time-first (Preemptive SJF)

- Now we add the concepts of varying arrival times and preemption to the analysis

Process	***Arrival* Time**	Burst Time
P_1	0	~~8~~ 7
P_2	1	~~4~~ ~~3~~ 2
P_3	2	9
P_4	**3**	5

3 ms

- ***Preemptive*** SJF Gantt Chart



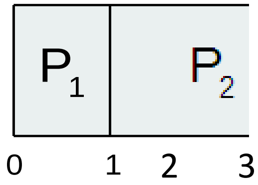
2.2 Shortest-remaining-time-first (Preemptive SJF)

- Now we add the concepts of varying arrival times and preemption to the analysis

Process	***Arrival* Time**	Burst Time
P_1	0	~~8~~ 7
P_2	1	~~4~~ ~~3~~ 2
P_3	2	9
P_4	**3**	5

3 ms

- ***Preemptive*** SJF Gantt Chart



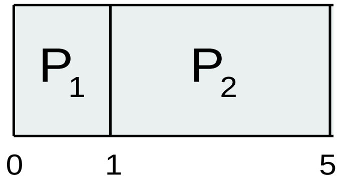
2.2 Shortest-remaining-time-first (Preemptive SJF)

- Now we add the concepts of varying arrival times and preemption to the analysis

Process	***Arrival*** **Time**	Burst Time
P_1	0	~~8~~ 7
P_2	1	~~4~~ ~~3~~ ~~2~~
P_3	2	9
P_4	3	5

5 ms

- ***Preemptive*** SJF Gantt Chart



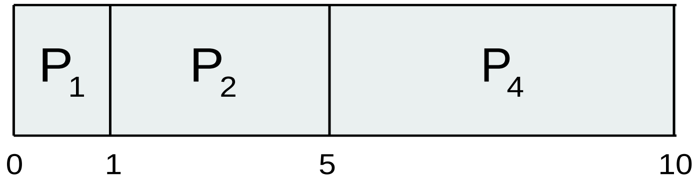
2.2 Shortest-remaining-time-first (Preemptive SJF)

- Now we add the concepts of varying arrival times and preemption to the analysis

Process	***Arrival* Time**	Burst Time
P_1	0	~~8~~ 7
P_2	1	~~4~~ ~~3~~ ~~2~~
P_3	2	9
P_4	3	~~5~~

10 ms

- ***Preemptive*** SJF Gantt Chart



Scheduling Algorithms (27/30)

2.2 Shortest-remaining-time-first (Preemptive SJF)

- Now we add the concepts of varying arrival times and preemption to the analysis

Process	***Arrival* Time**	Burst Time
P_1	0	~~8~~ ~~7~~
P_2	1	~~4~~ ~~3~~ ~~2~~
P_3	2	9
P_4	3	~~5~~

17 ms

- ***Preemptive*** SJF Gantt Chart

P_1	P_2	P_4	P_1

0 1 5 10 17

2.2 Shortest-remaining-time-first (Preemptive SJF)

- Now we add the concepts of varying arrival times and preemption to the analysis

Process	***Arrival*** **Time**	Burst Time
P_1	0	~~8~~ ~~7~~
P_2	1	~~4~~ ~~3~~ ~~2~~
P_3	2	~~9~~
P_4	3	~~5~~

26 ms

- ***Preemptive*** SJF Gantt Chart

P_1	P_2	P_4	P_1	P_3
0–1	1–5	5–10	10–17	17–26

0 1 5 10 17 26

2.2 Shortest-remaining-time-first (Preemptive SJF)

Process	*Arrival* Time	Burst Time
P_1	0	8
P_2	1	4
P_3	2	9
P_4	3	5

- *Preemptive* SJF Gantt Chart

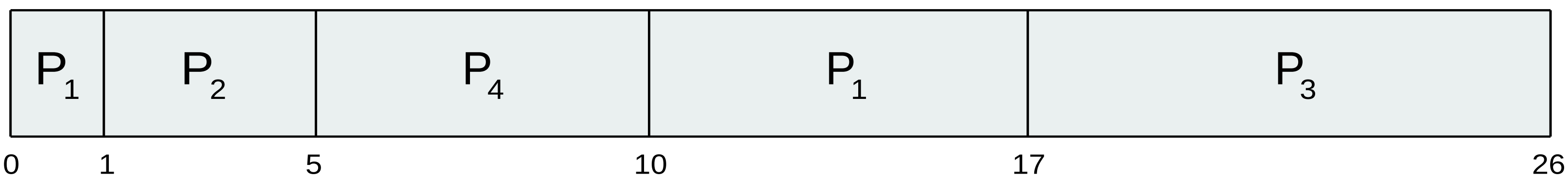
P_1	P_2	P_4	P_1	P_3
0 – 1	1 – 5	5 – 10	10 – 17	17 – 26

Waiting Time			
P_1	P_2	P_3	P_4

2.2 Shortest-remaining-time-first (Preemptive SJF)

Process	*Arrival* Time	Burst Time
P_1	0	8
P_2	1	4
P_3	2	9
P_4	3	5

- *Preemptive* SJF Gantt Chart



Waiting Time			
P_1	P_2	P_3	P_4
$10 - 1$	$1 - 1$	$17 - 2$	$5 - 3$
= 9	**= 0**	**= 15**	**= 2**

Average waiting time = [9+0+15+2]/4 = 26/4 = 6.5 msec

2.2 Shortest-remaining-time-first (Preemptive SJF)

Process	*Arrival* Time	Burst Time
P_1	0	8
P_2	1	4
P_3	2	9
P_4	3	5

- *Preemptive* SJF Gantt Chart

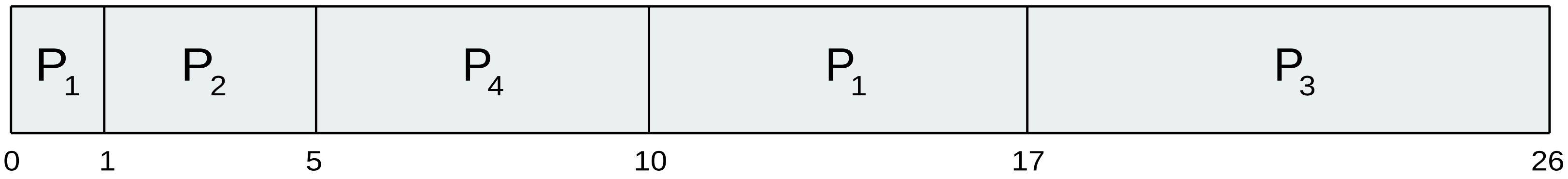
P_1	P_2	P_4	P_1	P_3	
0	1	5	10	17	26

Turnaround Time			
P_1	P_2	P_3	P_4

2.2 Shortest-remaining-time-first (Preemptive SJF)

Process	*Arrival* Time	Burst Time
P_1	0	8
P_2	1	4
P_3	2	9
P_4	3	5

- *Preemptive* SJF Gantt Chart



Turnaround Time			
P_1	P_2	P_3	P_4
17 – 0	5 – 1	26 – 2	10 – 3
= 17	**= 4**	**= 24**	**= 7**

Average turnaround time = [17+4+24+7]/4 = 52/4 = 13 msec

2.2 Shortest-remaining-time-first (Preemptive SJF)

Process	*Arrival* Time	Burst Time
P_1	0	8
P_2	1	4
P_3	2	9
P_4	3	5

- *Preemptive* SJF Gantt Chart

P_1	P_2	P_4	P_1	P_3
0–1	1–5	5–10	10–17	17–26

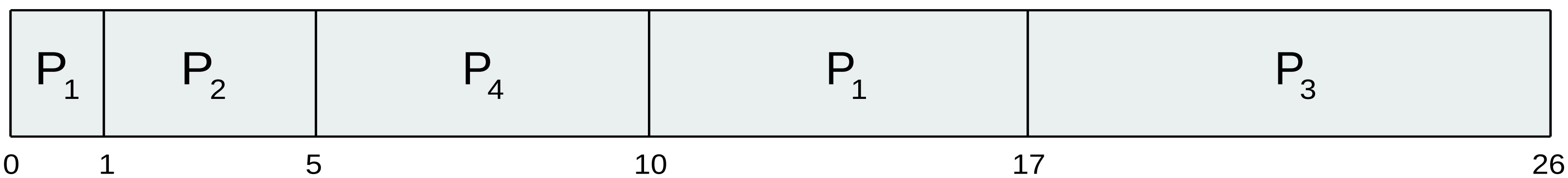
0 1 5 10 17 26

Response Time			
P_1	P_2	P_3	P_4

2.2 Shortest-remaining-time-first (Preemptive SJF)

Process	*Arrival* Time	Burst Time
P_1	0	8
P_2	1	4
P_3	2	9
P_4	3	5

- *Preemptive* SJF Gantt Chart



Response Time			
P_1	P_2	P_3	P_4
$0-0$	$1-1$	$17-2$	$5-3$
= 0	**= 0**	**= 15**	**= 2**

Average response time = [0+0+15+2]/4 = 17/4 = 4.25 msec

Scheduling Criteria (1/2)

- **CPU utilization** – keep the CPU as busy as possible.
- **Throughput** – #of processes that complete their execution per time unit.
- **Turnaround time** – amount of time to execute a particular process. (time from submission to termination)
- **Waiting time** – amount of time a process has been waiting in the ready queue.
- **Response time** – amount of time it takes from when a request was submitted until the first response is produced, not output.

Scheduling Criteria (2/2)

Scheduling Algorithm Optimization Criteria

- **Max** CPU utilization.
- **Max** throughput.
- **Min** turnaround time.
- **Min** waiting time.
- **Min** response time.

Scheduling Algorithms (1/25)

- There are many different CPU-scheduling algorithms:
  1. First Come, First Served (FCFS).
  2. Shortest Job First (SJF).
     - Preemptive SJF.
     - Non-Preemptive SJF.
  3. Priority.
  4. Round Robin.
  5. Multilevel queues.

Scheduling Algorithms (1/25)

3. Priority Scheduling

- A priority number (integer) is associated with each process
- The CPU is allocated to the process with the highest priority (smallest integer ≡ highest priority)
  - Preemptive
  - Nonpreemptive
- SJF is priority scheduling where priority is the inverse of predicted next CPU burst time
- Problem ≡ **Starvation** – low priority processes may never execute
- Solution ≡ **Aging** – as time progresses increase the priority of the process

Scheduling Algorithms (2/25)

3. Priority Scheduling

Process	Burst Time	Priority
P_1	10	3
P_2	1	1
P_3	2	4
P_4	1	5
P_5	5	2

- Priority scheduling Gantt Chart

3. Priority Scheduling

Process	Burst Time	Priority
P_1	10	3
P_2	1	1
P_3	2	4
P_4	1	5
P_5	5	2

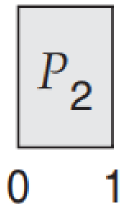
Highest priority

- Priority scheduling Gantt Chart

3. Priority Scheduling

Process	Burst Time	Priority
P_1	10	3
P_2	1	1
P_3	2	4
P_4	1	5
P_5	5	2

- Priority scheduling Gantt Chart

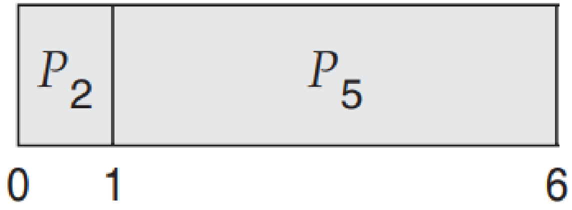


Scheduling Algorithms (3/25)

3. Priority Scheduling

Process	Burst Time	Priority
P_1	10	3
P_2	1	1
P_3	2	4
P_4	1	5
P_5	5	2

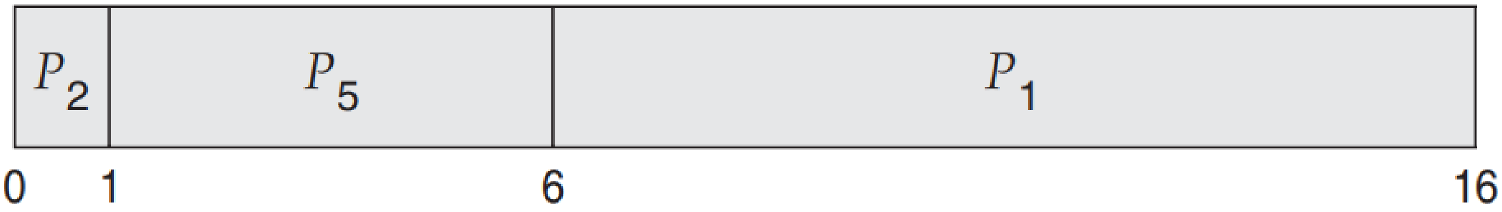
- Priority scheduling Gantt Chart



3. Priority Scheduling

Process	Burst Time	Priority
P_1	10	3
P_2	1	1
P_3	2	4
P_4	1	5
P_5	5	2

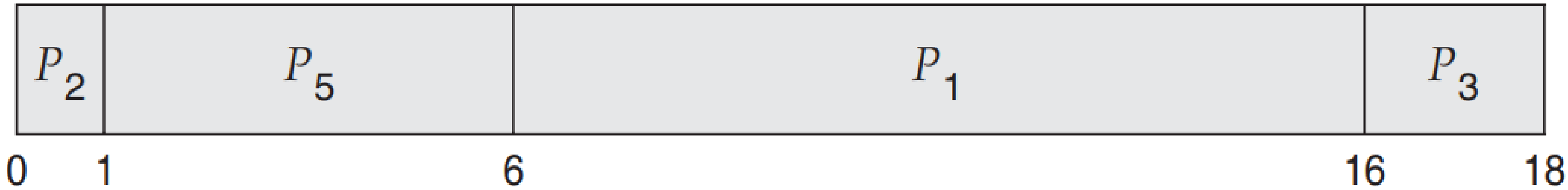
- Priority scheduling Gantt Chart



3. Priority Scheduling

Process	Burst Time	Priority
P_1	10	3
P_2	1	1
P_3	2	4
P_4	1	5
P_5	5	2

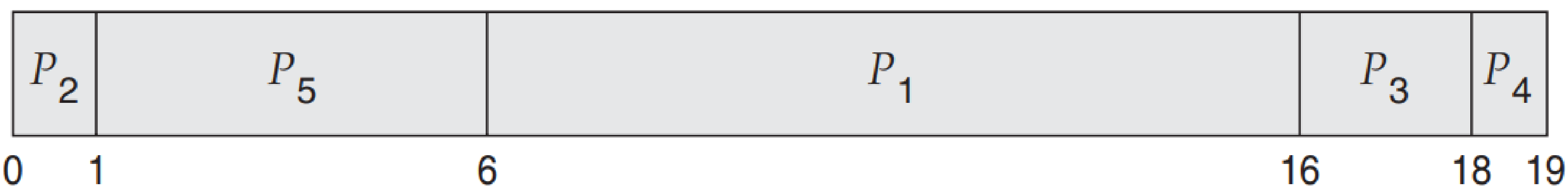
- Priority scheduling Gantt Chart



3. Priority Scheduling

Process	Burst Time	Priority
P_1	10	3
P_2	1	1
P_3	2	4
P_4	1	5
P_5	5	2

- Priority scheduling Gantt Chart

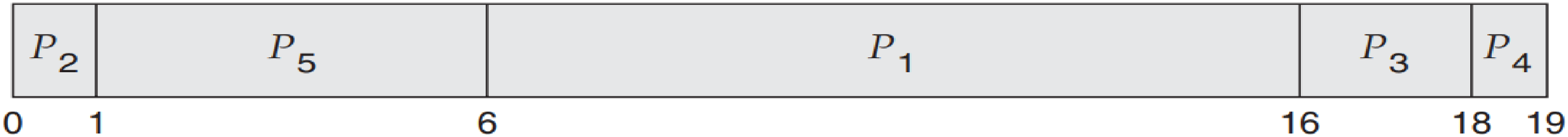


Scheduling Algorithms (4/25)

3. Priority Scheduling

Process	Burst Time	Priority
P_1	10	3
P_2	1	1
P_3	2	4
P_4	1	5
P_5	5	2

- Priority scheduling Gantt Chart



Waiting Time				
P_1	P_2	P_3	P_4	P_5

3. Priority Scheduling

Process	Burst Time	Priority
P_1	10	3
P_2	1	1
P_3	2	4
P_4	1	5
P_5	5	2

- Priority scheduling Gantt Chart

P_2	P_5	P_1	P_3	P_4

0 1 6 16 18 19

Waiting Time				
P_1	P_2	P_3	P_4	P_5
6	0	16	18	1

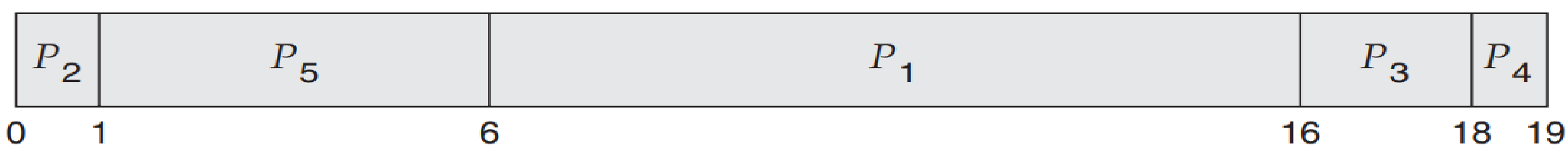
Average Waiting time = [6+0+16+18+1]/5 = 8.2 msec

Scheduling Algorithms (5/25)

3. Priority Scheduling

Process	Burst Time	Priority
P_1	10	3
P_2	1	1
P_3	2	4
P_4	1	5
P_5	5	2

- Priority scheduling Gantt Chart

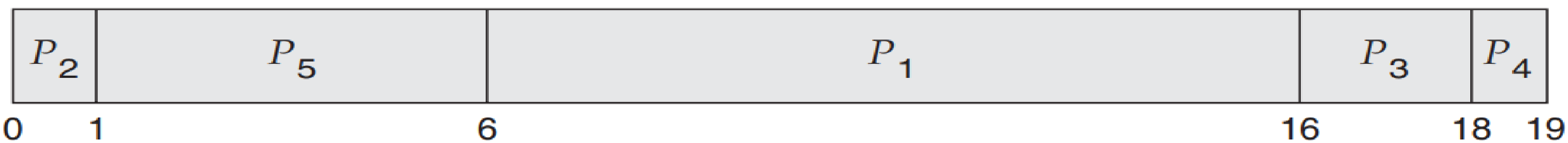


Turnaround Time				
P_1	P_2	P_3	P_4	P_5

3. Priority Scheduling

Process	Burst Time	Priority
P_1	10	3
P_2	1	1
P_3	2	4
P_4	1	5
P_5	5	2

- Priority scheduling Gantt Chart



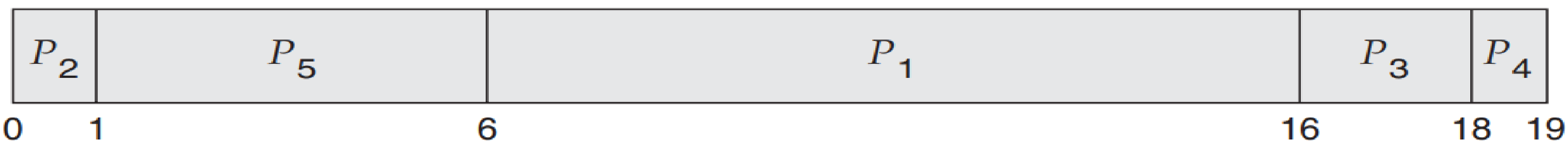
Turnaround Time				
P_1	P_2	P_3	P_4	P_5
16	1	18	19	6

Average Turnaround time = [16+1+18+19+6]/5 = 12 msec

3. Priority Scheduling

Process	Burst Time	Priority
P_1	10	3
P_2	1	1
P_3	2	4
P_4	1	5
P_5	5	2

- Priority scheduling Gantt Chart

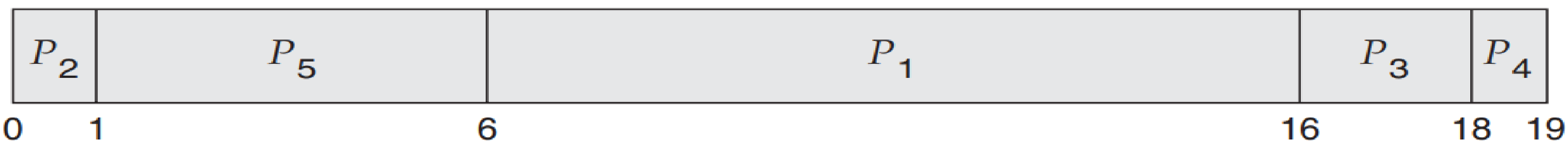


Response Time				
P_1	P_2	P_3	P_4	P_5

3. Priority Scheduling

Process	Burst Time	Priority
P_1	10	3
P_2	1	1
P_3	2	4
P_4	1	5
P_5	5	2

- Priority scheduling Gantt Chart



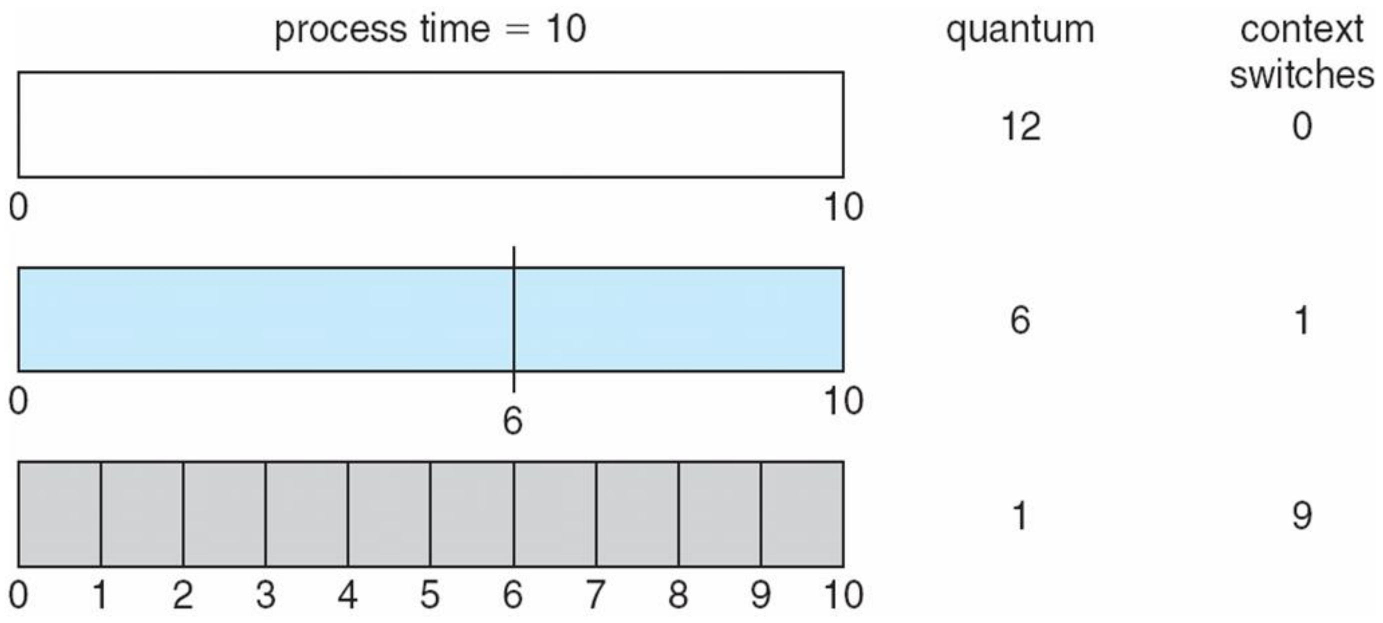
Response Time				
P_1	P_2	P_3	P_4	P_5
6	0	16	18	1

Average Response time = [6+0+16+18+1]/5 = 8.2 msec

4. Round Robin (RR) Scheduling

- Each process gets a small unit of CPU time (time quantum q), usually 10-100 milliseconds. After this time has elapsed, the process is preempted and added to the end of the ready queue.
- If there are n processes in the ready queue and the time quantum is q, then each process gets $1/n$ of the CPU time in chunks of at most q time units at once.
- No process waits more than $(n-1)q$ time units.

4. Round Robin (RR) Scheduling



4. Round Robin (RR) Scheduling

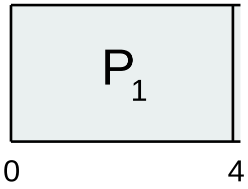
Process	Burst Time
P_1	24
P_2	3
P_3	3

- All the processes **arrive** at the same time **0**.
- Round Robin (RR) scheduling of quantum: **4** ms

4. Round Robin (RR) Scheduling

Process	Burst Time
P_1	24
P_2	3
P_3	3

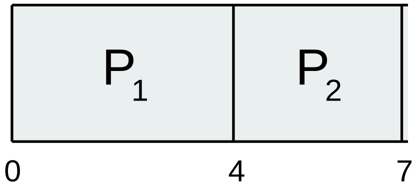
- All the processes **arrive** at the same time **0**.
- Round Robin (RR) scheduling of quantum: **4** ms



4. Round Robin (RR) Scheduling

Process	Burst Time
P_1	24
P_2	3
P_3	3

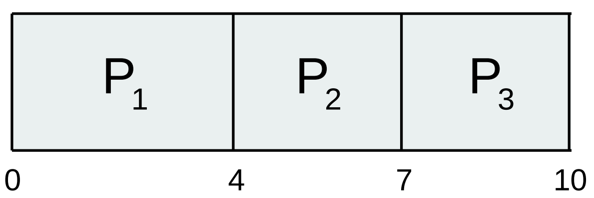
- All the processes **arrive** at the same time **0**.
- Round Robin (RR) scheduling of quantum: **4** ms



4. Round Robin (RR) Scheduling

Process	Burst Time
P_1	24
P_2	3
P_3	3

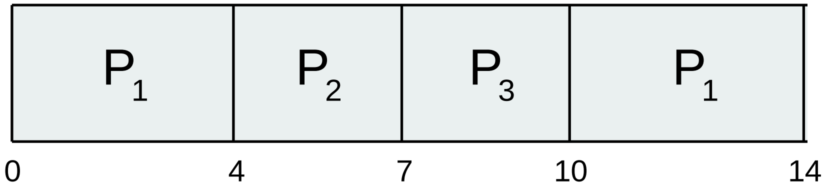
- All the processes **arrive** at the same time **0**.
- Round Robin (RR) scheduling of quantum: **4** ms



4. Round Robin (RR) Scheduling

Process	Burst Time
P_1	24
P_2	3
P_3	3

- All the processes **arrive** at the same time **0**.
- Round Robin (RR) scheduling of quantum: **4** ms



4. Round Robin (RR) Scheduling

Process	Burst Time
P_1	24
P_2	3
P_3	3

- All the processes **arrive** at the same time **0**.
- Round Robin (RR) scheduling of quantum: **4** ms

P_1	P_2	P_3	P_1	P_1	
0	4	7	10	14	18

4. Round Robin (RR) Scheduling

Process	Burst Time
P_1	24
P_2	3
P_3	3

- All the processes **arrive** at the same time **0**.
- Round Robin (RR) scheduling of quantum: **4** ms

P_1	P_2	P_3	P_1	P_1	P_1
0–4	4–7	7–10	10–14	14–18	18–22

0 4 7 10 14 18 22

4. Round Robin (RR) Scheduling

Process	Burst Time
P_1	24
P_2	3
P_3	3

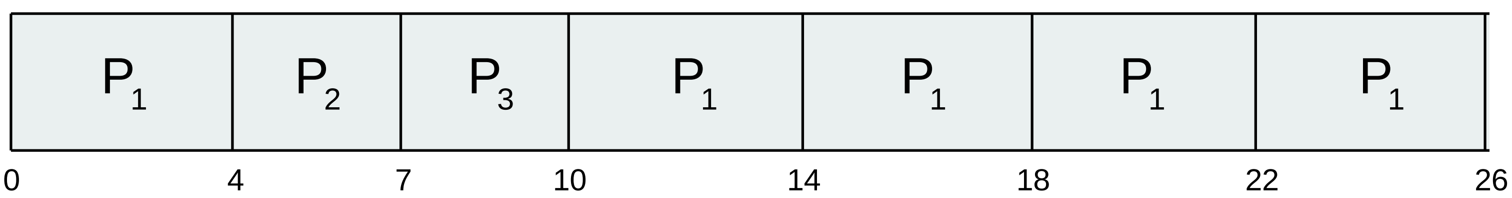
- All the processes **arrive** at the same time **0**.
- Round Robin (RR) scheduling of quantum: **4** ms

P_1	P_2	P_3	P_1	P_1	P_1	P_1
0–4	4–7	7–10	10–14	14–18	18–22	22–26

4. Round Robin (RR) Scheduling

Process	Burst Time
P_1	24
P_2	3
P_3	3

- All the processes **arrive** at the same time **0**.
- Round Robin (RR) scheduling of quantum: **4** ms



4. Round Robin (RR) Scheduling

Process	Burst Time
P_1	24
P_2	3
P_3	3

- All the processes **arrive** at the same time **0**.
- Round Robin (RR) scheduling of quantum: **4** ms

P_1	P_2	P_3	P_1	P_1	P_1	P_1	P_1

0 4 7 10 14 18 22 26 30

4. Round Robin (RR) Scheduling

Process	Burst Time
P_1	24
P_2	3
P_3	3

- All the processes **arrive** at the same time **0**.
- Round Robin (RR) scheduling of quantum: **4** ms

P_1	P_2	P_3	P_1	P_1	P_1	P_1	P_1

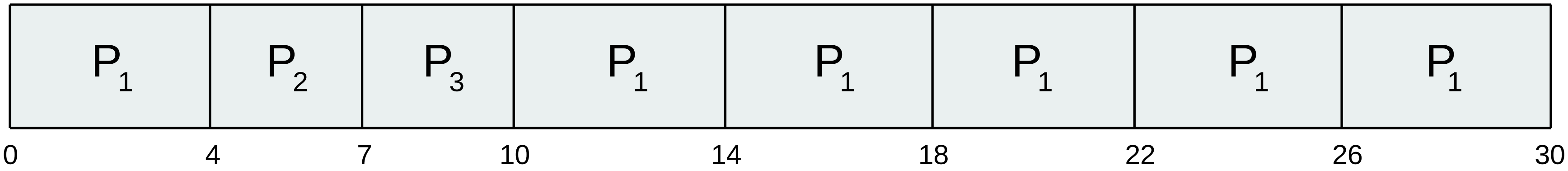
0 4 7 10 14 18 22 26 30

- # of context switches = ??

4. Round Robin (RR) Scheduling

Process	Burst Time
P_1	24
P_2	3
P_3	3

- All the processes **arrive** at the same time **0**.
- Round Robin (RR) scheduling of quantum: **4** ms



- # of context switches = 7

4. Round Robin (RR) Scheduling

Process	Burst Time
P_1	24
P_2	3
P_3	3

- All the processes **arrive** at the same time **0**.
- Round Robin (RR) scheduling of quantum: **4** ms

P_1	P_2	P_3	P_1	P_1	P_1	P_1	P_1

0 4 7 10 14 18 22 26 30

Waiting Time		
P_1	P_2	P_3

4. Round Robin (RR) Scheduling

Process	Burst Time
P_1	24
P_2	3
P_3	3

- All the processes **arrive** at the same time **0**.
- Round Robin (RR) scheduling of quantum: **4** ms

P_1	P_2	P_3	P_1	P_1	P_1	P_1	P_1
0 – 4	7	10	14	18	22	26	30

Waiting Time		
P_1	P_2	P_3
$0 + (10 - 4)$	4	7

Average waiting time: (6 + 4 + 7)/3 = 5.667 ms

4. Round Robin (RR) Scheduling

Process	Burst Time
P_1	24
P_2	3
P_3	3

- All the processes **arrive** at the same time **0**.
- Round Robin (RR) scheduling of quantum: **4** ms

P_1	P_2	P_3	P_1	P_1	P_1	P_1	P_1

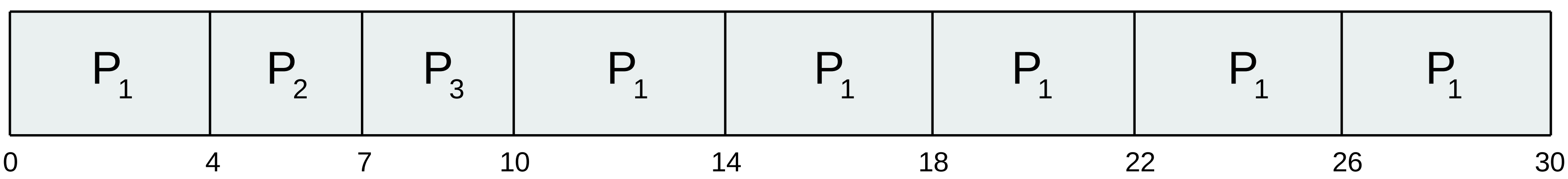
0 4 7 10 14 18 22 26 30

Turnaround Time		
P_1	P_2	P_3

4. Round Robin (RR) Scheduling

Process	Burst Time
P_1	24
P_2	3
P_3	3

- All the processes **arrive** at the same time **0**.
- Round Robin (RR) scheduling of quantum: **4** ms



Turnaround Time		
P_1	P_2	P_3
30	7	10

Average Turnaround time: (30 + 7 + 10)/3 = 15.667 ms

4. Round Robin (RR) Scheduling

Process	Burst Time
P_1	24
P_2	3
P_3	3

- All the processes **arrive** at the same time **0**.
- Round Robin (RR) scheduling of quantum: **4** ms

P_1	P_2	P_3	P_1	P_1	P_1	P_1	P_1

0 4 7 10 14 18 22 26 30

Response Time		
P_1	P_2	P_3

4. Round Robin (RR) Scheduling

Process	Burst Time
P_1	24
P_2	3
P_3	3

- All the processes **arrive** at the same time **0**.
- Round Robin (RR) scheduling of quantum: **4** ms

P_1	P_2	P_3	P_1	P_1	P_1	P_1	P_1

0, 4, 7, 10, 14, 18, 22, 26, 30

Response Time		
P_1	P_2	P_3
0	4	7

Average Response time: (0 + 4 + 7)/3 = 3.667 ms

Scheduling Algorithms (14/25)

- There are many different CPU-scheduling algorithms:
  1. First Come, First Served (FCFS).
  2. Shortest Job First (SJF).
     - Preemptive SJF.
     - Non-Preemptive SJF.
  3. Priority.
  4. Round Robin.
  5. Multilevel queues.

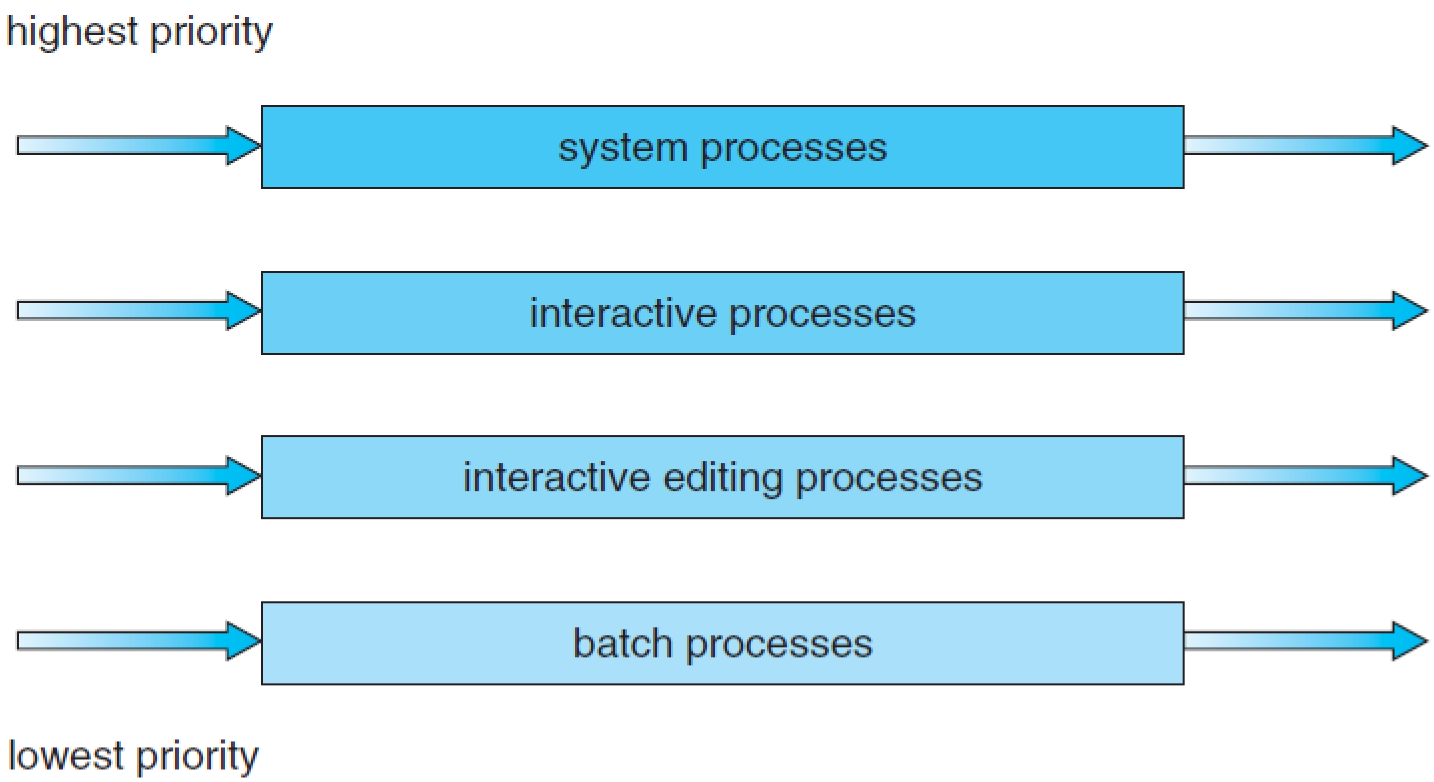
5. Multilevel Queue Scheduling

- Ready queue is partitioned into separate queues, eg:
  - **foreground** (interactive)
  - **background** (batch)
- Process permanently in a given queue
- Each queue has its own scheduling algorithm:
  - foreground – RR.
  - background – FCFS.

5. Multilevel Queue Scheduling

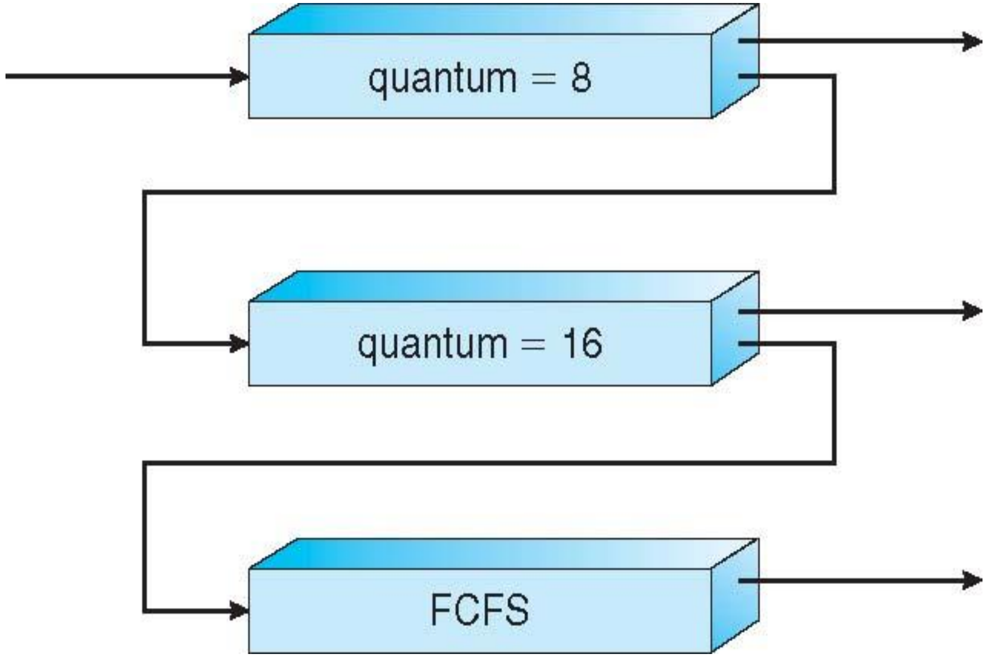
- Scheduling must be done between the queues:
  - Fixed priority scheduling; (i.e., serve all from foreground then from background). Possibility of starvation.
  - Time slice – each queue gets a certain amount of CPU time which it can schedule amongst its processes; i.e., 80% to foreground in RR.
  - 20% to background in FCFS.

5. Multilevel Queue Scheduling

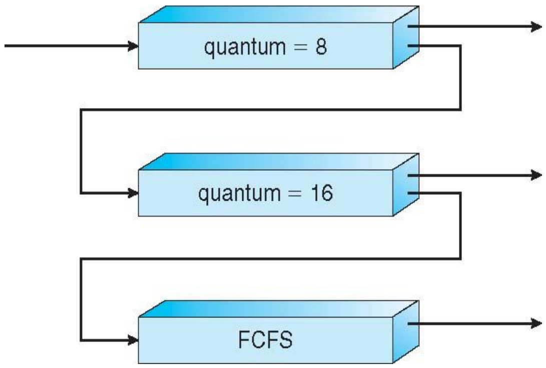


5. Multilevel Queue Scheduling

- Three queues:
  - Q_0 – **RR** with time quantum **8** milliseconds
  - Q_1 – **RR** time quantum **16** milliseconds
  - Q_2 – **FCFS**



5. Multilevel Queue Scheduling



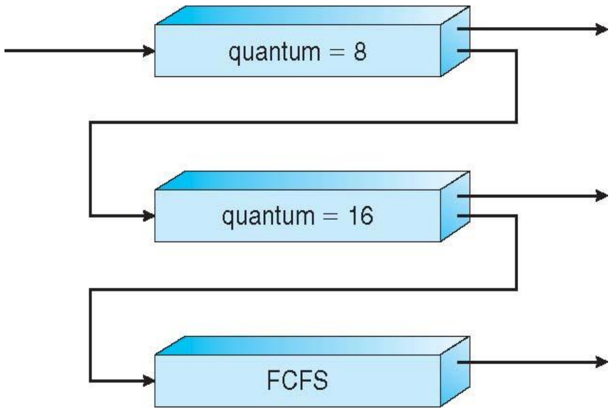
- Scheduling
- A new job enters queue Q_0 which is served FCFS
  - When it gains CPU, job receives 8 milliseconds.
  - If it does not finish in 8 milliseconds, job is moved to queue Q_1.
- At Q_1 job is again served FCFS and receives 16 additional milliseconds
  - If it still does not complete, it is preempted and moved to queue Q_2.

5. Multilevel Queue Scheduling

- Now we add the concepts of varying arrival times and preemption to the analysis

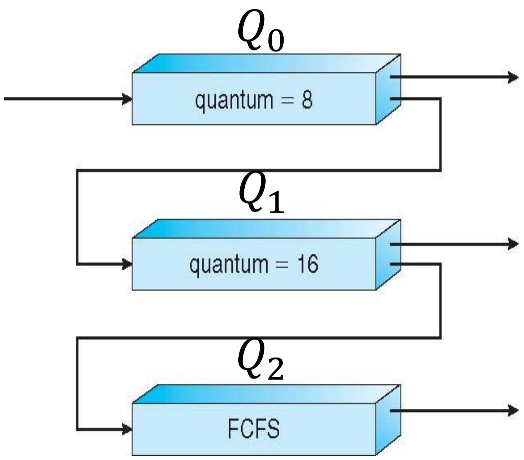
Process	***Arrival*** **Time**	Burst Time
P_1	0	7
P_2	1	60
P_3	2	20
P_4	3	40

Using multi-processors or multi-core processor

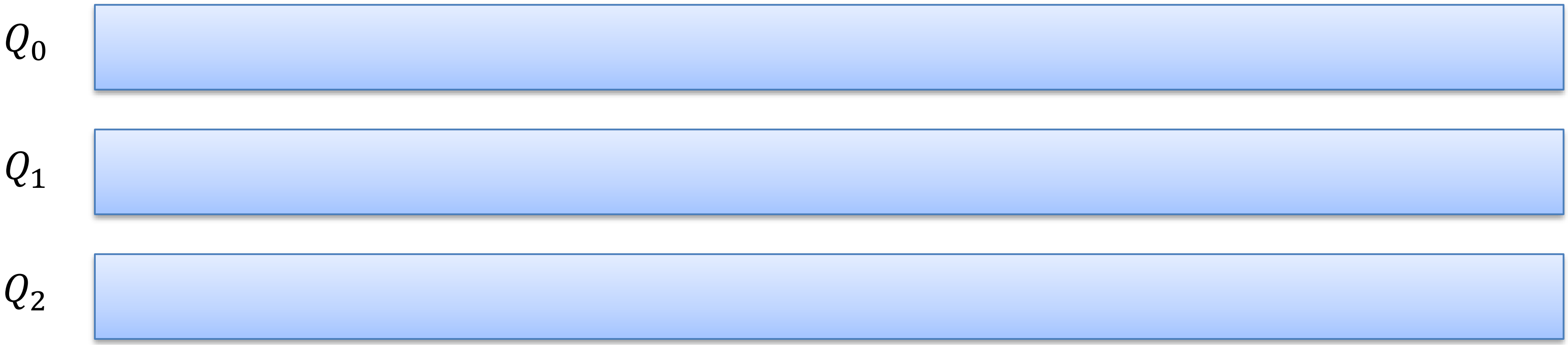


Multilevel Queue Fixed priority
non-preemptive

5. Multilevel Queue Scheduling

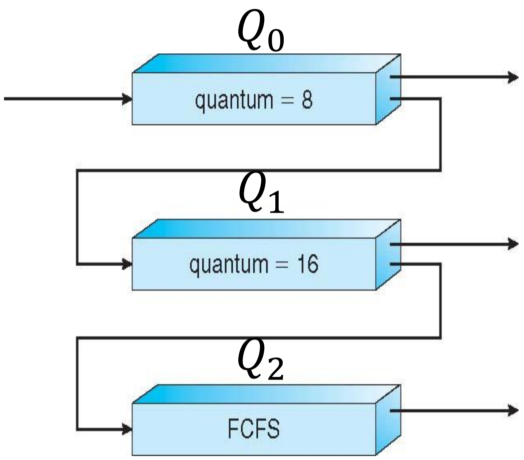


Process	***Arrival* Time**	Burst Time
P_1	0	7
P_2	1	60
P_3	2	20
P_4	3	40



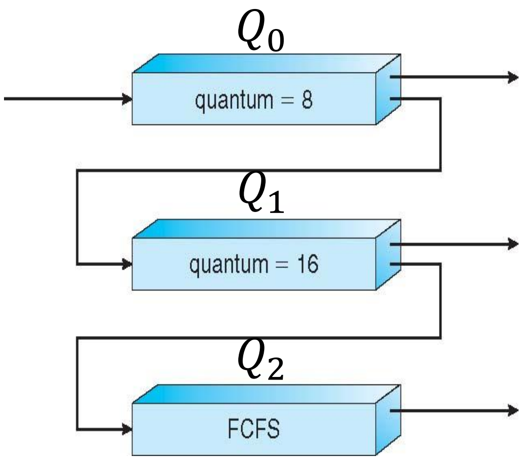
5. Multilevel Queue Scheduling

Process	***Arrival* Time**	Burst Time
P_1	0	~~7~~
P_2	1	60
P_3	2	20
P_4	3	40



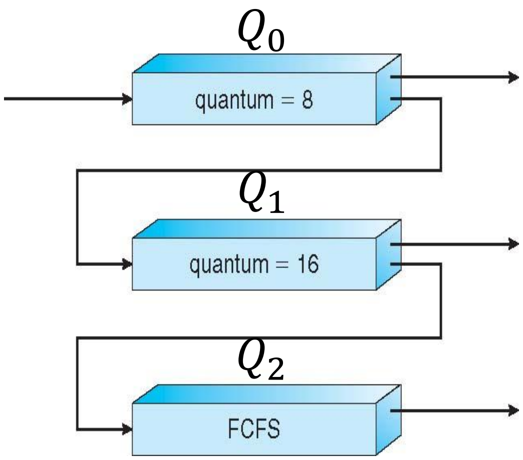
5. Multilevel Queue Scheduling

Process	***Arrival* Time**	Burst Time
$\cancel{P_1}$	0	$\cancel{7}$
P_2	1	60
P_3	2	20
P_4	3	40

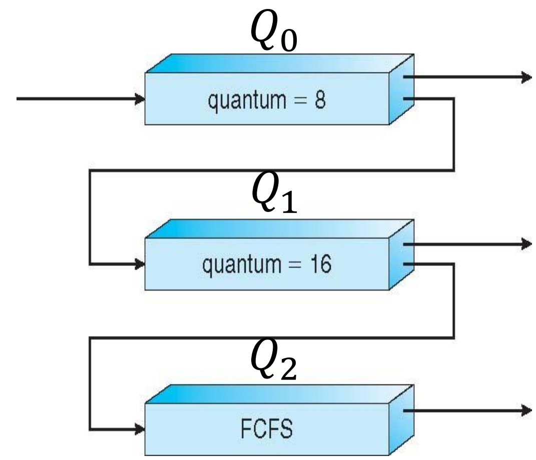


5. Multilevel Queue Scheduling

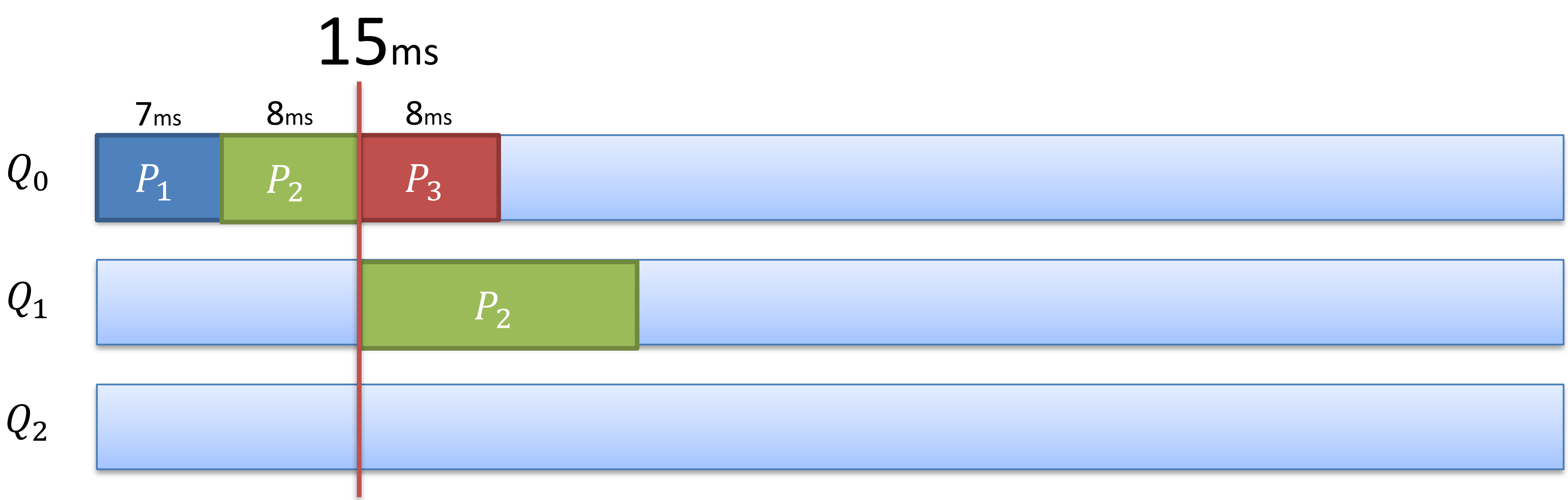
Process	***Arrival*** **Time**	Burst Time
$\cancel{P_1}$	0	$\cancel{7}$
P_2	1	$\cancel{60}$ 52
P_3	2	20
P_4	3	40



5. Multilevel Queue Scheduling

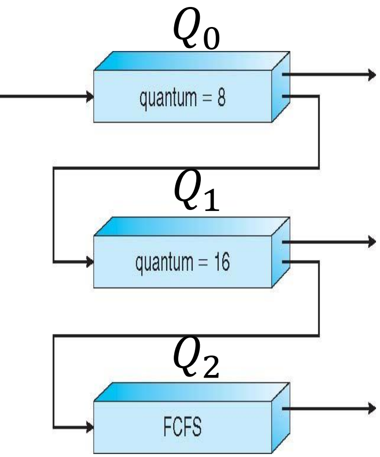


Process	***Arrival*** **Time**	Burst Time
$\cancel{P_1}$	0	$\cancel{7}$
P_2	1	$\cancel{60}$ 52
P_3	2	20
P_4	3	40

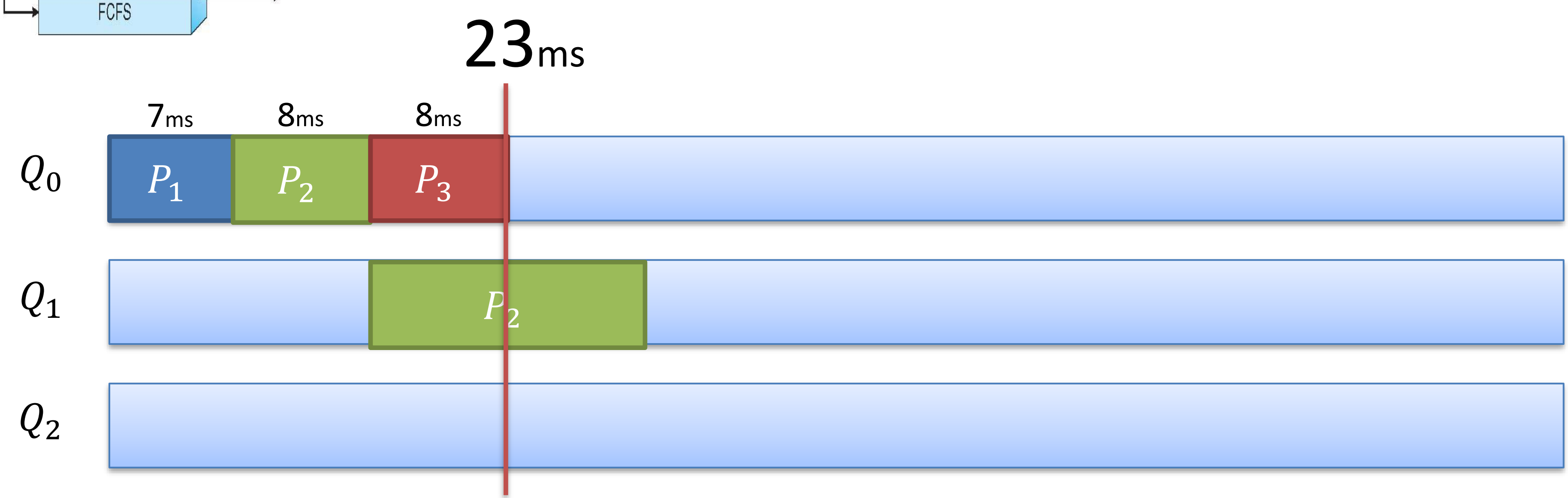


Scheduling Algorithms (21/32)

5. Multilevel Queue Scheduling

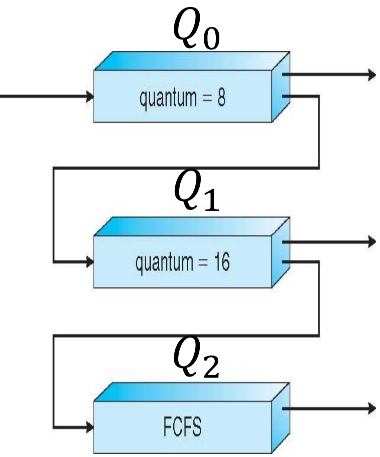


Process	***Arrival*** **Time**	Burst Time
$\cancel{P_1}$	0	$\cancel{7}$
P_2	1	$\cancel{60}$ $\cancel{52}$ 44
P_3	2	$\cancel{20}$ 12
P_4	3	40

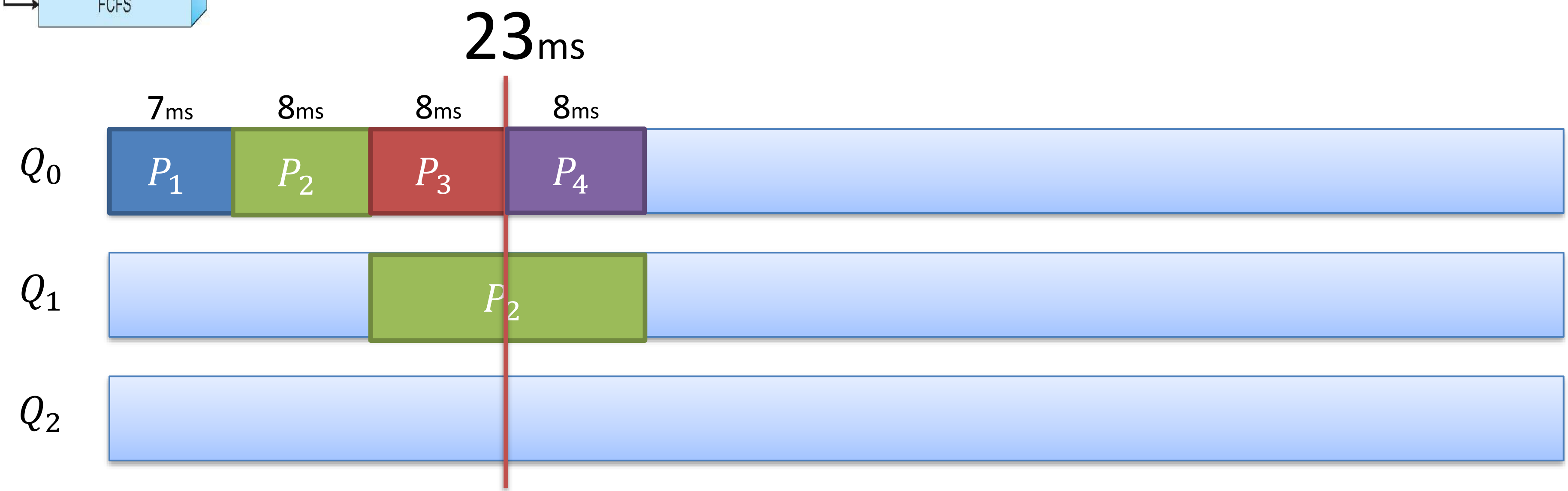


Scheduling Algorithms (21/32)

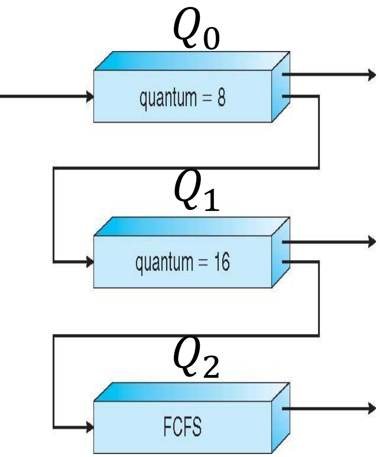
5. Multilevel Queue Scheduling



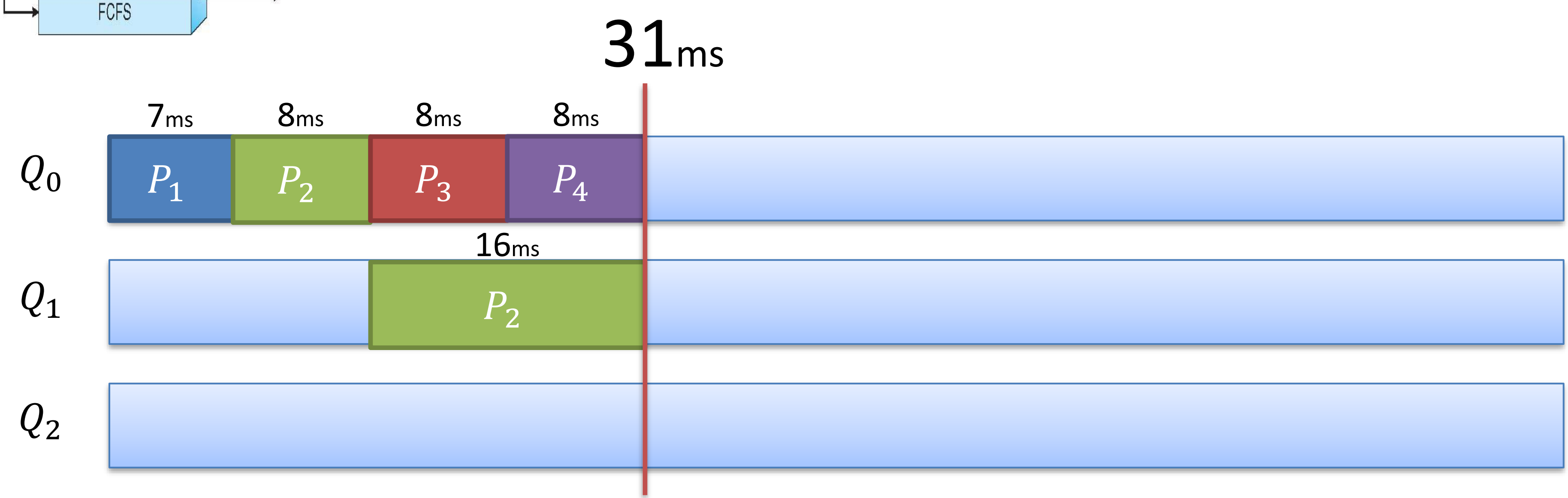
Process	***Arrival*** **Time**	Burst Time
~~P_1~~	0	~~7~~
P_2	1	~~60~~ ~~52~~ 44
P_3	2	~~20~~ 12
P_4	3	40



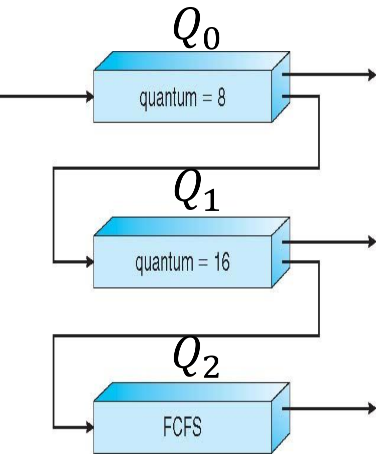
5. Multilevel Queue Scheduling



Process	***Arrival* Time**	Burst Time
~~P_1~~	0	~~7~~
P_2	1	~~60~~ ~~52~~ ~~44~~ 36
P_3	2	~~20~~ 12
P_4	3	~~40~~ 32

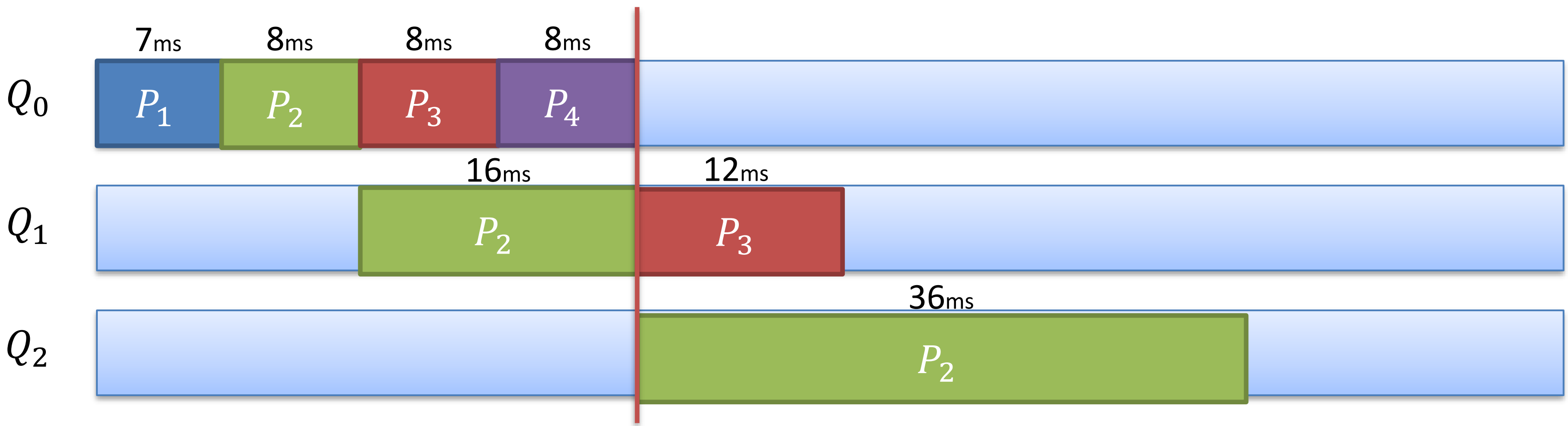


5. Multilevel Queue Scheduling



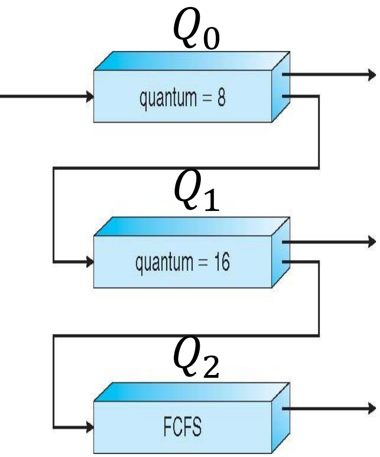
Process	***Arrival* Time**	Burst Time
~~P_1~~	0	~~7~~
P_2	1	~~60~~ ~~52~~ ~~44~~ 36
P_3	2	~~20~~ 12
P_4	3	~~40~~ 32

31ms

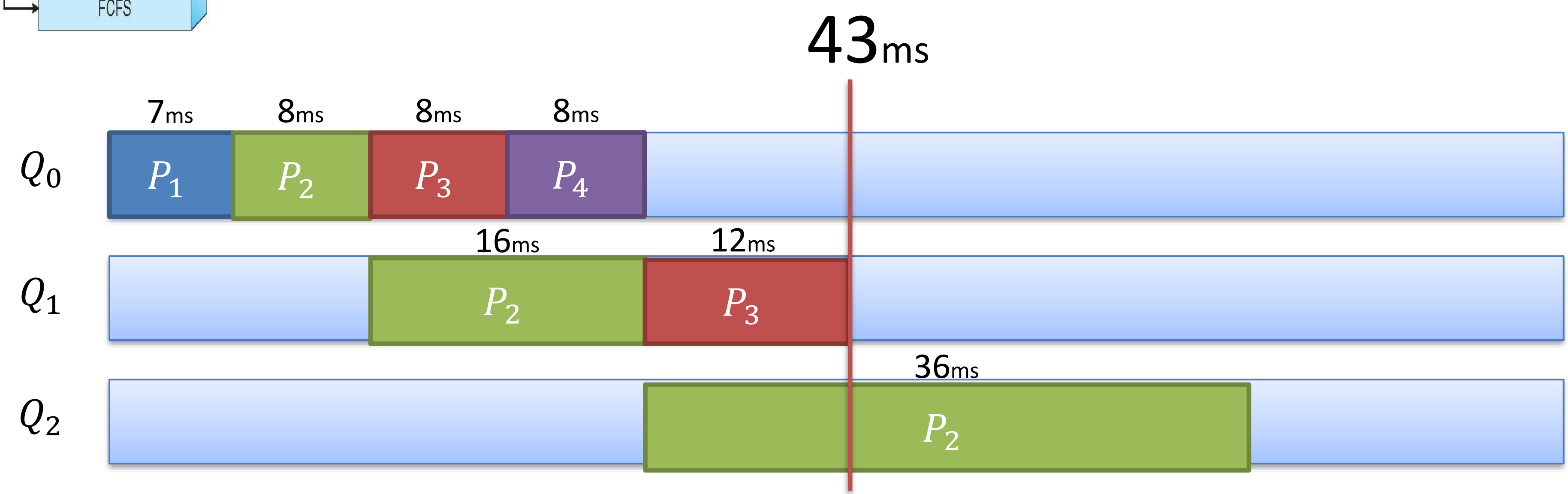


Scheduling Algorithms (21/32)

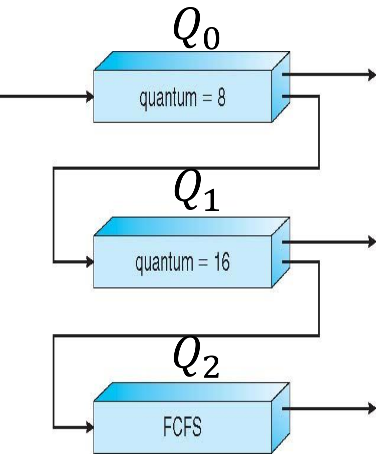
5. Multilevel Queue Scheduling



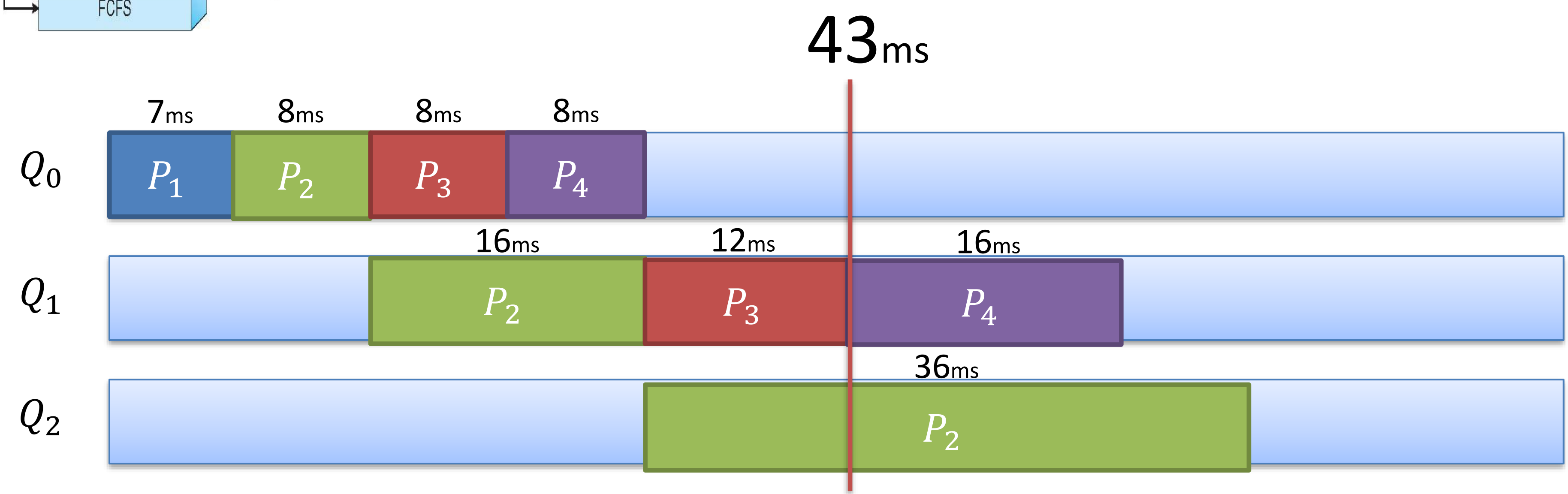
Process	***Arrival*** **Time**	Burst Time
~~P_1~~	0	~~7~~
P_2	1	~~60~~ ~~52~~ ~~44~~ ~~36~~ 24
P_3	2	~~20~~ ~~12~~
P_4	3	~~40~~ 32



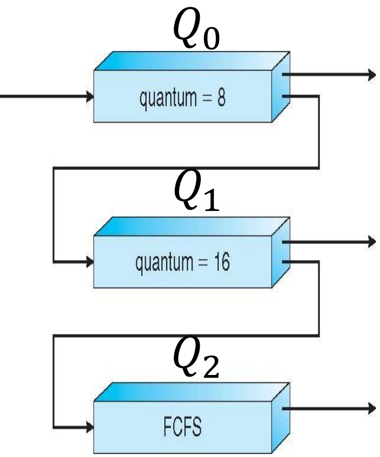
5. Multilevel Queue Scheduling



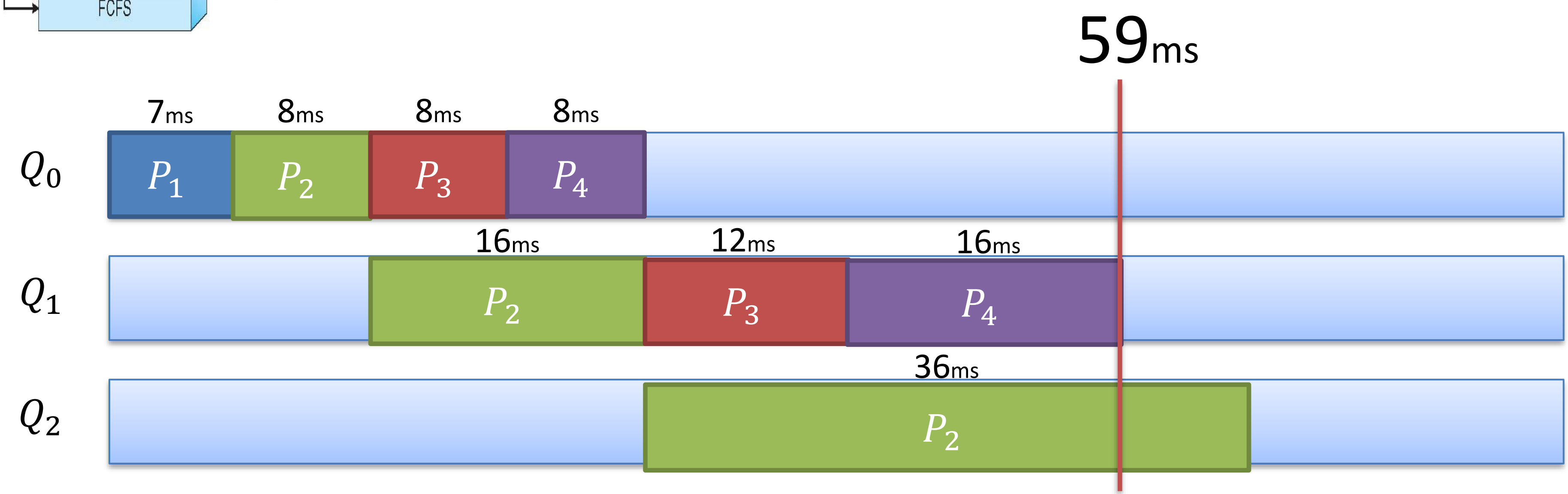
Process	***Arrival* Time**	Burst Time
~~P_1~~	0	~~7~~
P_2	1	~~60~~ ~~52~~ ~~44~~ ~~36~~ 24
~~P_3~~	2	~~20~~ ~~12~~
P_4	3	~~40~~ 32



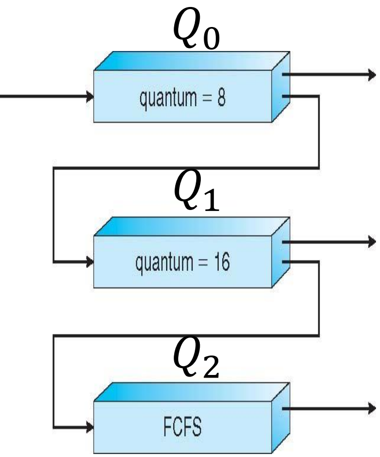
5. Multilevel Queue Scheduling



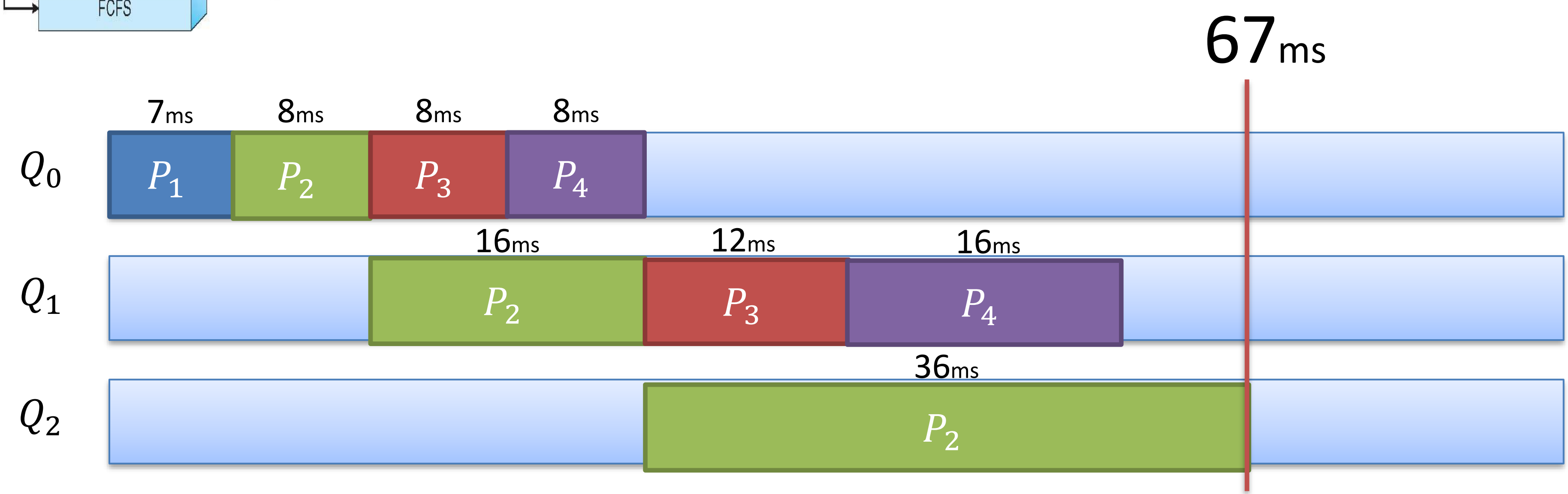
Process	***Arrival*** **Time**	Burst Time
$\cancel{P_1}$	0	~~7~~
P_2	1	~~60~~ ~~52~~ ~~44~~ ~~36~~ ~~24~~ 8
$\cancel{P_3}$	2	~~20~~ ~~12~~
P_4	3	~~40~~ ~~32~~ 16



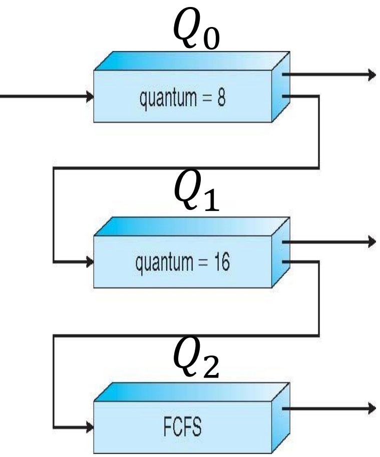
5. Multilevel Queue Scheduling



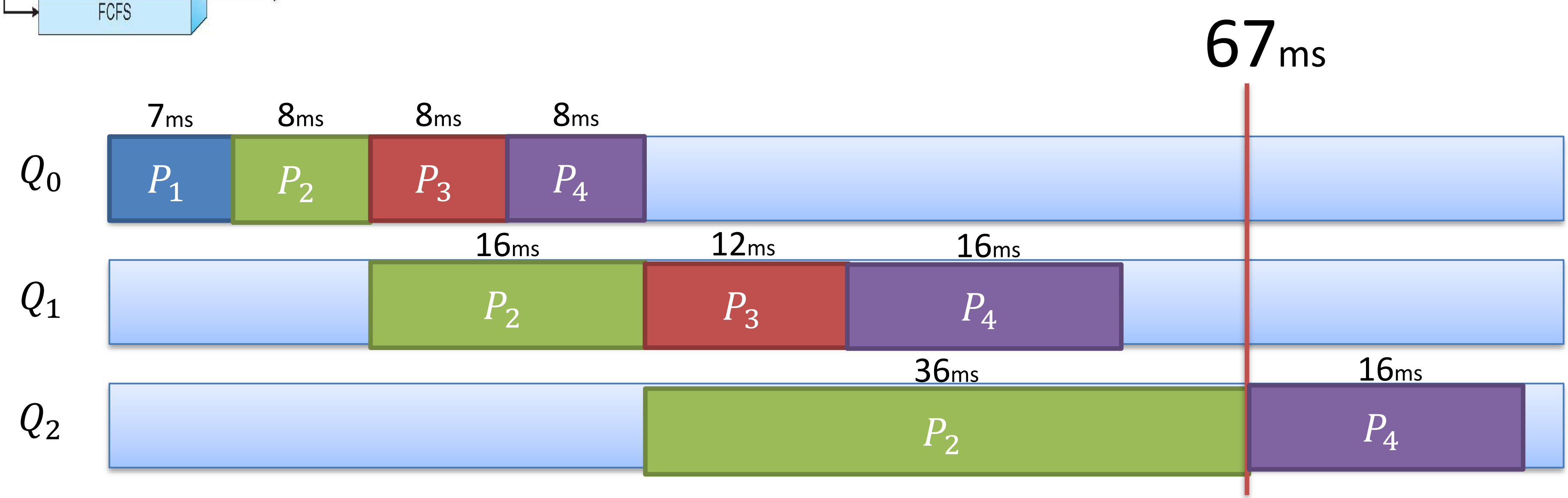
Process	***Arrival*** **Time**	Burst Time
~~P_1~~	0	~~7~~
P_2	1	~~60~~ ~~52~~ ~~44~~ ~~36~~ ~~24~~ ~~8~~
~~P_3~~	2	~~20~~ ~~12~~
P_4	3	~~40~~ ~~32~~ 16



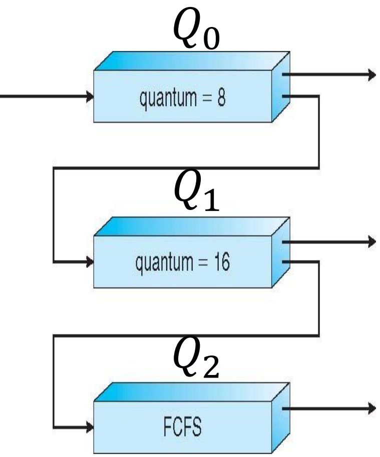
5. Multilevel Queue Scheduling



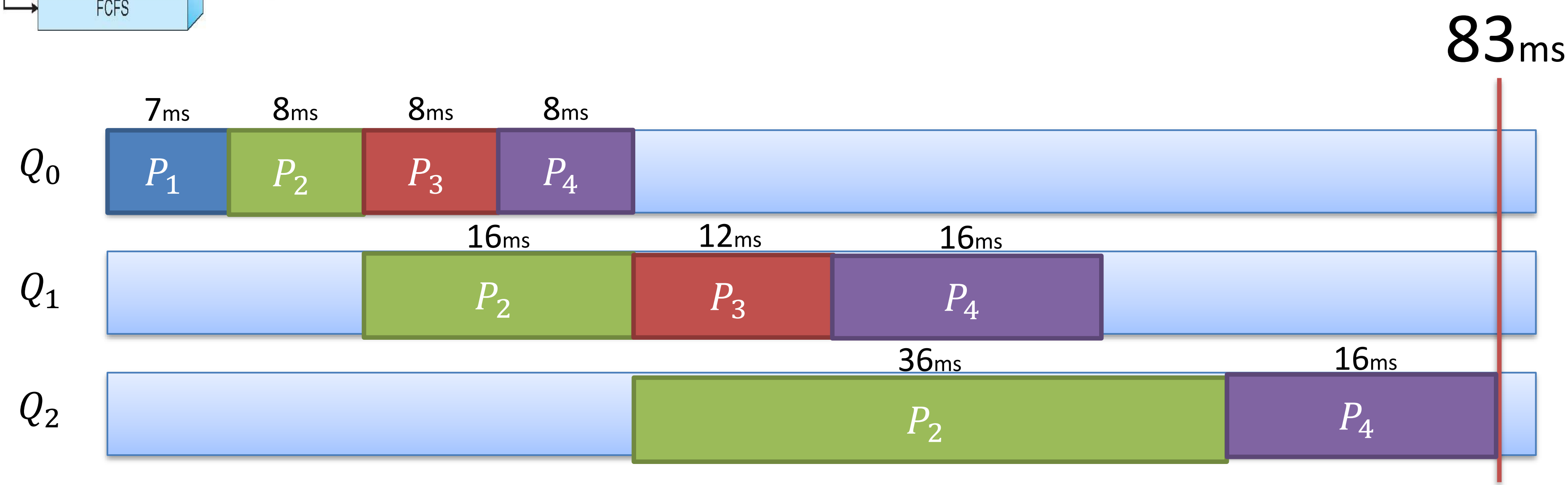
Process	***Arrival*** **Time**	Burst Time
$\cancel{P_1}$	0	~~7~~
$\cancel{P_2}$	1	~~60~~ ~~52~~ ~~44~~ ~~36~~ ~~24~~ ~~8~~
$\cancel{P_3}$	2	~~20~~ ~~12~~
P_4	3	~~40~~ ~~32~~ 16



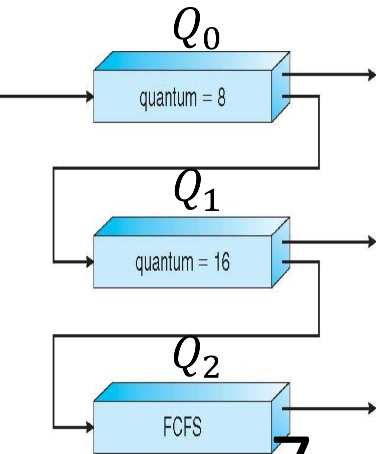
5. Multilevel Queue Scheduling



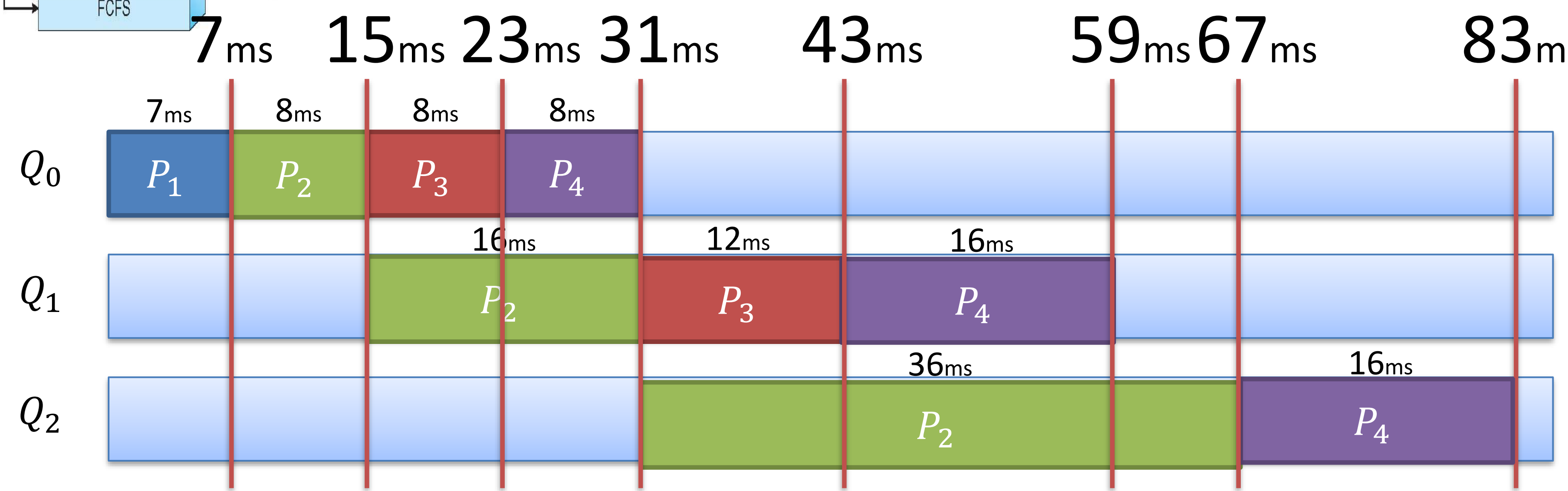
Process	***Arrival* Time**	Burst Time
~~P_1~~	0	~~7~~
~~P_2~~	1	~~60~~ ~~52~~ ~~44~~ ~~36~~ ~~24~~ ~~8~~
~~P_3~~	2	~~20~~ ~~12~~
~~P_4~~	3	~~40~~ ~~32~~ ~~16~~



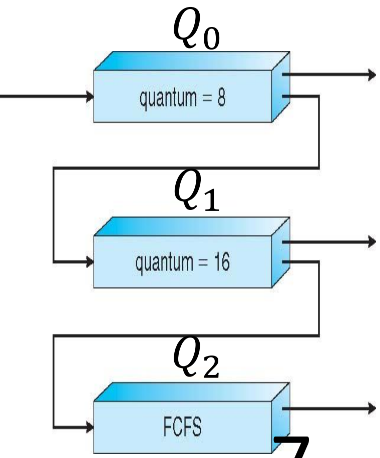
5. Multilevel Queue Scheduling



Process	***Arrival*** **Time**	Burst Time
~~P_1~~	0	~~7~~
~~P_2~~	1	~~60~~ ~~52~~ ~~44~~ ~~36~~ ~~24~~ ~~8~~
~~P_3~~	2	~~20~~ ~~12~~
~~P_4~~	3	~~40~~ ~~32~~ ~~16~~

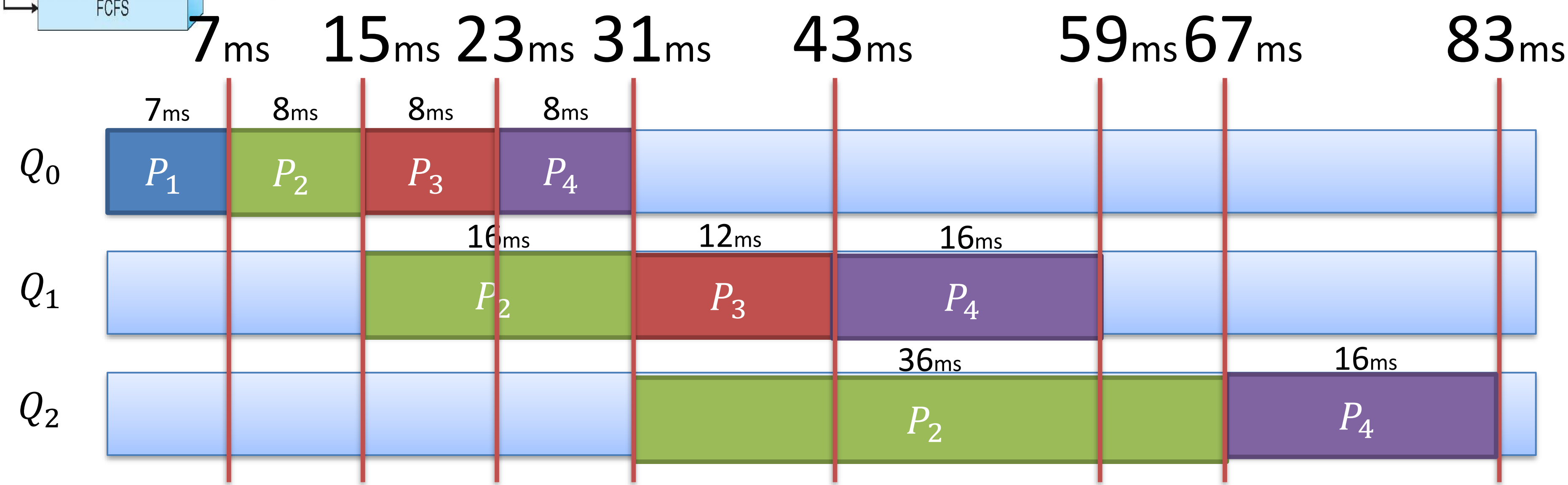


5. Multilevel Queue Scheduling

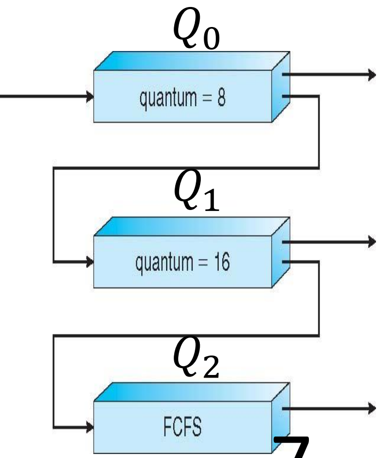


Process	*Arrival* Time
P_1	0
P_2	1
P_3	2
P_4	3

Waiting Time			
P_1	P_2	P_3	P_4

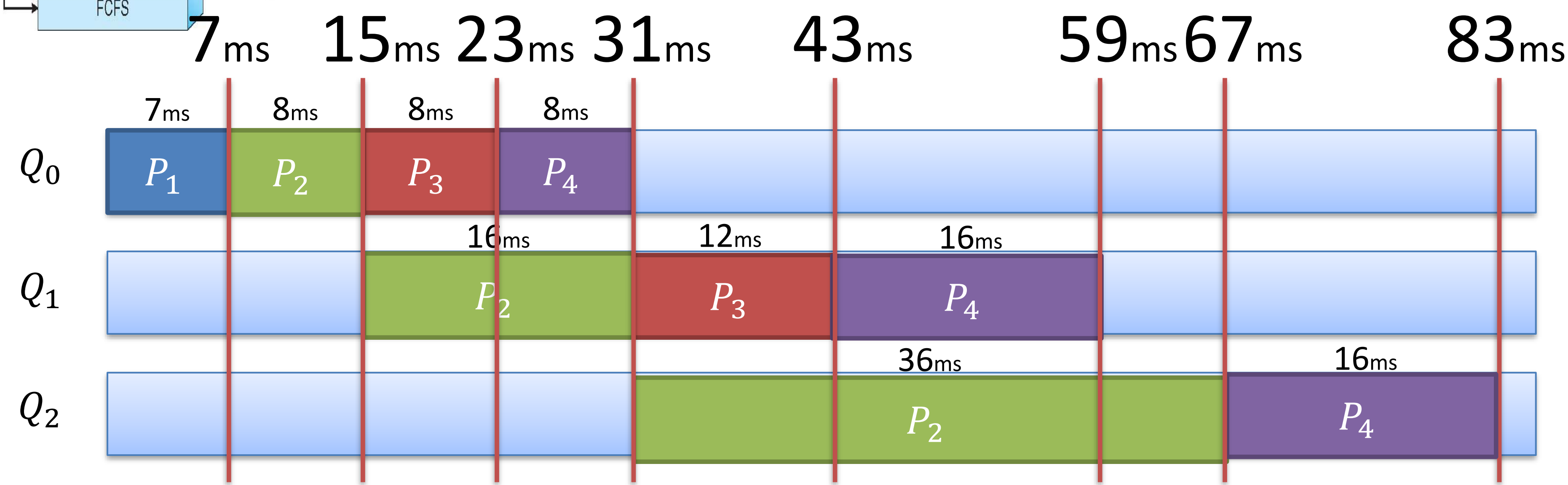


5. Multilevel Queue Scheduling

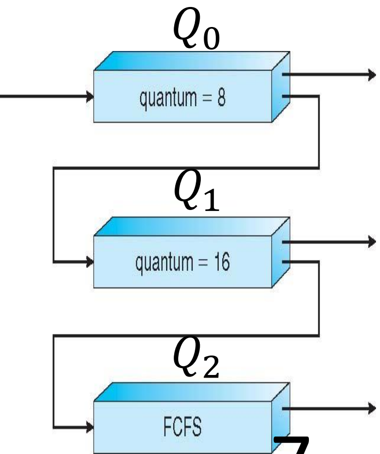


Process	***Arrival*** **Time**
P_1	0
P_2	1
P_3	2
P_4	3

Waiting Time			
P_1	P_2	P_3	P_4
0			
= 0			

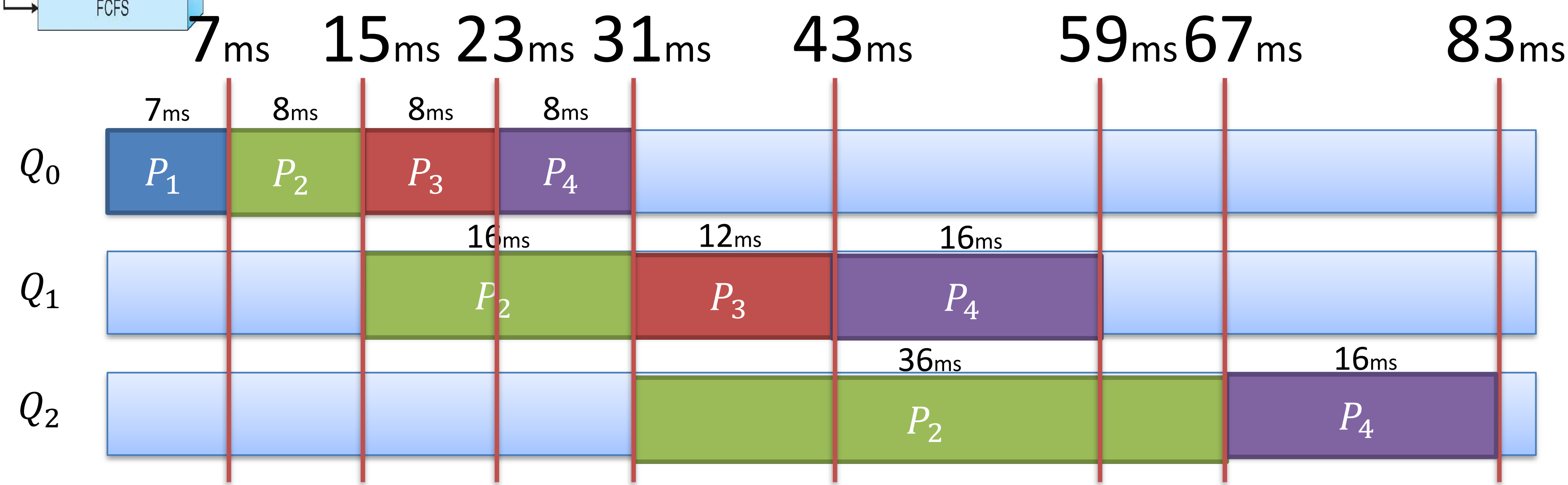


5. Multilevel Queue Scheduling

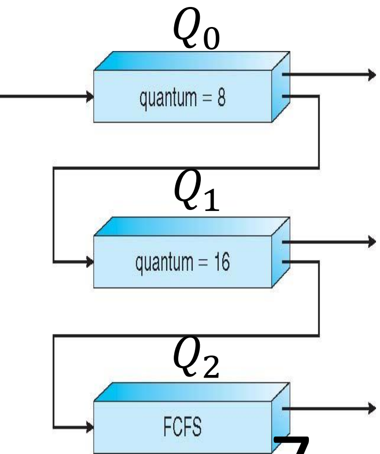


Process	***Arrival*** **Time**
P_1	0
P_2	1
P_3	2
P_4	3

Waiting Time			
P_1	P_2	P_3	P_4
0	7 − 1		
= **0**	= **6**		

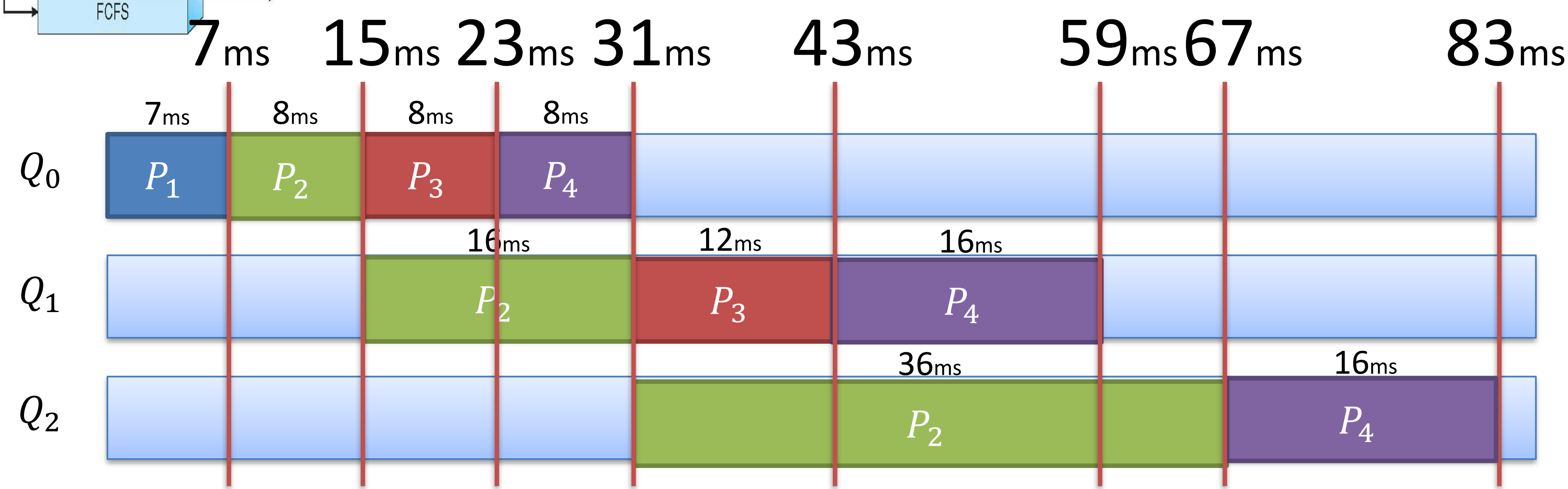


5. Multilevel Queue Scheduling

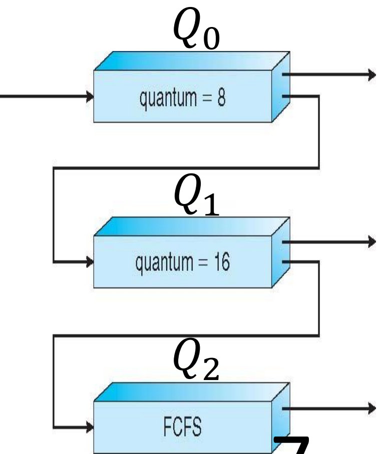


Process	***Arrival*** **Time**
P_1	0
P_2	1
P_3	2
P_4	3

Waiting Time			
P_1	P_2	P_3	P_4
0	$7-1$	$15+8-2$	
= 0	**= 6**	**= 21**	

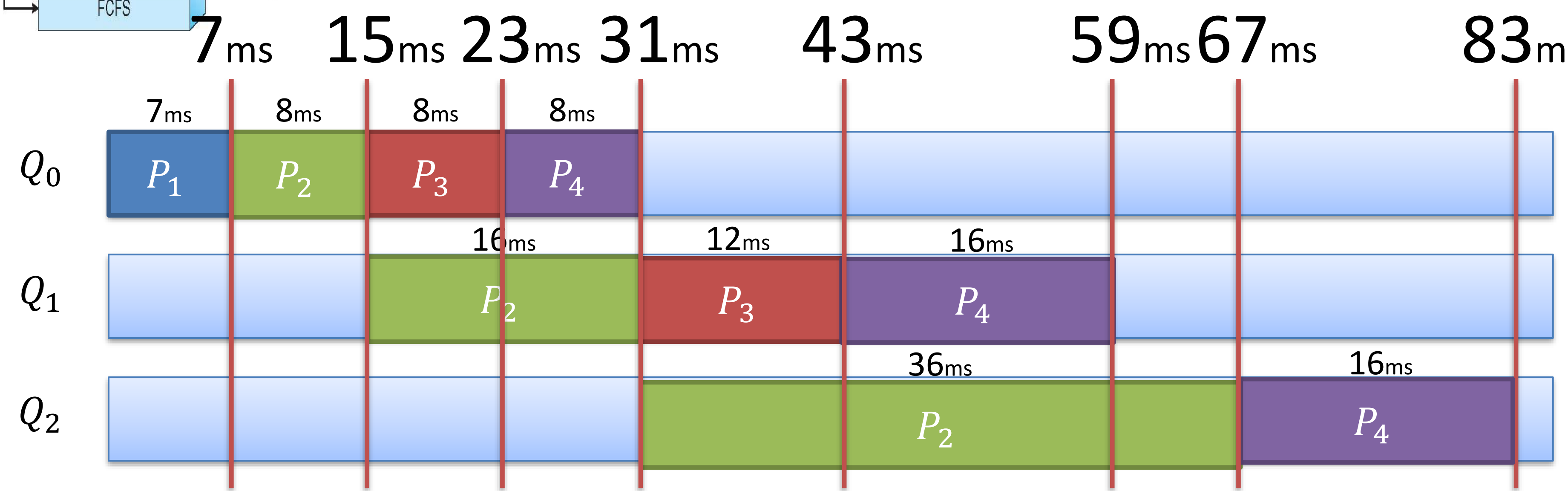


5. Multilevel Queue Scheduling

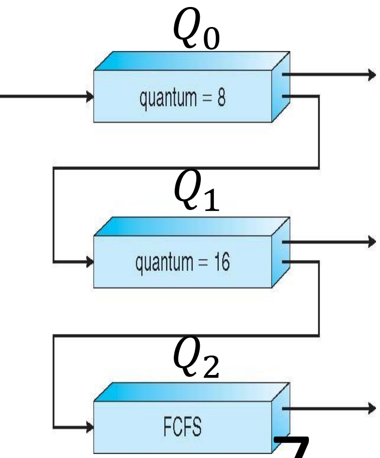


Process	Arrival Time
P_1	0
P_2	1
P_3	2
P_4	3

Waiting Time			
P_1	P_2	P_3	P_4
0	$7-1$	$15+8-2$	$23+12+8-3$
= 0	**= 6**	**= 21**	**= 40**

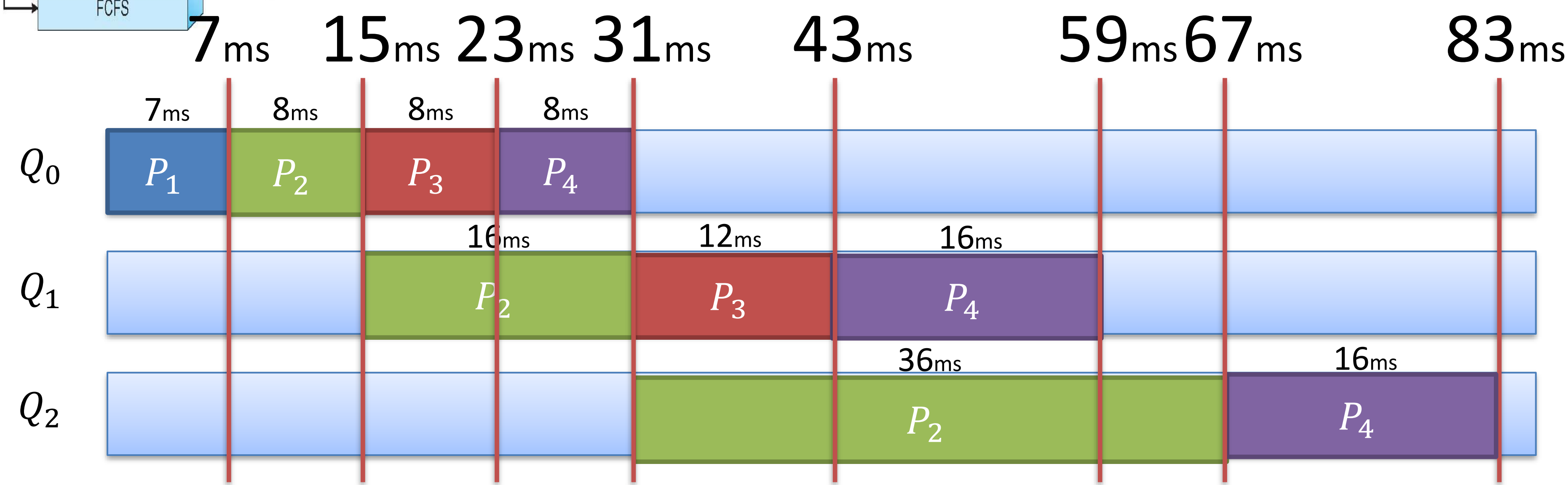


5. Multilevel Queue Scheduling

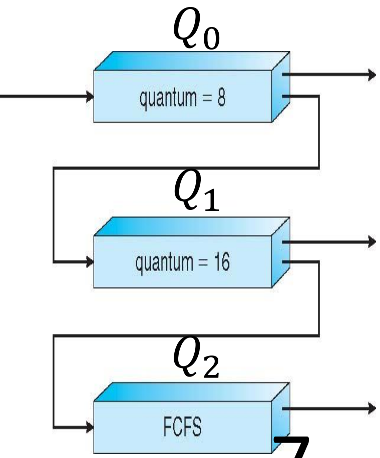


Process	***Arrival*** **Time**
P_1	0
P_2	1
P_3	2
P_4	3

Turnaround Time			
P_1	P_2	P_3	P_4

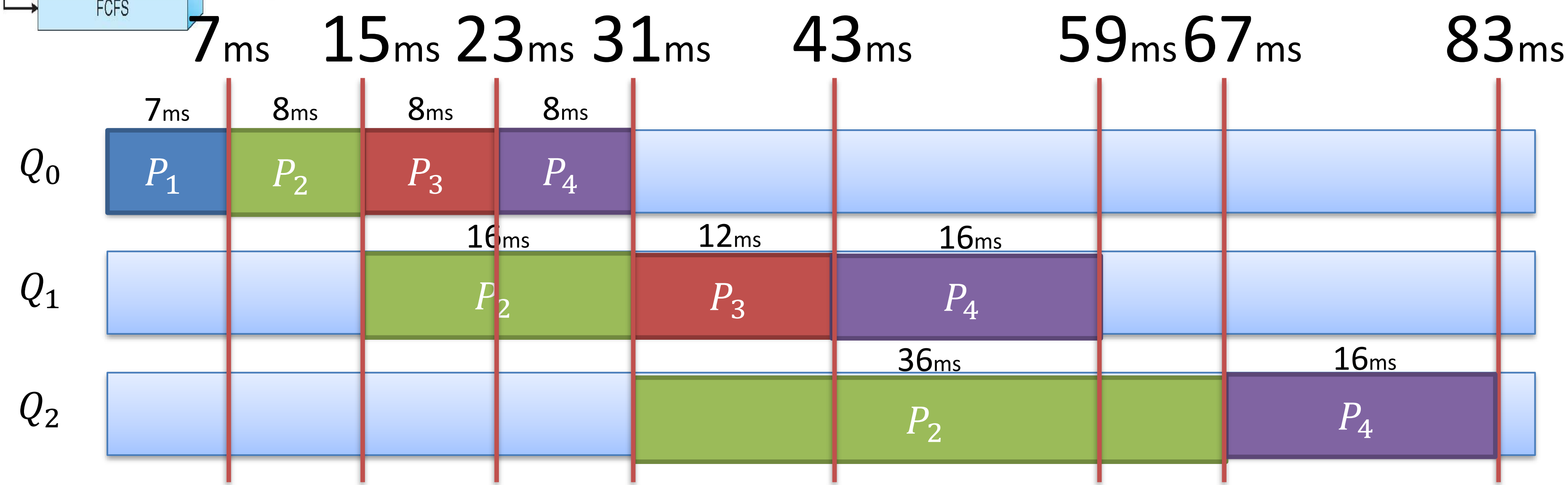


5. Multilevel Queue Scheduling

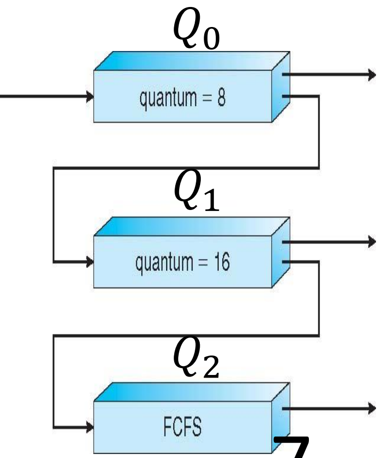


Process	***Arrival*** **Time**
P_1	0
P_2	1
P_3	2
P_4	3

Turnaround Time			
P_1	P_2	P_3	P_4
$7 - 0$			
$=$ **7**			

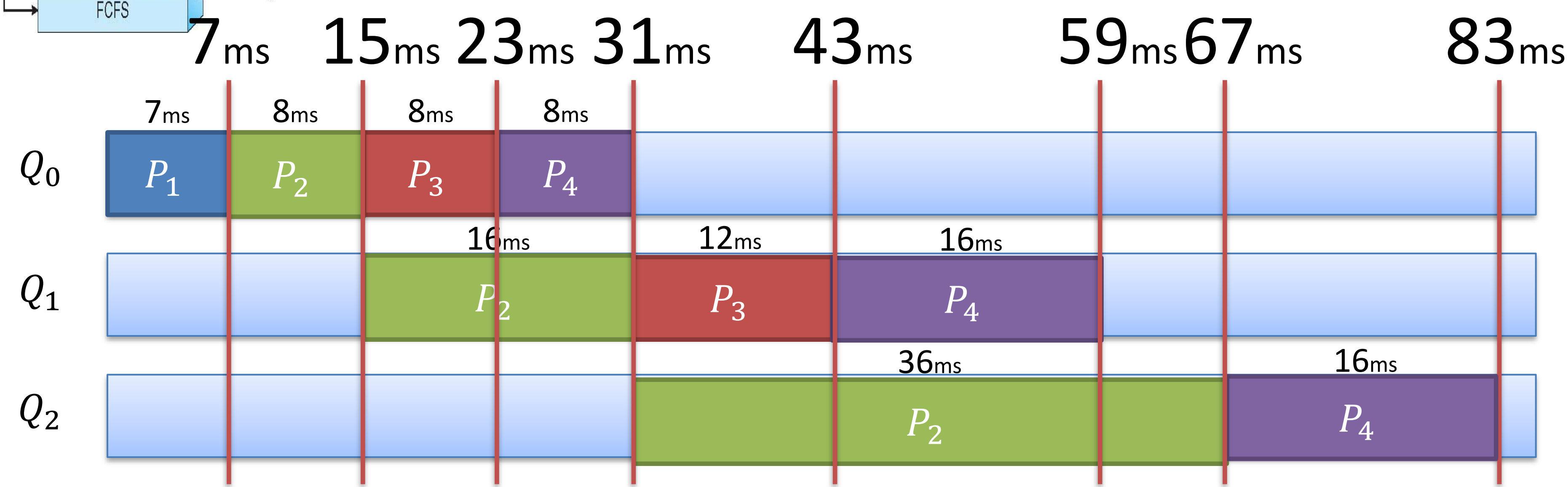


5. Multilevel Queue Scheduling

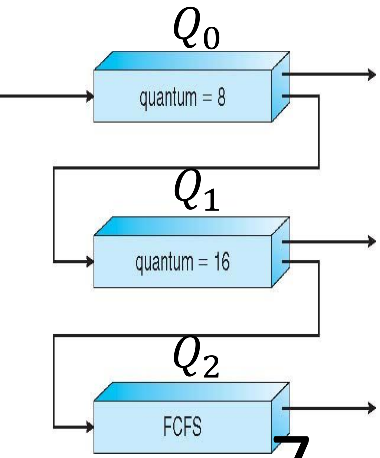


Process	***Arrival*** **Time**
P_1	0
P_2	1
P_3	2
P_4	3

Turnaround Time			
P_1	P_2	P_3	P_4
$7-0$	$67-1$		
$= \mathbf{7}$	$= \mathbf{66}$		

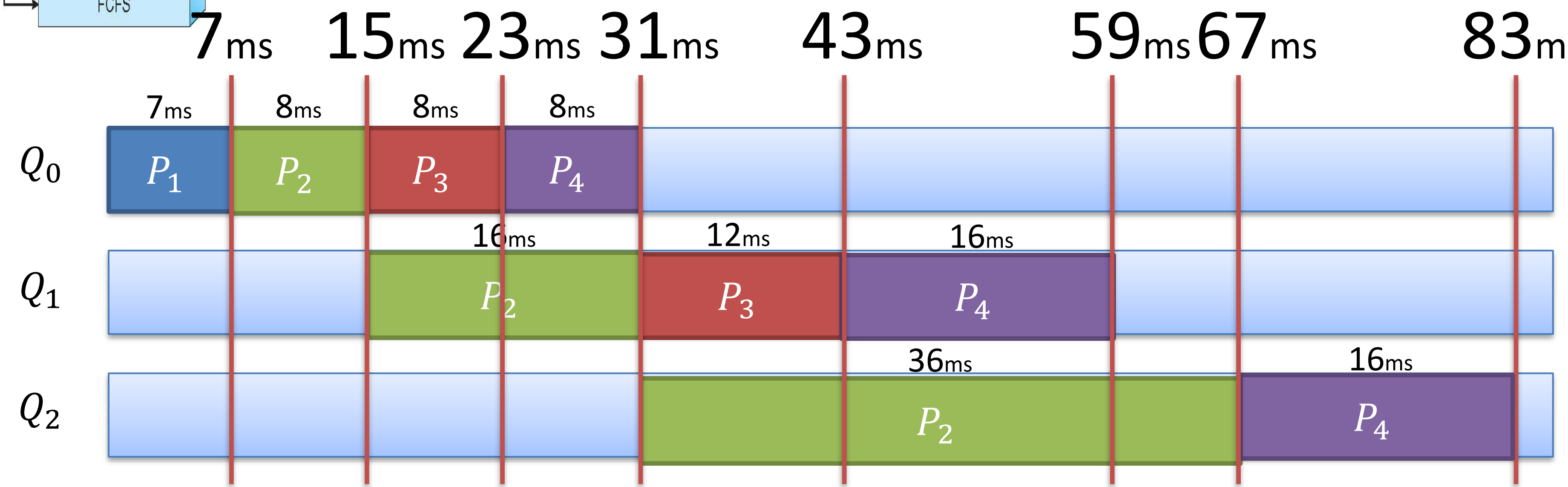


5. Multilevel Queue Scheduling

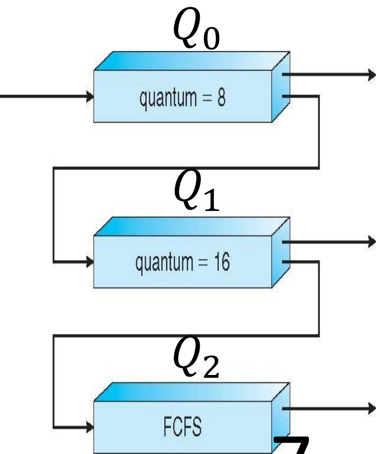


Process	*Arrival* Time
P_1	0
P_2	1
P_3	2
P_4	3

Turnaround Time			
P_1	P_2	P_3	P_4
$7-0$	$67-1$	$43-2$	
$=7$	$=66$	$=41$	

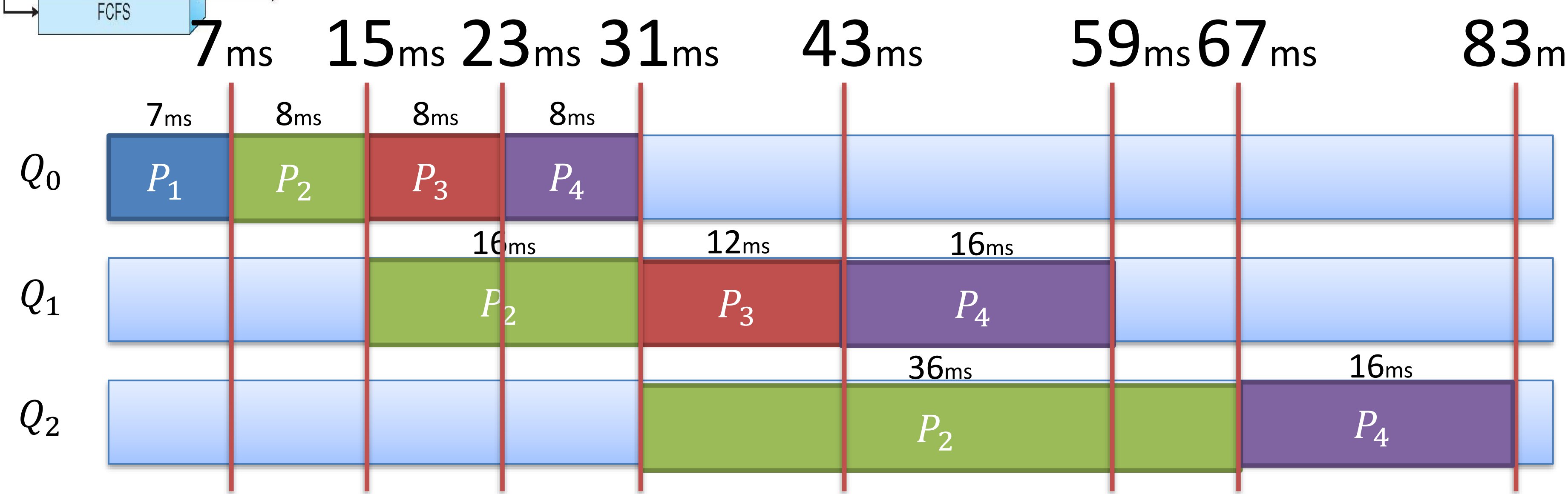


5. Multilevel Queue Scheduling

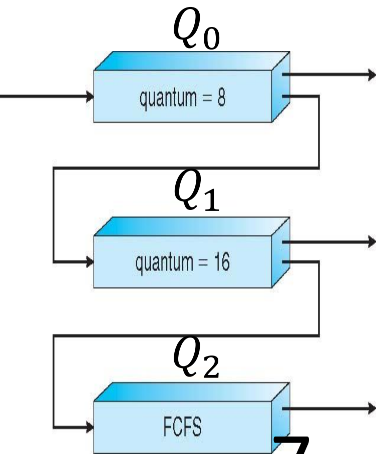


Process	*Arrival* Time
P_1	0
P_2	1
P_3	2
P_4	3

Turnaround Time			
P_1	P_2	P_3	P_4
$7-0$	$67-1$	$43-2$	$83-3$
$=7$	$=66$	$=41$	$=80$

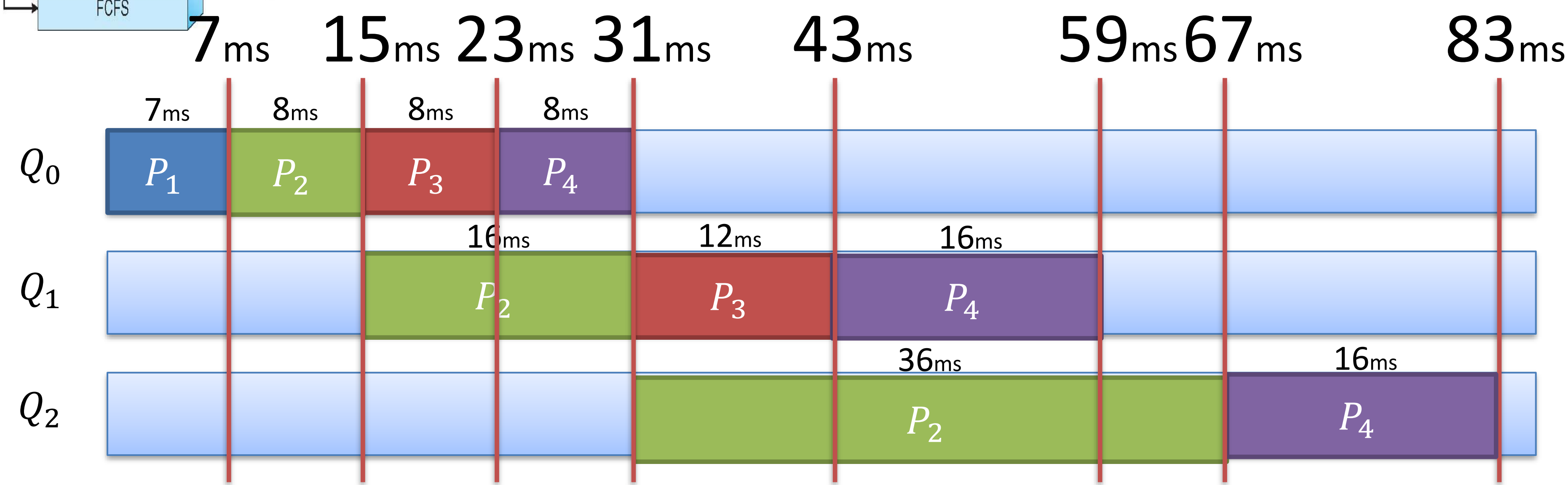


5. Multilevel Queue Scheduling

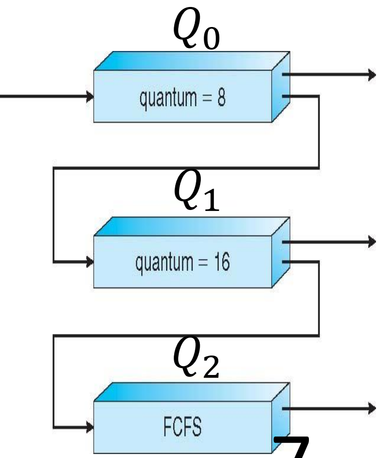


Process	*Arrival* Time
P_1	0
P_2	1
P_3	2
P_4	3

Response Time			
P_1	P_2	P_3	P_4

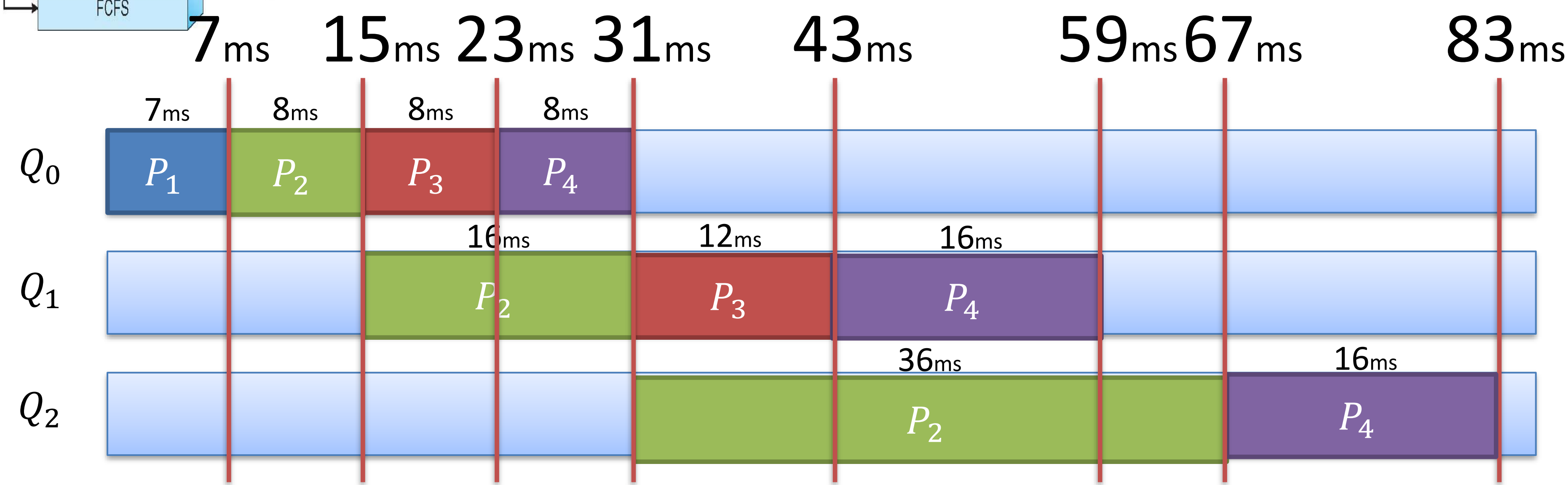


5. Multilevel Queue Scheduling

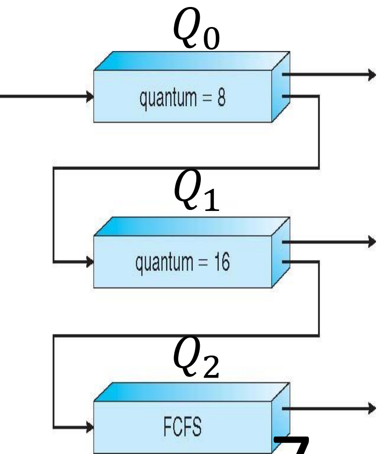


Process	***Arrival*** **Time**
P_1	0
P_2	1
P_3	2
P_4	3

Response Time			
P_1	P_2	P_3	P_4
0 – 0			
= **0**			

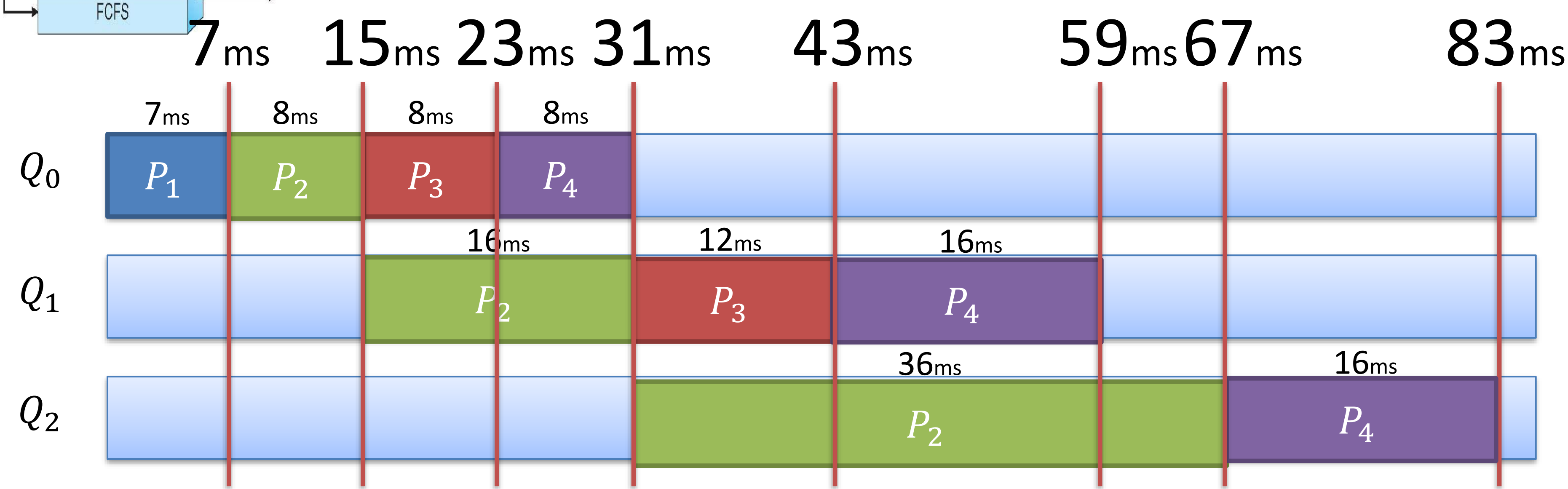


5. Multilevel Queue Scheduling



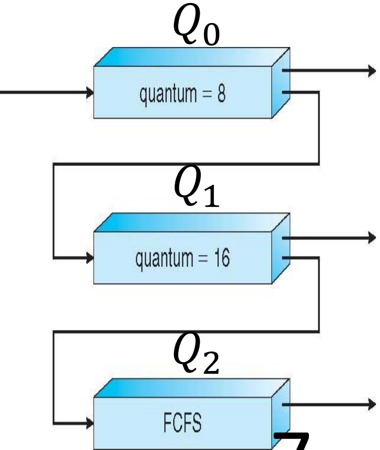
Process	*Arrival* Time
P_1	0
P_2	1
P_3	2
P_4	3

Response Time			
P_1	P_2	P_3	P_4
$0-0$	$7-1$		
$=0$	$=6$		



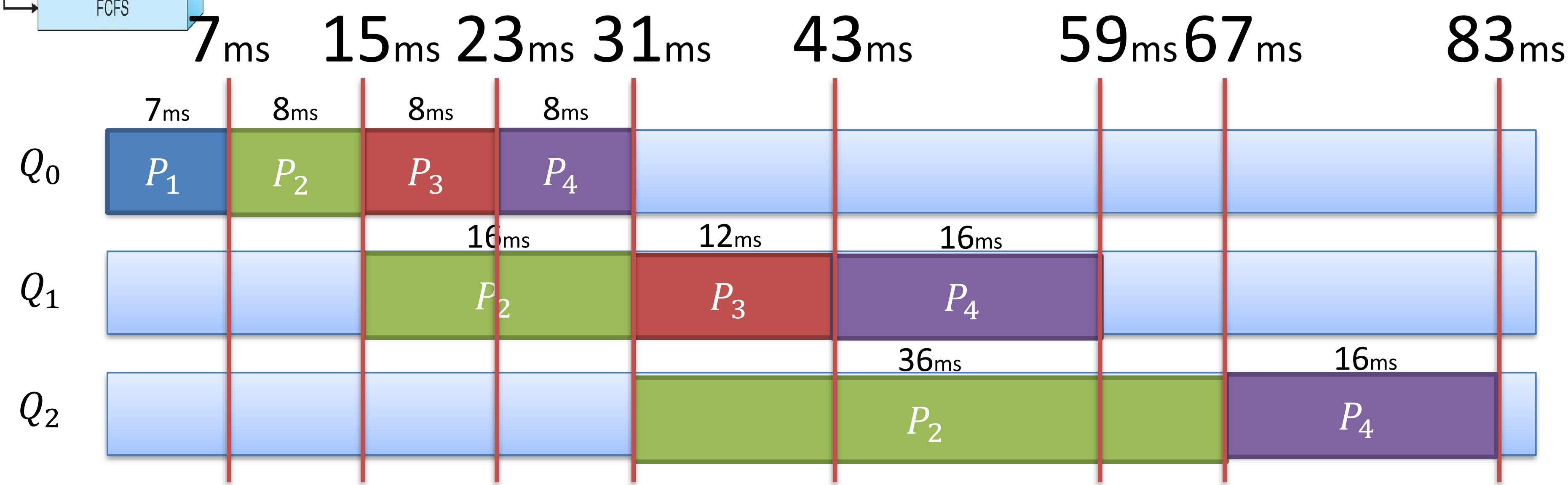
Scheduling Algorithms (24/32)

5. Multilevel Queue Scheduling

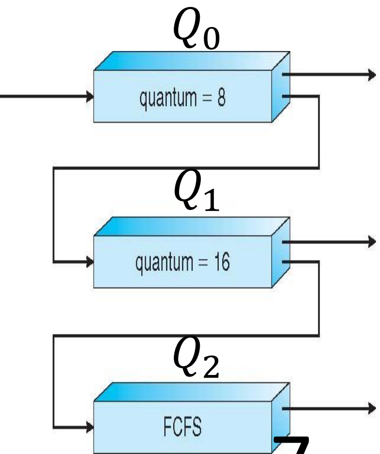


Process	*Arrival* Time
P_1	0
P_2	1
P_3	2
P_4	3

Response Time			
P_1	P_2	P_3	P_4
0 – 0	7 – 1	15 – 2	
= 0	= 6	= 13	

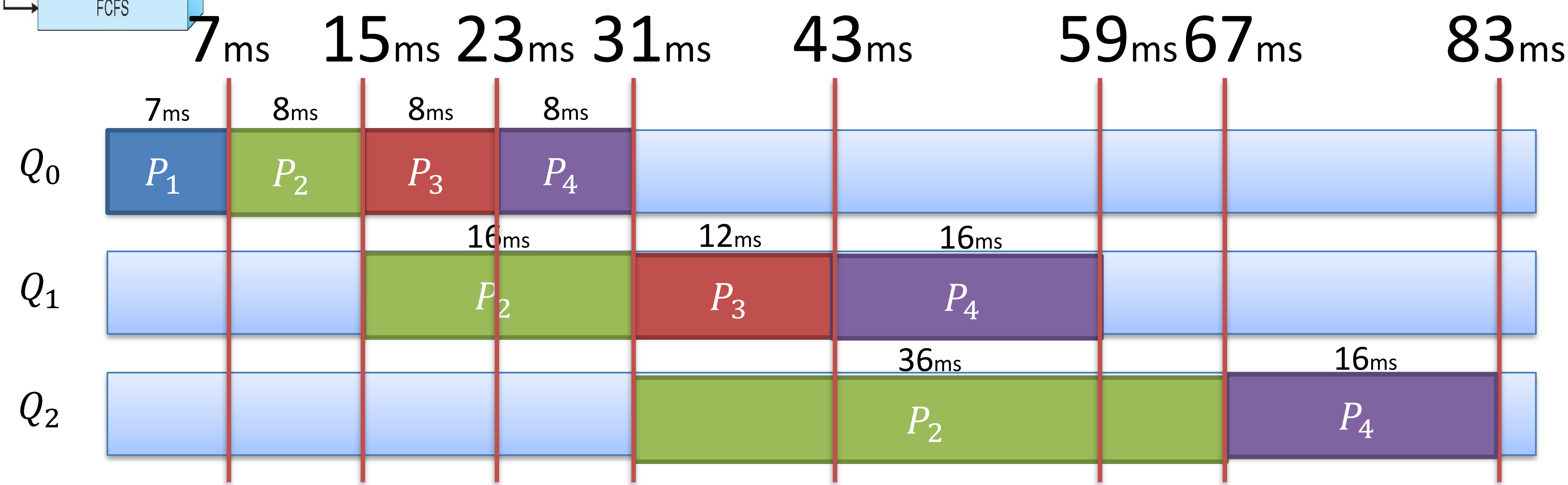


5. Multilevel Queue Scheduling



Process	***Arrival*** **Time**
P_1	0
P_2	1
P_3	2
P_4	3

Response Time			
$\boldsymbol{P_1}$	$\boldsymbol{P_2}$	$\boldsymbol{P_3}$	$\boldsymbol{P_4}$
$0-0$	$7-1$	$15-2$	$23-3$
$\mathbf{=0}$	$\mathbf{=6}$	$\mathbf{=13}$	$\mathbf{=20}$

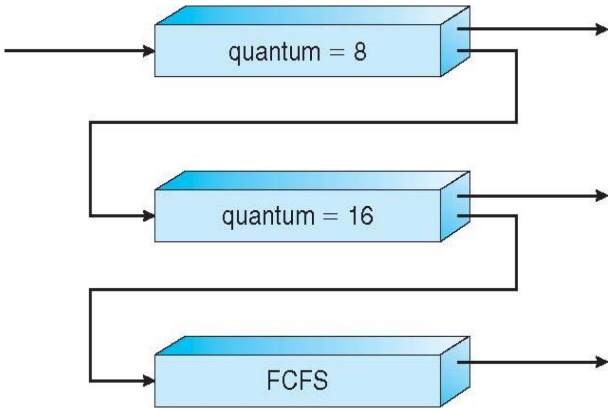


5. Multilevel Queue Scheduling

- Now we add the concepts of varying arrival times and preemption to the analysis

Process	***Arrival*** **Time**	Burst Time
P_1	0	7
P_2	1	60
P_3	2	20
P_4	3	40

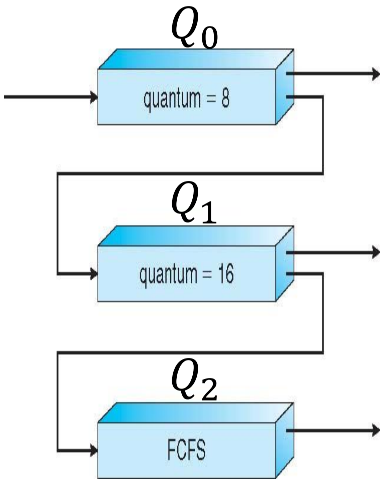
Using single-core processor



Multilevel Queue Fixed priority
non-preemptive

5. Multilevel Queue Scheduling

Process	***Arrival*** **Time**	Burst Time
P_1	0	7
P_2	1	60
P_3	2	20
P_4	3	40

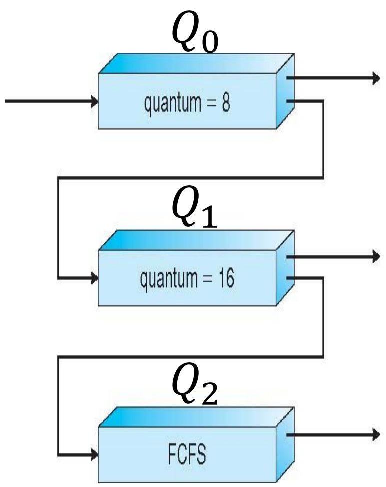


Q_0

Q_1

Q_2

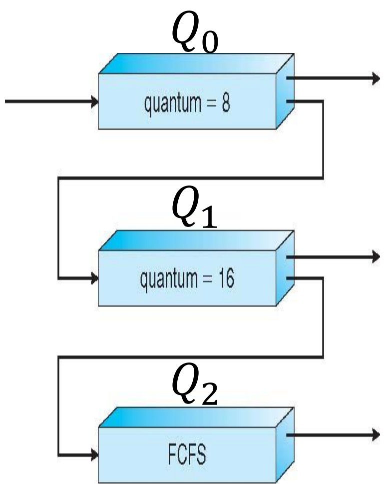
5. Multilevel Queue Scheduling



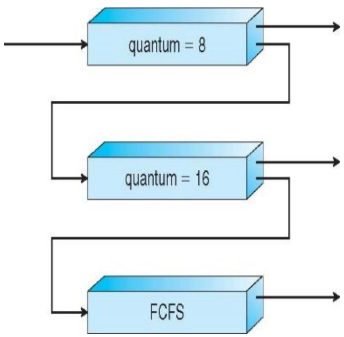
Process	***Arrival* Time**	Burst Time
P_1	0	~~7~~
P_2	1	60
P_3	2	20
P_4	3	40

5. Multilevel Queue Scheduling

Process	***Arrival*** **Time**	Burst Time
$\cancel{P_1}$	0	$\cancel{7}$
P_2	1	60
P_3	2	20
P_4	3	40

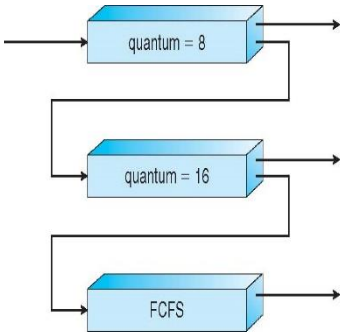


5. Multilevel Queue Scheduling



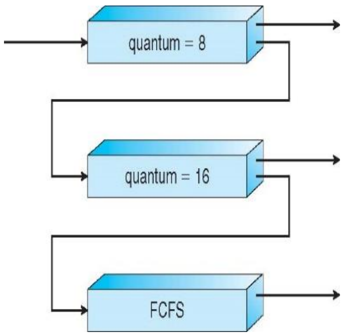
Process	***Arrival* Time**	Burst Time
~~P_1~~	0	~~7~~
P_2	1	~~60~~ 52
P_3	2	20
P_4	3	40

5. Multilevel Queue Scheduling

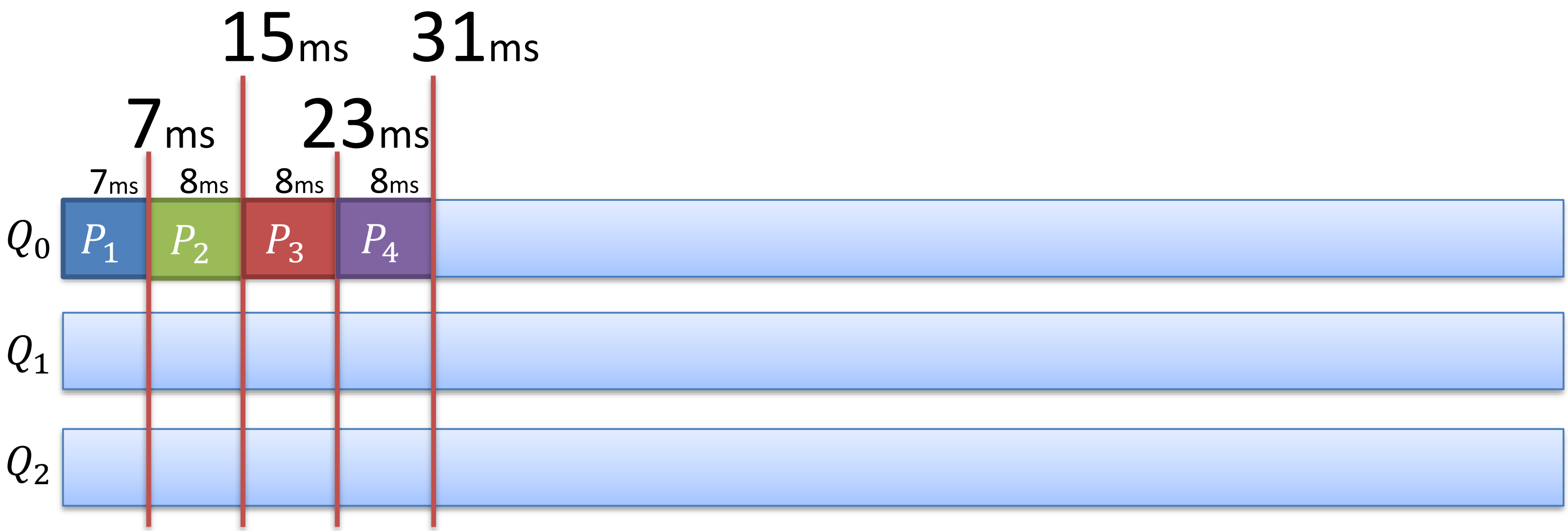


Process	*Arrival* Time	Burst Time
~~P_1~~	0	~~7~~
P_2	1	~~60~~ 52
P_3	2	~~20~~ 12
P_4	3	40

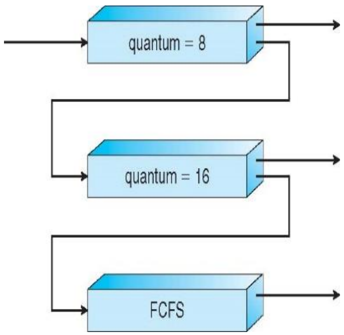
5. Multilevel Queue Scheduling



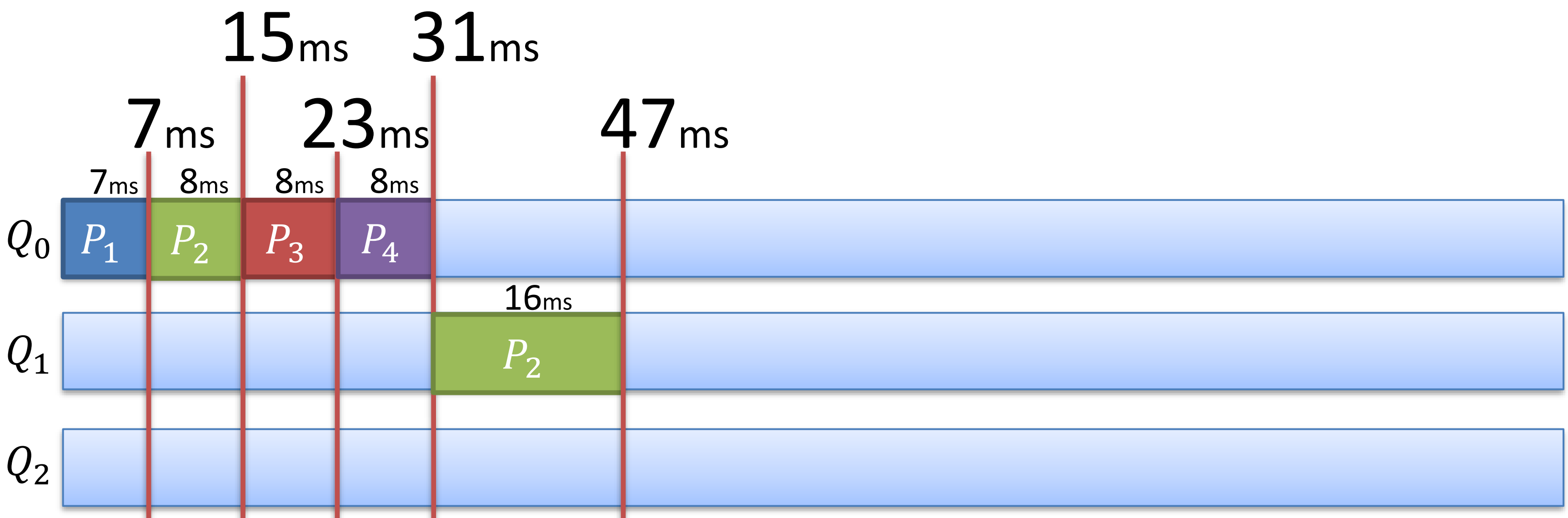
Process	***Arrival* Time**	Burst Time
$\cancel{P_1}$	0	$\cancel{7}$
P_2	1	$\cancel{60}$ 52
P_3	2	$\cancel{20}$ 12
P_4	3	$\cancel{40}$ 32



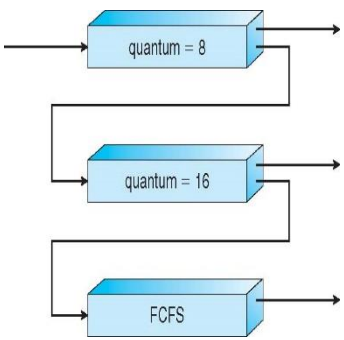
5. Multilevel Queue Scheduling



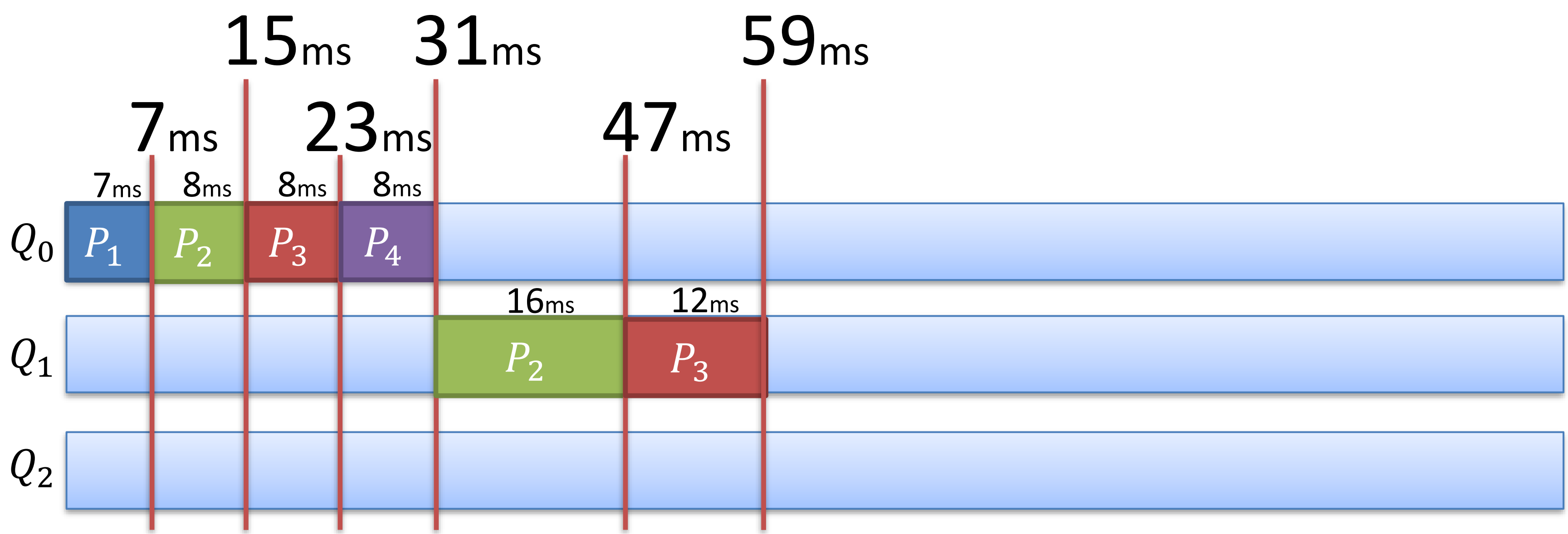
Process	***Arrival* Time**	Burst Time
$\cancel{P_1}$	0	$\cancel{7}$
P_2	1	$\cancel{60}$ $\cancel{52}$ 36
P_3	2	$\cancel{20}$ 12
P_4	3	$\cancel{40}$ 32



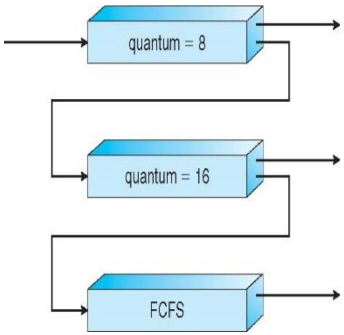
5. Multilevel Queue Scheduling



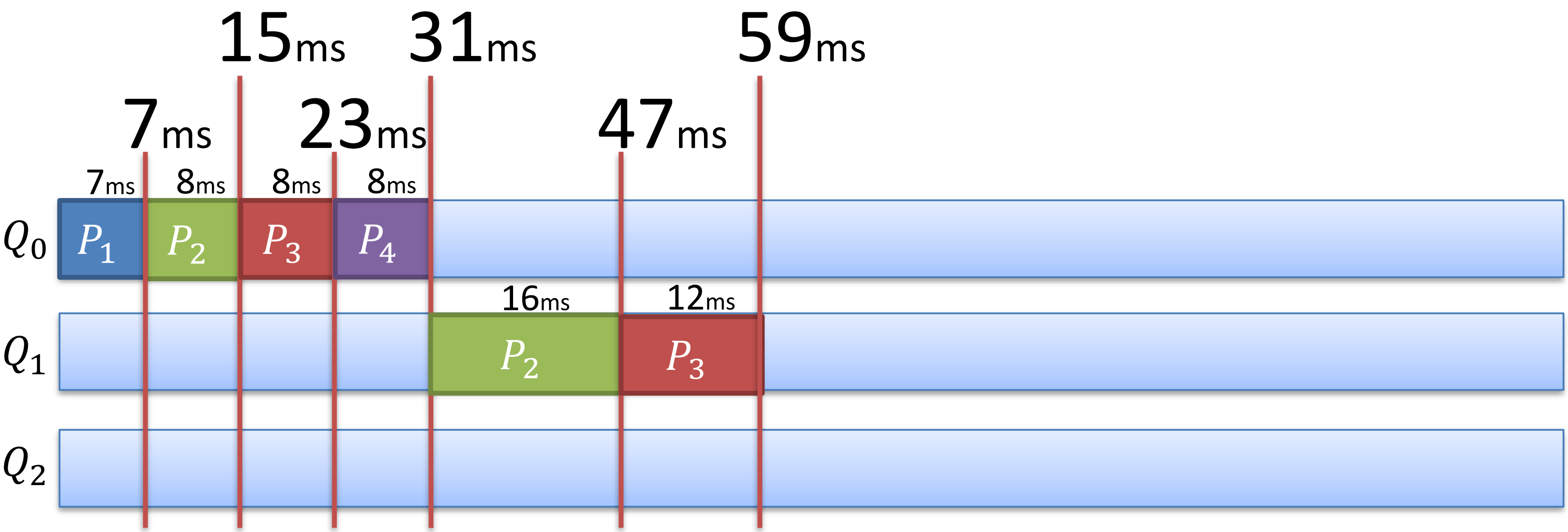
Process	***Arrival*** **Time**	Burst Time
~~P_1~~	0	~~7~~
P_2	1	~~60~~ ~~52~~ 36
P_3	2	~~20~~ ~~12~~
P_4	3	~~40~~ 32



5. Multilevel Queue Scheduling

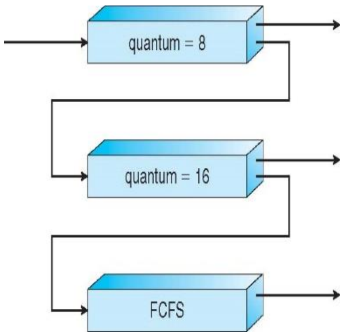


Process	*Arrival* Time	Burst Time
~~P_1~~	0	~~7~~
P_2	1	~~60~~ ~~52~~ 36
~~P_3~~	2	~~20~~ ~~12~~
P_4	3	~~40~~ 32

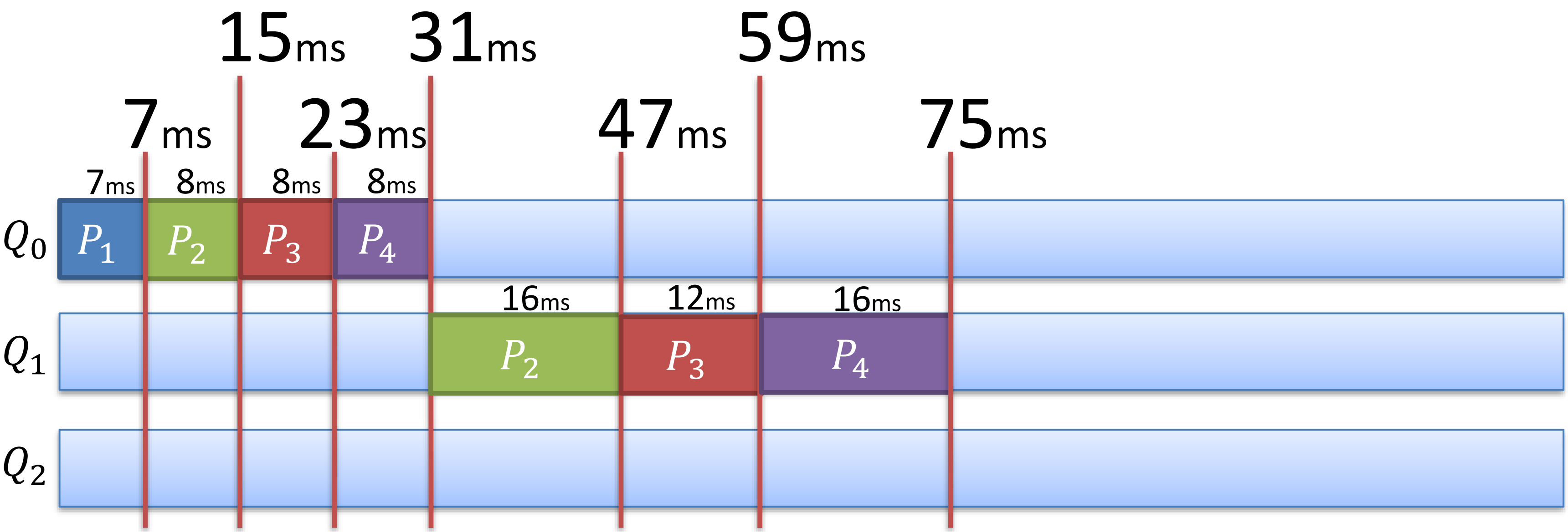


Scheduling Algorithms (27/32)

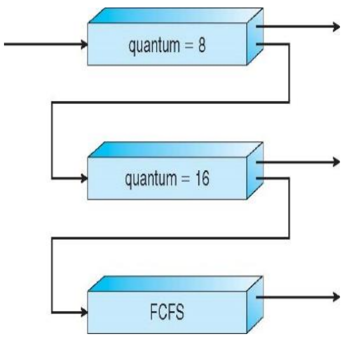
5. Multilevel Queue Scheduling



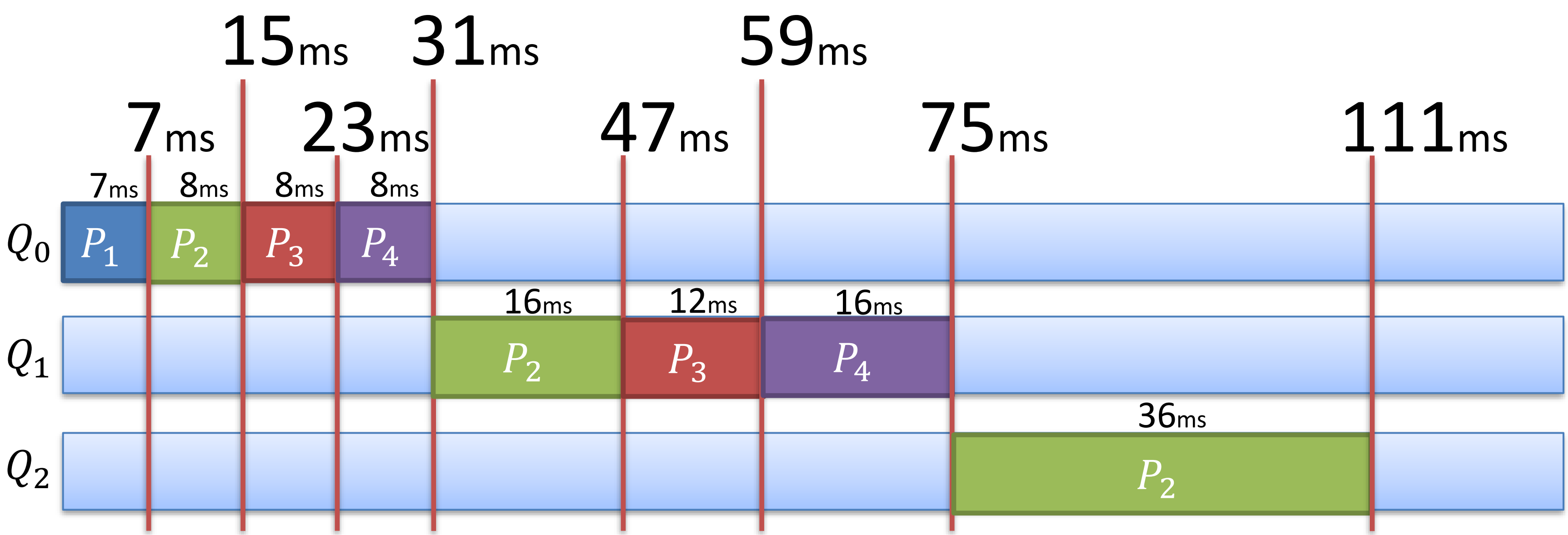
Process	***Arrival*** **Time**	Burst Time
~~P_1~~	0	~~7~~
P_2	1	~~60~~ ~~52~~ 36
~~P_3~~	2	~~20~~ ~~12~~
P_4	3	~~40~~ ~~32~~ 16



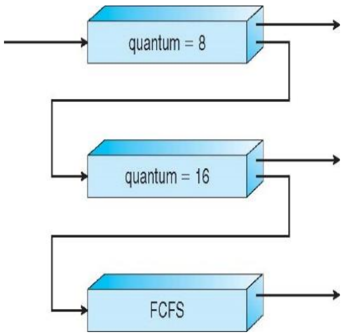
5. Multilevel Queue Scheduling



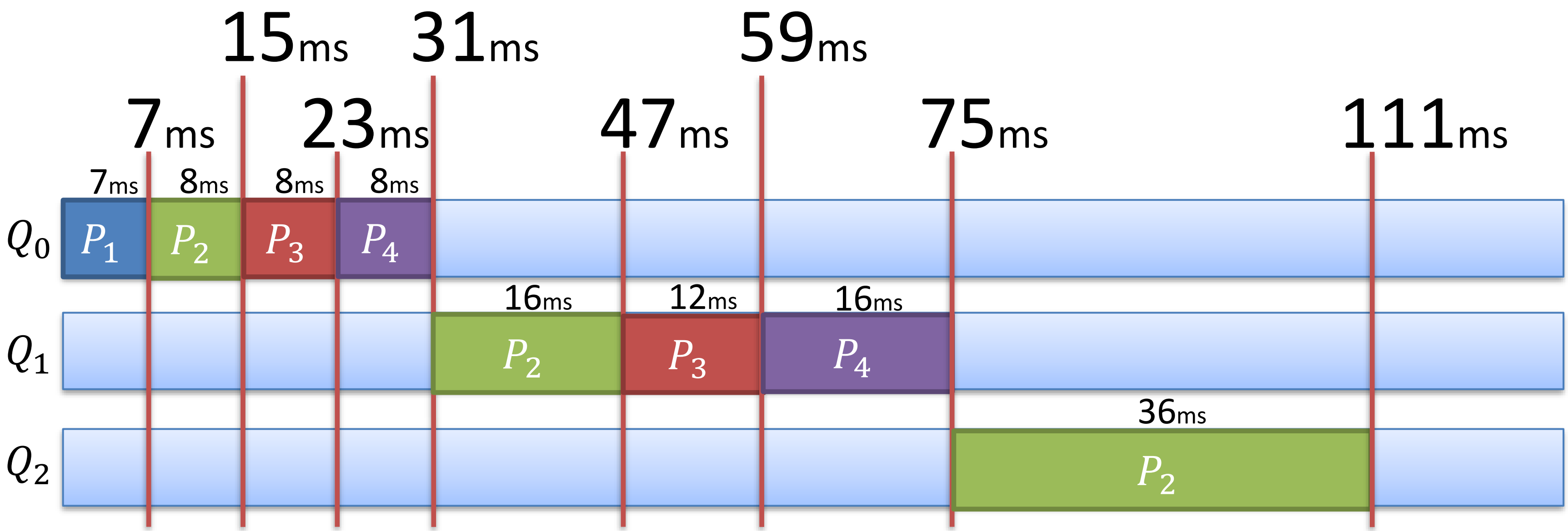
Process	*Arrival* Time	Burst Time
$\cancel{P_1}$	0	$\cancel{7}$
P_2	1	$\cancel{60}$ $\cancel{52}$ $\cancel{36}$
$\cancel{P_3}$	2	$\cancel{20}$ $\cancel{12}$
P_4	3	$\cancel{40}$ $\cancel{32}$ 16



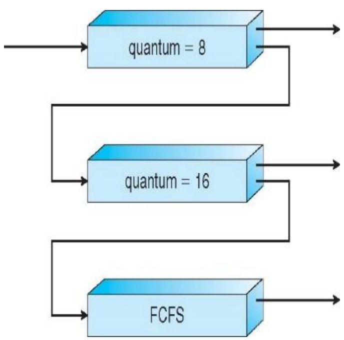
5. Multilevel Queue Scheduling



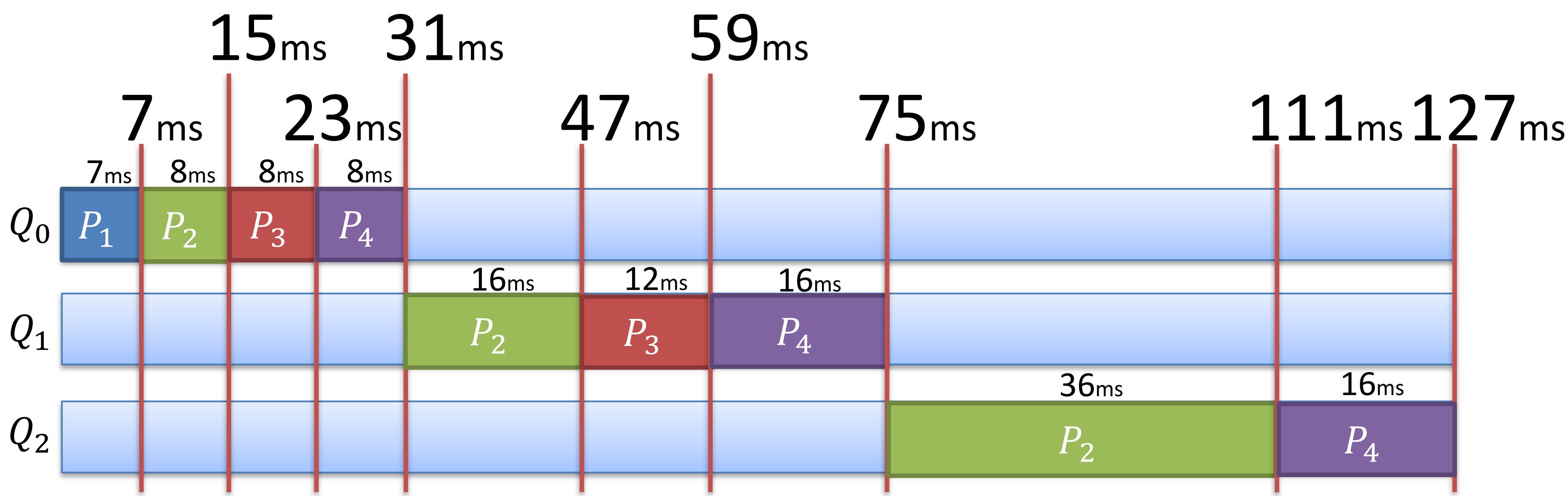
Process	*Arrival* Time	Burst Time
~~P_1~~	0	~~7~~
~~P_2~~	1	~~60~~ ~~52~~ ~~36~~
~~P_3~~	2	~~20~~ ~~12~~
P_4	3	~~40~~ ~~32~~ 16



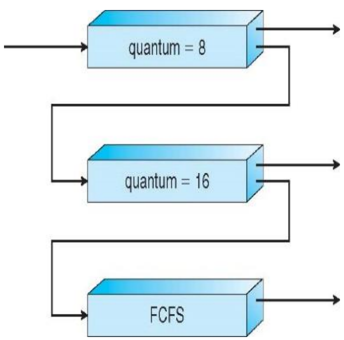
5. Multilevel Queue Scheduling



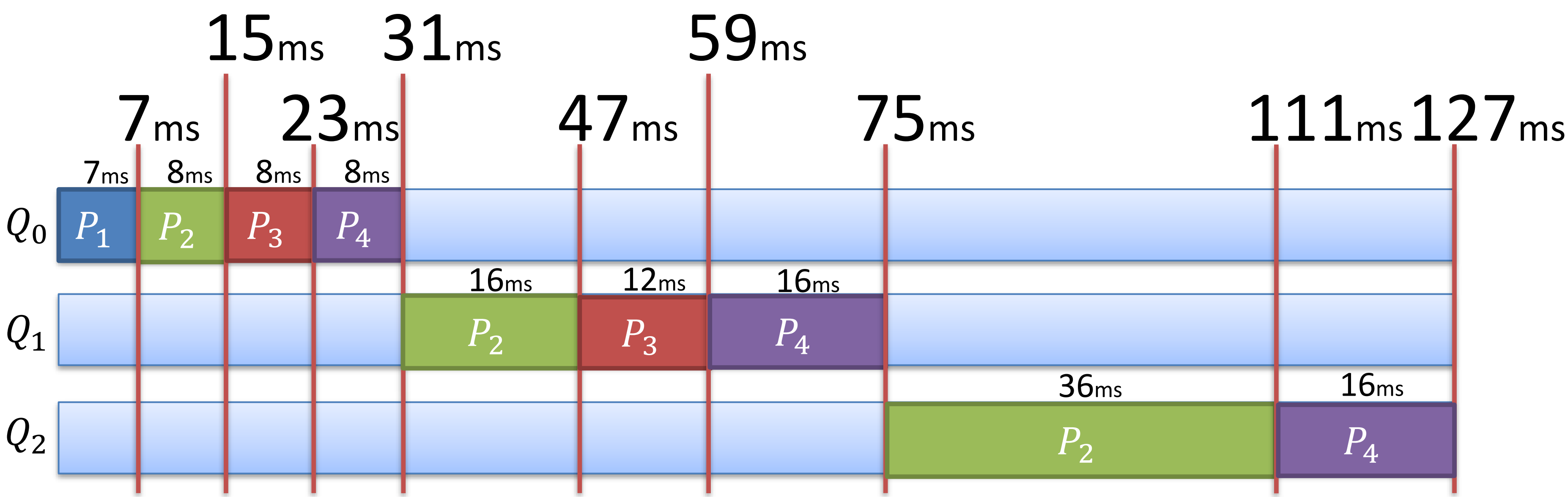
Process	***Arrival*** **Time**	Burst Time
~~P_1~~	0	~~7~~
~~P_2~~	1	~~60~~ ~~52~~ ~~36~~
~~P_3~~	2	~~20~~ ~~12~~
P_4	3	~~40~~ ~~32~~ ~~16~~



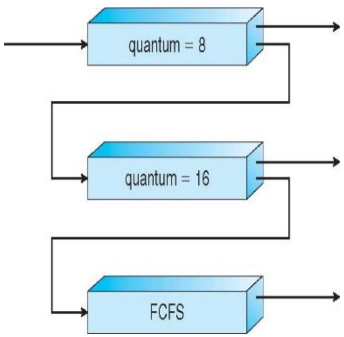
5. Multilevel Queue Scheduling



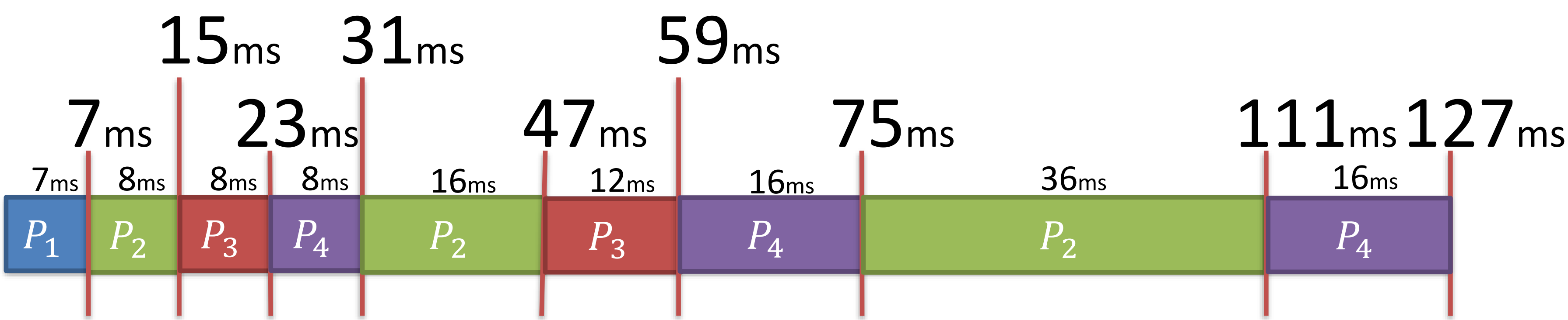
Process	***Arrival*** **Time**	Burst Time
~~P_1~~	0	~~7~~
~~P_2~~	1	~~60~~ ~~52~~ ~~36~~
~~P_3~~	2	~~20~~ ~~12~~
~~P_4~~	3	~~40~~ ~~32~~ ~~16~~



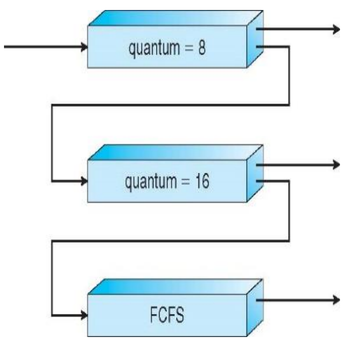
5. Multilevel Queue Scheduling



Process	***Arrival* Time**	Burst Time
$\cancel{P_1}$	0	~~7~~
$\cancel{P_2}$	1	~~60~~ ~~52~~ ~~36~~
$\cancel{P_3}$	2	~~20~~ ~~12~~
$\cancel{P_4}$	3	~~40~~ ~~32~~ ~~16~~

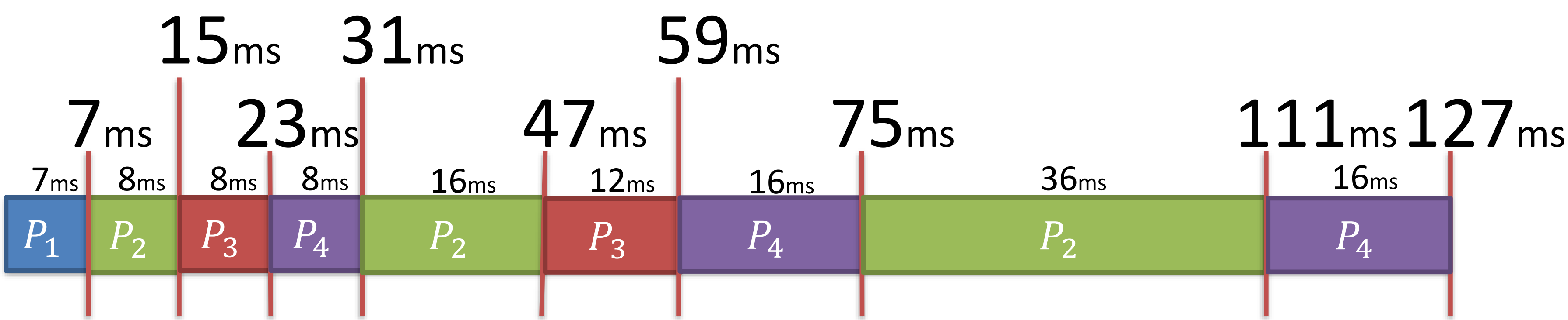


5. Multilevel Queue Scheduling



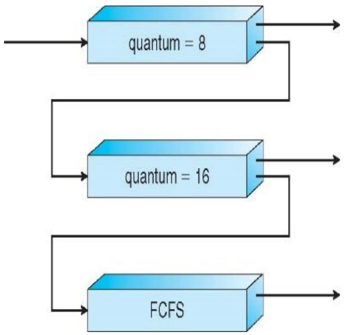
Process	*Arrival* Time
P_1	0
P_2	1
P_3	2
P_4	3

Waiting Time			
P_1	P_2	P_3	P_4
0	6 + 16 + 28	13 + 24	20 + 28 + 36
= 0	= 50	= 37	= 84



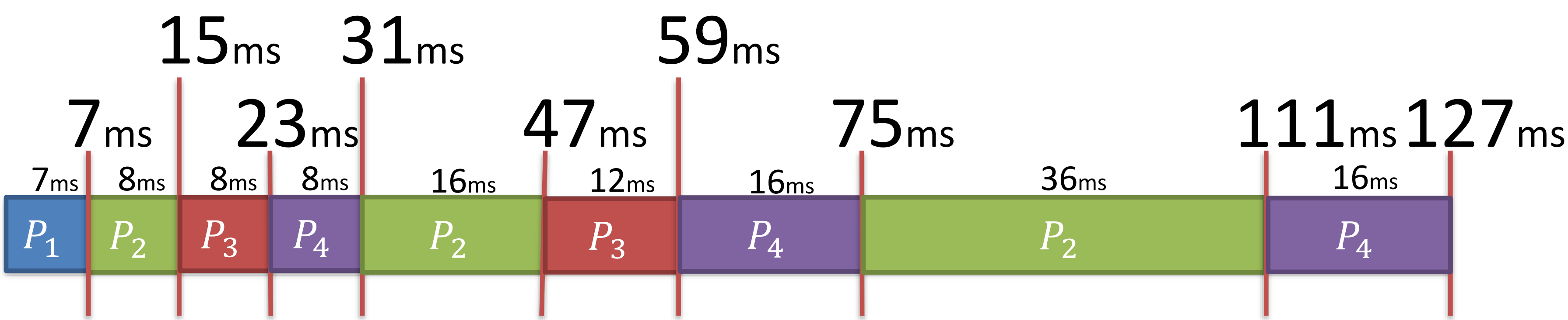
Scheduling Algorithms (30/32)

5. Multilevel Queue Scheduling

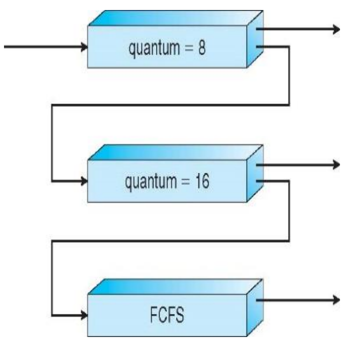


Process	*Arrival* Time
P_1	0
P_2	1
P_3	2
P_4	3

Turnaround Time			
$\boldsymbol{P_1}$	$\boldsymbol{P_2}$	$\boldsymbol{P_3}$	$\boldsymbol{P_4}$
$7-0$	$111-1$	$59-2$	$127-3$
$\boldsymbol{=7}$	$\boldsymbol{=110}$	$\boldsymbol{=57}$	$\boldsymbol{=124}$

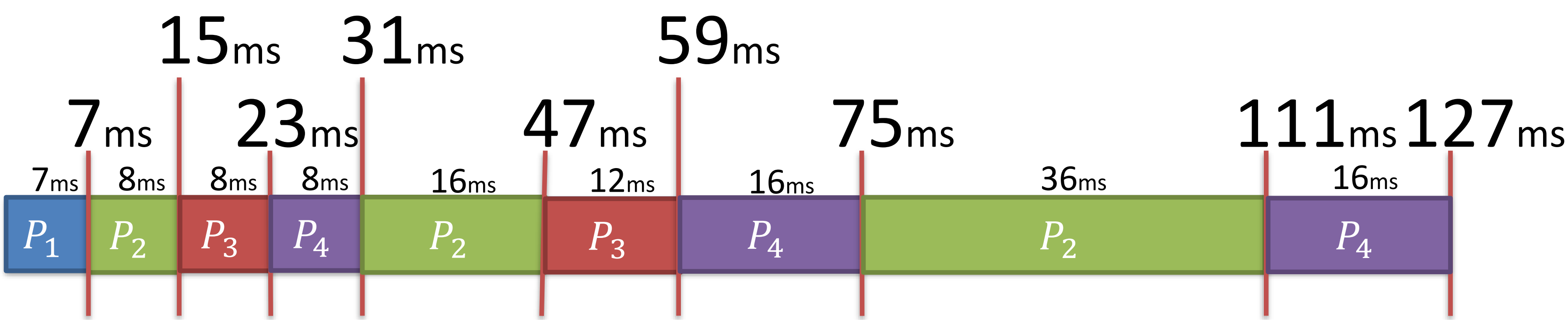


5. Multilevel Queue Scheduling



Process	***Arrival*** **Time**
P_1	0
P_2	1
P_3	2
P_4	3

Response Time			
$\boldsymbol{P_1}$	$\boldsymbol{P_2}$	$\boldsymbol{P_3}$	$\boldsymbol{P_4}$
$0 - 0$	$7 - 1$	$15 - 2$	$23 - 3$
$\mathbf{= 0}$	$\mathbf{= 6}$	$\mathbf{= 13}$	$\mathbf{= 20}$



Scheduling Algorithms (32/32)

Student Example:

- Using a Multilevel Queue Scheduling algorithm in Fig.1. Consider the following processes with the relative CPU bursts.

Process	***Arrival*** **Time**	Burst Time
P_1	0	20
P_2	1	60
P_3	2	5
P_4	2	40

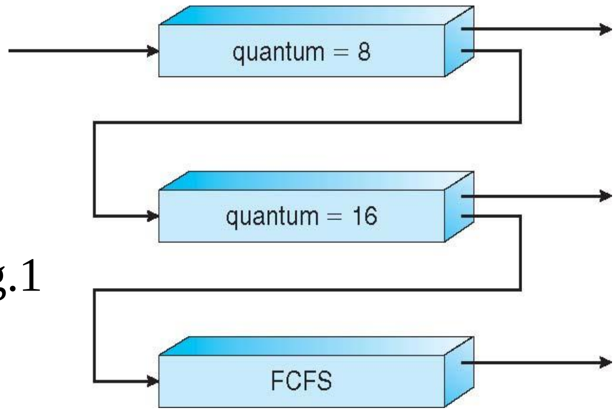


Fig.1

Multilevel Queue Fixed priority
non-preemptive

Thank You

Dr. Ahmed Hagag

ahagag@fci.bu.edu.eg

https://www.youtube.com/channel/UCzYgAyyZTLfnLFjQexOKxbQ

https://www.facebook.com/ahmed.hagag.71/