Priority continued and Preemptive scheduling

Dr. Naser Al Madi

Learning objectives

- Priority scheduling (non-preemptive)
- Preemptive scheduling:
  - Round Robin
  - Shortest Remaining Time
  - Preemptive Priority

Late project policy

- You get 1 week after the due date to submit your project without extensions (26 points max).
- Once in the semester, you get to submit a project by the following Monday WITH extensions (30 points max). An email by the project due date is necessary to use this option.
- That said, getting behind is not a good idea.

Priority Scheduling

Priority Scheduling

- Associate a priority with each process
- Run the process with the highest priority
- **Externally** defined:
  - ex: based on importance
  - employee's processes given higher preference than visitor's

- **Internally** defined, based on memory requirements, file requirements, CPU requirements vs. I/O requirements, etc.

Priority Scheduling

- SJF is priority scheduling, where priority is inversely proportional to length of next CPU burst

Priority Scheduling

- Priority can be represented in two ways:
- Correlating:
    - For example, on a scale from 1 to 500 with 1 is lowest priority and 500 is highest priority

$$^{L}[1 - 500]^{H}$$

- Inverse:
    - For example, on a scale from 1 to 500 with 1 is highest priority and 500 is lowest priority

$$^{H}[1 - 500]^{L}$$

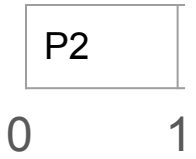
Priority Scheduling

L [1 - 20] H

Process	CPU time	Priority
P0	5	8
P1	3	10
P2	1	15
P3	2	15

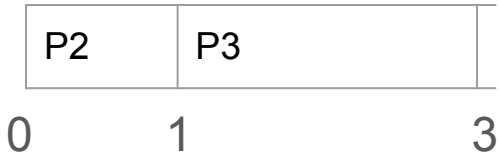
Priority Scheduling

Process	CPU time	Priority
P0	5	8
P1	3	10
P2	1	15
P3	2	15



Priority Scheduling

Process	CPU time	Priority
P0	5	8
P1	3	10
P2	1	15
P3	2	15



Priority Scheduling

Process	CPU time	Priority
P0	5	8
P1	3	10
P2	1	15
P3	2	15

Priority Scheduling

Process	CPU time	Priority
P0	5	8
P1	3	10
P2	1	15
P3	2	15

Priority Scheduling

Process	CPU time	Priority
P0	5	8
P1	3	10
P2	1	15
P3	2	15

	Wait-time	Turnaround-time
P0	start-arrival	
P1		
P2		
P3		

Priority Scheduling

Process	CPU time	Priority
P0	5	8
P1	3	10
P2	1	15
P3	2	15

	Wait-time	Turnaround-time
P0	6 - 0 = 6	
P1		
P2		
P3		

Priority Scheduling

Process	CPU time	Priority
P0	5	8
P1	3	10
P2	1	15
P3	2	15

	Wait-time	Turnaround-time
P0	6 - 0 = 6	
P1	3 - 0 = 3	
P2		
P3		

Priority Scheduling

Process	CPU time	Priority
P0	5	8
P1	3	10
P2	1	15
P3	2	15

	Wait-time	Turnaround-time
P0	6 - 0 = 6	
P1	3 - 0 = 3	
P2	0 - 0 = 0	
P3		

Priority Scheduling

Process	CPU time	Priority
P0	5	8
P1	3	10
P2	1	15
P3	2	15

	Wait-time	Turnaround-time
P0	6 - 0 = 6	
P1	3 - 0 = 3	
P2	0 - 0 = 0	
P3	1 - 0 = 1	

Priority Scheduling

Process	CPU time	Priority
P0	5	8
P1	3	10
P2	1	15
P3	2	15

	Wait-time	Turnaround-time
P0	6 - 0 = 6	Wait + CPU_time
P1	3 - 0 = 3	
P2	0 - 0 = 0	
P3	1 - 0 = 1	

Priority Scheduling

Process	CPU time	Priority
P0	5	8
P1	3	10
P2	1	15
P3	2	15

	Wait-time	Turnaround-time
P0	6 - 0 = 6	6 + 5 =11
P1	3 - 0 = 3	
P2	0 - 0 = 0	
P3	1 - 0 = 1	

Priority Scheduling

Process	CPU time	Priority
P0	5	8
P1	3	10
P2	1	15
P3	2	15

	Wait-time	Turnaround-time
P0	6 - 0 = 6	6 + 5 =11
P1	3 - 0 = 3	3 + 3 = 6
P2	0 - 0 = 0	
P3	1 - 0 = 1	

Priority Scheduling

Process	CPU time	Priority
P0	5	8
P1	3	10
P2	1	15
P3	2	15

	Wait-time	Turnaround-time
P0	6 - 0 = 6	6 + 5 =11
P1	3 - 0 = 3	3 + 3 = 6
P2	0 - 0 = 0	0 + 1 = 1
P3	1 - 0 = 1	

Priority Scheduling

Process	CPU time	Priority
P0	5	8
P1	3	10
P2	1	15
P3	2	15

	Wait-time	Turnaround-time
P0	6 - 0 = 6	6 + 5 =11
P1	3 - 0 = 3	3 + 3 = 6
P2	0 - 0 = 0	0 + 1 = 1
P3	1 - 0 = 1	1 + 2 = 3

Priority Scheduling Evaluation

- Non-preemptive

	Wait-time	Turnaround-time
P0	6 - 0 = 6	6 + 5 =11
P1	3 - 0 = 3	3 + 3 = 6
P2	0 - 0 = 0	0 + 1 = 1
P3	1 - 0 = 1	1 + 2 = 3
AVG	2.5	5.25

Priority Scheduling Evaluation

	Wait-time	Turnaround-time
P0	6 - 0 = 6	6 + 5 =11
P1	3 - 0 = 3	3 + 3 = 6
P2	0 - 0 = 0	0 + 1 = 1
P3	1 - 0 = 1	1 + 2 = 3
AVG	2.5	5.25

- Non-preemptive
- convoy effect?

Priority Scheduling Evaluation

	Wait-time	Turnaround-time
P0	6 - 0 = 6	6 + 5 =11
P1	3 - 0 = 3	3 + 3 = 6
P2	0 - 0 = 0	0 + 1 = 1
P3	1 - 0 = 1	1 + 2 = 3
AVG	2.5	5.25

- Non-preemptive
- convoy effect – one long-burst process is followed by many short-burst processes, short processes have to wait a long time
- fairness?

Priority Scheduling Evaluation

	Wait-time	Turnaround-time
P0	6 - 0 = 6	6 + 5 =11
P1	3 - 0 = 3	3 + 3 = 6
P2	0 - 0 = 0	0 + 1 = 1
P3	1 - 0 = 1	1 + 2 = 3
AVG	2.5	5.25

- Non-preemptive
- convoy effect – one long-burst process is followed by many short-burst processes, short processes have to wait a long time
- fairness — penalizes low-priority processes
- starvation?

Priority Scheduling Evaluation

	Wait-time	Turnaround-time
P0	6 - 0 = 6	6 + 5 =11
P1	3 - 0 = 3	3 + 3 = 6
P2	0 - 0 = 0	0 + 1 = 1
P3	1 - 0 = 1	1 + 2 = 3
AVG	2.5	5.25

- Non-preemptive
- convoy effect – one long-burst process is followed by many short-burst processes, short processes have to wait a long time
- fairness — penalizes low-priority processes
- starvation — possible
- overhead?

Priority Scheduling Evaluation

	Wait-time	Turnaround-time
P0	6 - 0 = 6	6 + 5 =11
P1	3 - 0 = 3	3 + 3 = 6
P2	0 - 0 = 0	0 + 1 = 1
P3	1 - 0 = 1	1 + 2 = 3
AVG	2.5	5.25

- Non-preemptive
- convoy effect – one long-burst process is followed by many short-burst processes, short processes have to wait a long time
- fairness — penalizes low-priority processes
- starvation — possible
- overhead — minimal

Priority Scheduling

- the previous example is simple.
- a more realistic situation would include arrival time.

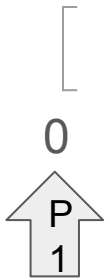
Priority Scheduling (with arrival time)

PID	CPU time	Arrival time	Priority
P0	5	1	8
P1	3	0	10
P2	1	2	15
P3	2	3	15

Can you calculate avg. WT & TT?

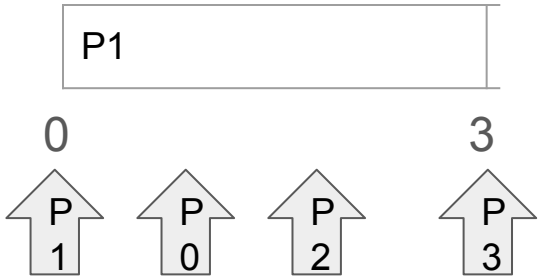
Priority Scheduling (with arrival time)

PID	CPU time	Arrival time	Priority
P0	5	1	8
P1	3	0	10
P2	1	2	15
P3	2	3	15



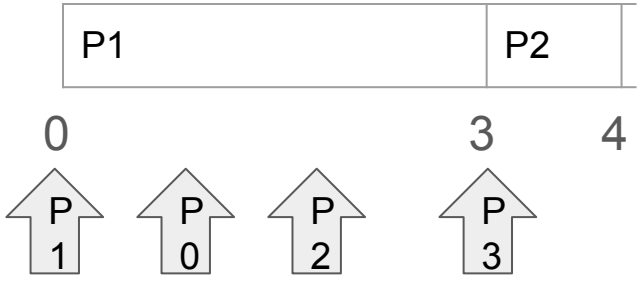
Priority Scheduling (with arrival time)

PID	CPU time	Arrival time	Priority
P0	5	1	8
P1	3	0	10
P2	1	2	15
P3	2	3	15



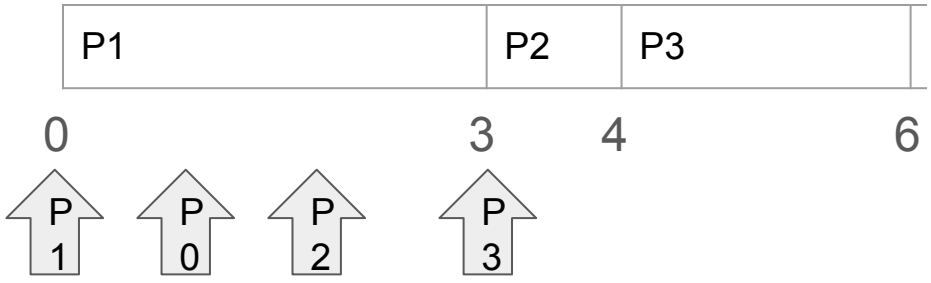
Priority Scheduling (with arrival time)

PID	CPU time	Arrival time	Priority
P0	5	1	8
P1	3	0	10
P2	1	2	15
P3	2	3	15



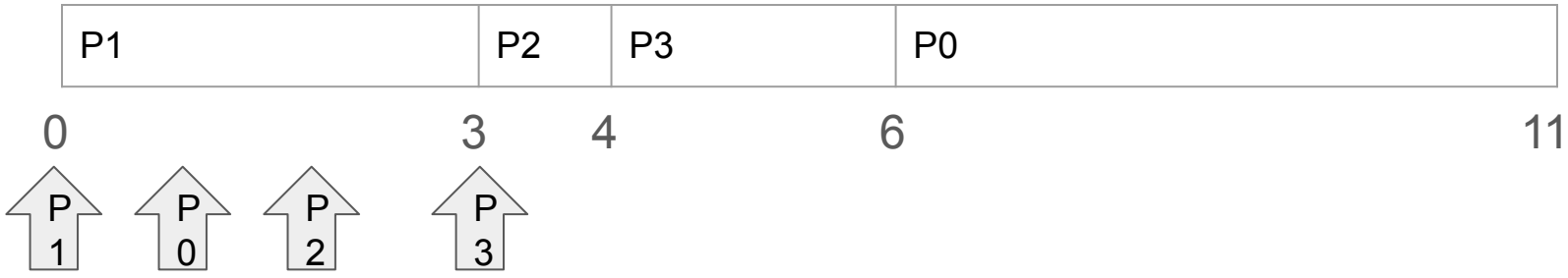
Priority Scheduling (with arrival time)

PID	CPU time	Arrival time	Priority
P0	5	1	8
P1	3	0	10
P2	1	2	15
P3	2	3	15



Priority Scheduling (with arrival time)

PID	CPU time	Arrival time	Priority
P0	5	1	8
P1	3	0	10
P2	1	2	15
P3	2	3	15



Priority Scheduling (with arrival time)

PID	CPU time	Arrival time	Priority
P0	5	1	8
P1	3	0	10
P2	1	2	15
P3	2	3	15

	Wait-time	Turnaround-time
P0	start-arrival	
P1		
P2		
P3		

Priority Scheduling (with arrival time)

PID	CPU time	Arrival time	Priority
P0	5	1	8
P1	3	0	10
P2	1	2	15
P3	2	3	15

	Wait-time	Turnaround-time
P0	6 - 1 = 5	
P1		
P2		
P3		

Priority Scheduling (with arrival time)

PID	CPU time	Arrival time	Priority
P0	5	1	8
P1	3	0	10
P2	1	2	15
P3	2	3	15

	Wait-time	Turnaround-time
P0	6 - 1 = 5	
P1	0 - 0 = 0	
P2		
P3		

Priority Scheduling (with arrival time)

PID	CPU time	Arrival time	Priority
P0	5	1	8
P1	3	0	10
P2	1	2	15
P3	2	3	15

	Wait-time	Turnaround-time
P0	6 - 1 = 5	
P1	0 - 0 = 0	
P2	3 - 2 = 1	
P3		

Priority Scheduling (with arrival time)

PID	CPU time	Arrival time	Priority
P0	5	1	8
P1	3	0	10
P2	1	2	15
P3	2	3	15

	Wait-time	Turnaround-time
P0	6 - 1 = 5	
P1	0 - 0 = 0	
P2	3 - 2 = 1	
P3	4 - 3 = 1	

Priority Scheduling (with arrival time)

PID	CPU time	Arrival time	Priority
P0	5	1	8
P1	3	0	10
P2	1	2	15
P3	2	3	15

	Wait-time	Turnaround-time
P0	6 - 1 = 5	Wait + CPU_time
P1	0 - 0 = 0	
P2	3 - 2 = 1	
P3	4 - 3 = 1	

Priority Scheduling (with arrival time)

PID	CPU time	Arrival time	Priority
P0	5	1	8
P1	3	0	10
P2	1	2	15
P3	2	3	15

	Wait-time	Turnaround-time
P0	6 - 1 = 5	5 + 5 = 10
P1	0 - 0 = 0	
P2	3 - 2 = 1	
P3	4 - 3 = 1	

Priority Scheduling (with arrival time)

PID	CPU time	Arrival time	Priority
P0	5	1	8
P1	3	0	10
P2	1	2	15
P3	2	3	15

	Wait-time	Turnaround-time
P0	6 - 1 = 5	5 + 5 = 10
P1	0 - 0 = 0	0 + 3 = 3
P2	3 - 2 = 1	
P3	4 - 3 = 1	

Priority Scheduling (with arrival time)

PID	CPU time	Arrival time	Priority
P0	5	1	8
P1	3	0	10
P2	1	2	15
P3	2	3	15

	Wait-time	Turnaround-time
P0	6 - 1 = 5	5 + 5 = 10
P1	0 - 0 = 0	0 + 3 = 3
P2	3 - 2 = 1	1 + 1 = 2
P3	4 - 3 = 1	

Priority Scheduling (with arrival time)

PID	CPU time	Arrival time	Priority
P0	5	1	8
P1	3	0	10
P2	1	2	15
P3	2	3	15

	Wait-time	Turnaround-time
P0	6 - 1 = 5	5 + 5 = 10
P1	0 - 0 = 0	0 + 3 = 3
P2	3 - 2 = 1	1 + 1 = 2
P3	4 - 3 = 1	1 + 2 = 3

Priority Scheduling (with arrival time) Evaluation

	Wait-time	Turnaround-time
P0	6 - 1 = 5	5 + 5 = 10
P1	0 - 0 = 0	0 + 3 = 3
P2	3 - 2 = 1	2 + 1 = 3
P3	4 - 3 = 1	1 + 2 = 3
AVG	1.75	4.75

- The most crucial point is giving each process the right priority.

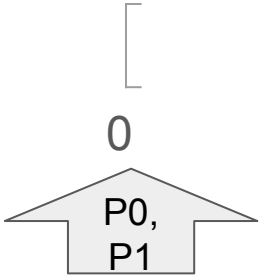
Starvation

Priority Scheduling (starvation)

- When a process is perpetually (continuing forever) denied necessary resources to process its work.
- Let’s take a look at a starvation situation.

Priority Scheduling (starvation)

PID	CPU time	Arrival time	Priority
P0	5	0	12
P1	1	0	5



Priority Scheduling (starvation)

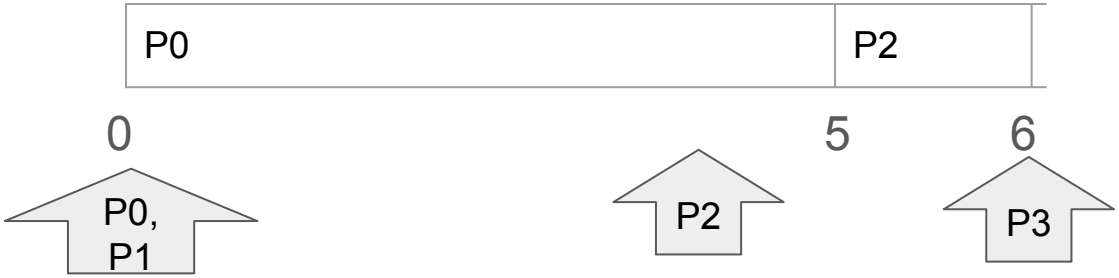
PID	CPU time	Arrival time	Priority
P0	5	0	12
P1	1	0	5

Priority Scheduling (starvation)

PID	CPU time	Arrival time	Priority
P0	5	0	12
P1	1	0	5
P2	1	4	15

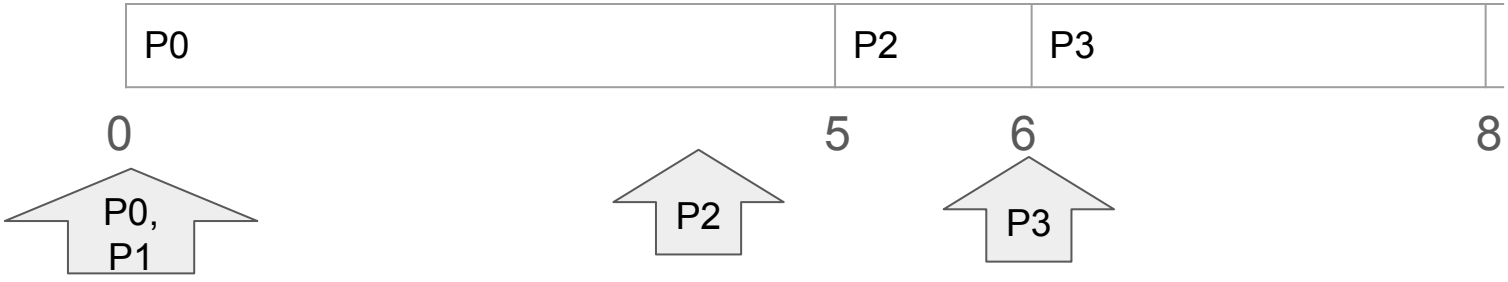
Priority Scheduling (starvation)

PID	CPU time	Arrival time	Priority
P0	5	0	12
P1	1	0	5
P2	1	4	15
P3	2	6	18



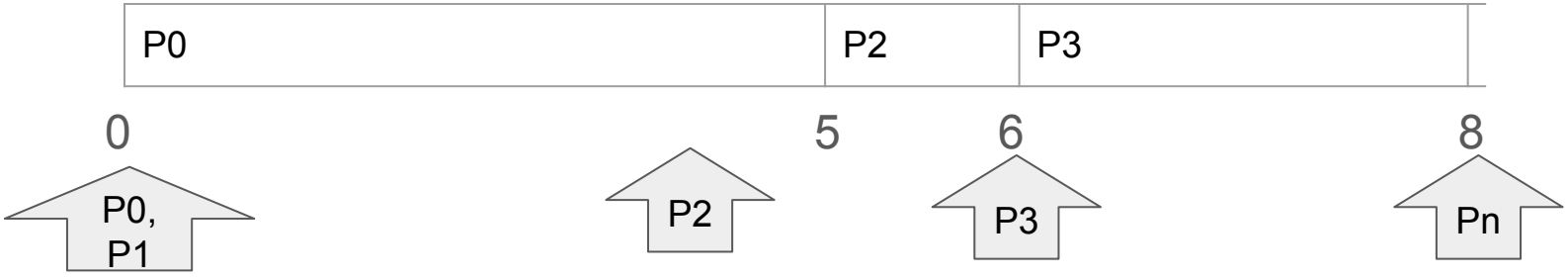
Priority Scheduling (starvation)

PID	CPU time	Arrival time	Priority
P0	5	0	12
P1	1	0	5
P2	1	4	15
P3	2	6	18



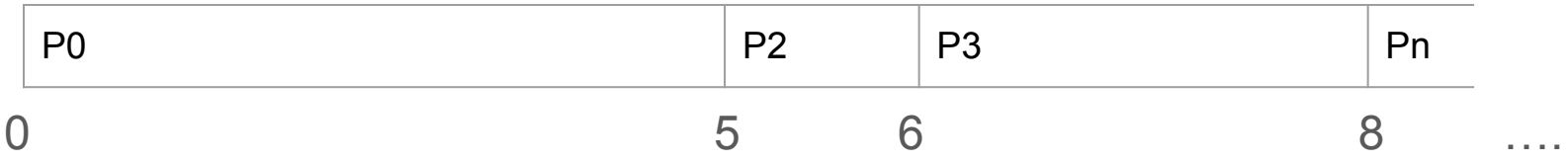
Priority Scheduling (starvation)

PID	CPU time	Arrival time	Priority
P0	5	0	12
P1	1	0	5
P2	1	4	15
P3	2	6	18
Pn	...	...	...



Priority Scheduling (starvation)

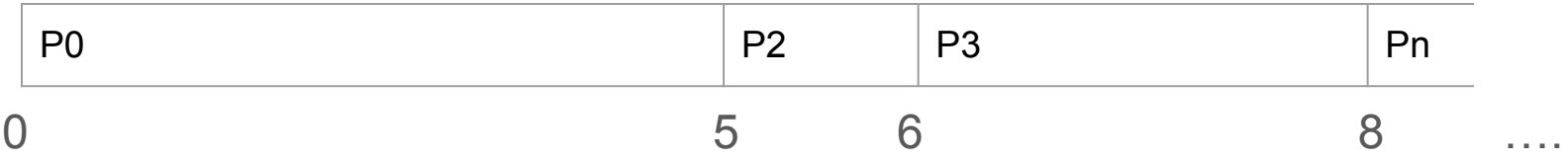
PID	CPU time	Arrival time	Priority
P0	5	0	12
P1	1	0	5
P2	1	4	15
P3	2	6	18
Pn	...	...	...



Priority Scheduling (starvation)

PID	CPU time	Arrival time	Priority
P0	5	0	12
P1	1	0	5
P2	1	4	15
P3	2	6	18
Pn	...	...	...

Process 1

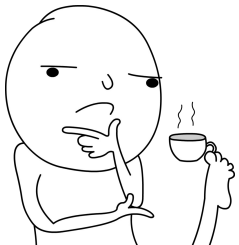


Priority Scheduling Problem

- Starvation — possible for low-priority processes:
    - A process with higher priority keeps arriving for an extended period of time, keeping a low priority process in the ready stage without a chance of getting execution time.

Priority Scheduling Problem

- Starvation — possible for low-priority processes:
    - A process with higher priority keeps arriving for an extended period of time, keeping a low priority process in the ready stage without a chance of getting execution time.
- How can we fix it?



Priority Scheduling Problem

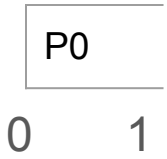
- Starvation — possible for low-priority processes:
    - A process with higher priority keeps arriving for an extended period of time, keeping a low priority process in the ready stage without a chance of getting execution time.
- Solution is **Aging** – as time progresses, increase the priority of the process

Priority Scheduling (aging)

PID	CPU time	Arrival time	Priority
P0	5	0	12
P1	1	0	6

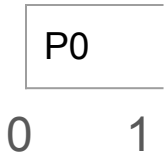
Priority Scheduling (aging)

PID	CPU time	Arrival time	Priority
P0	5	0	12
P1	1	0	6



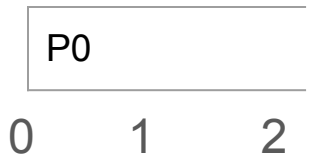
Priority Scheduling (aging)

PID	CPU time	Arrival time	Priority
P0	5	0	12
P1	1	0	8



Priority Scheduling (aging)

PID	CPU time	Arrival time	Priority
P0	5	0	12
P1	1	0	10



Priority Scheduling (aging)

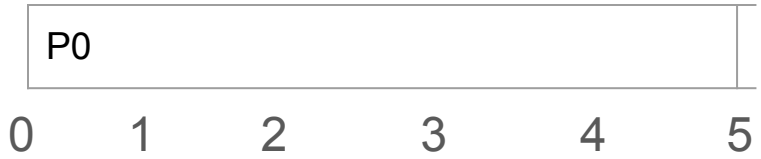
PID	CPU time	Arrival time	Priority
P0	5	0	12
P1	1	0	12

Priority Scheduling (aging)

PID	CPU time	Arrival time	Priority
P0	5	0	12
P1	1	0	14

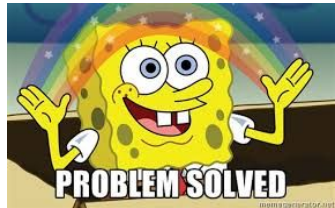
Priority Scheduling (aging)

PID	CPU time	Arrival time	Priority
P0	5	0	12
P1	1	0	16
P2	1	5	15



Priority Scheduling (aging)

PID	CPU time	Arrival time	Priority
P0	5	0	12
P1	1	0	16
P2	1	5	17



Preemptive scheduling

CPU Scheduling Algorithms

Preemptive:

- Round Robin (RR)
- Shortest Job First (SJF)
- Priority (PP)

Non-Preemptive:

- **Shortest Job First (SJF)**
- **First Come First Serve (FCFS)**
- **Priority (P)**

Round-Robin (RR)

- preemptive version of FCFS

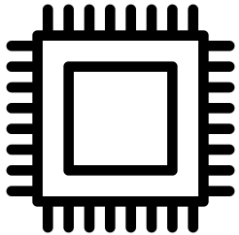
- define a fixed time slice (also called a time quantum) – e.g. 10-100ms

- implement using hardware timer that interrupts at periodic intervals

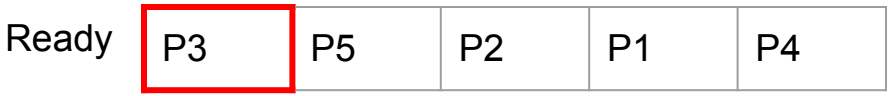
Round-Robin (RR) algorithm

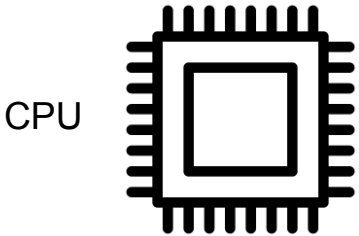
Ready	P3	P5	P2	P1	P4

CPU



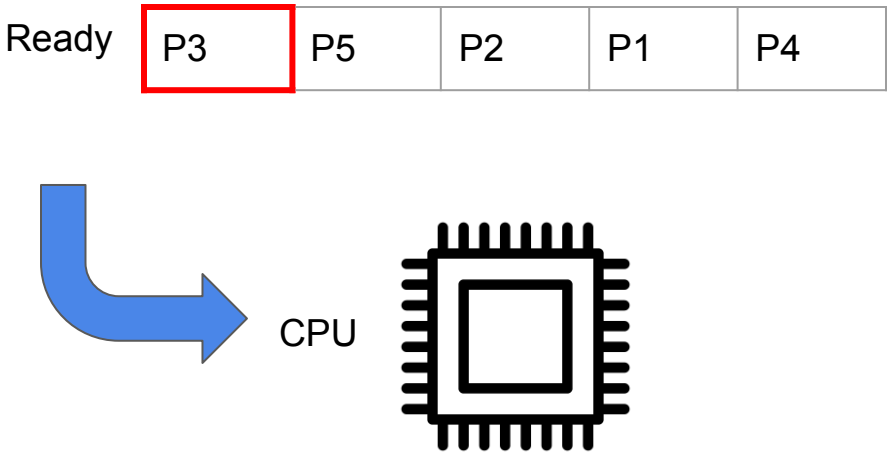
Round-Robin (RR) algorithm





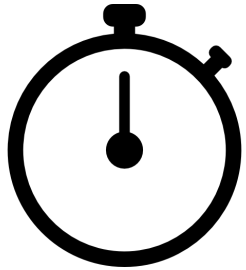
- choose process from head of ready queue

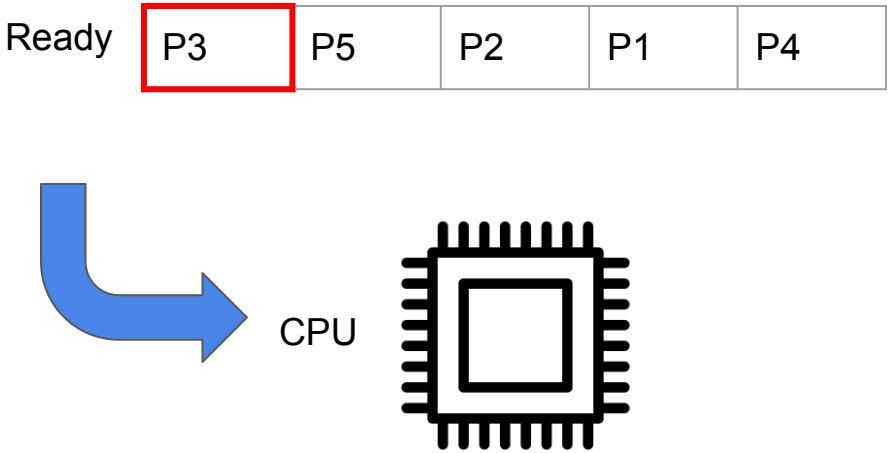
Round-Robin (RR) algorithm



- run that process for at most one time slice

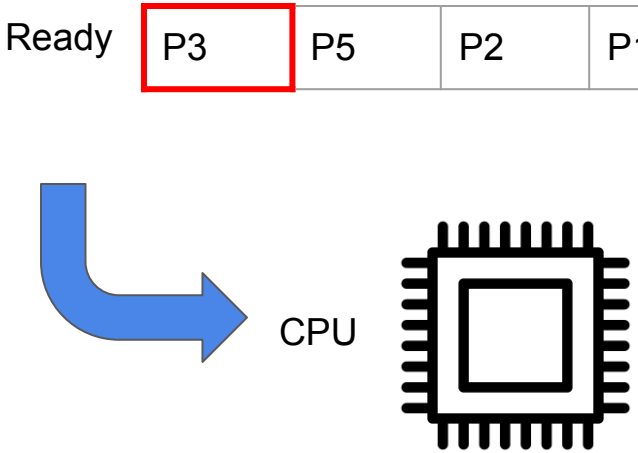
Round-Robin (RR) algorithm





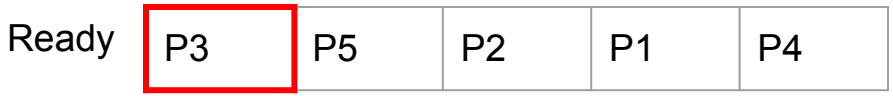
- run that process for at most one time slice

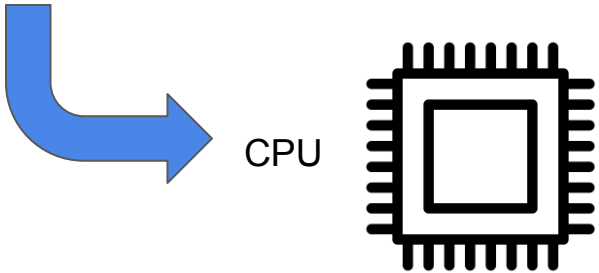
Round-Robin (RR) algorithm



- run that process for at most one time slice

Round-Robin (RR) algorithm

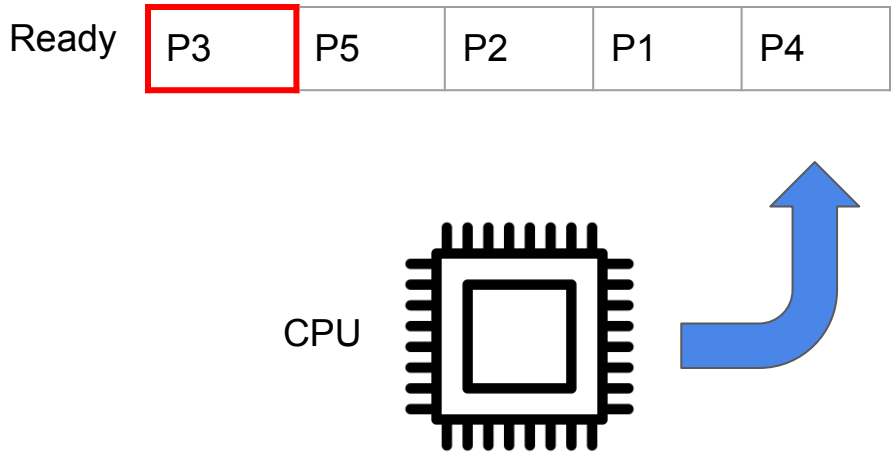




1 time slice. done!

- run that process for at most one time slice

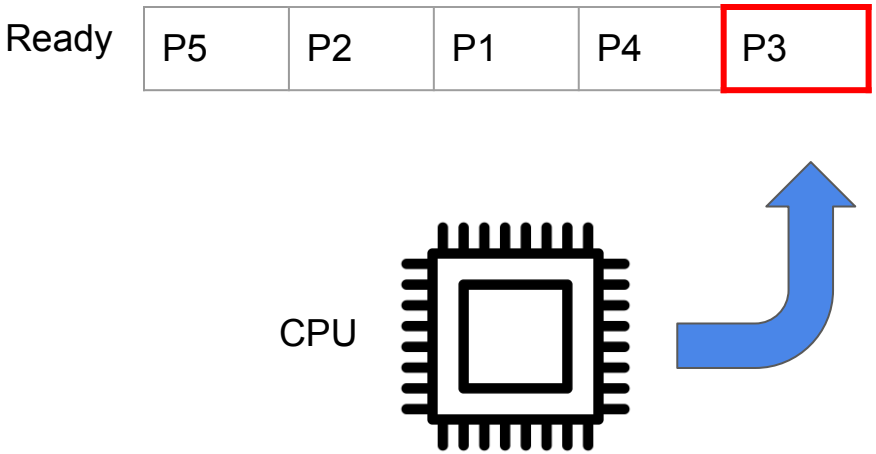
Round-Robin (RR) algorithm



1 time slice. done!

- run that process for at most one time slice, and if it hasn’t completed or blocked, add it to the tail of the ready queue

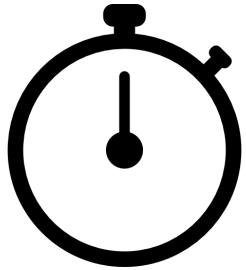
Round-Robin (RR) algorithm

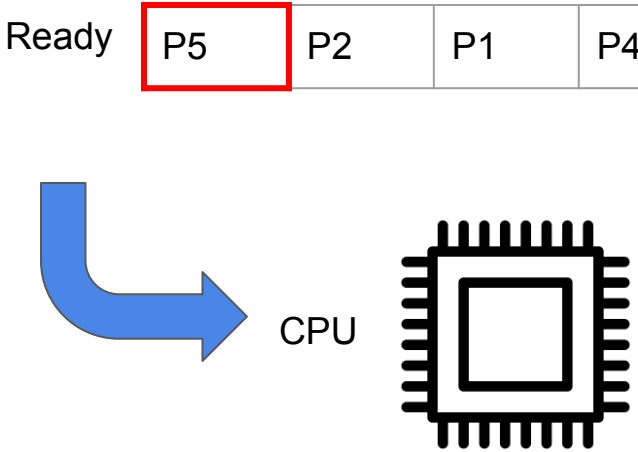


1 time slice. done!

- run that process for at most one time slice, and if it hasn’t completed or blocked, add it to the tail of the ready queue

Round-Robin (RR) algorithm



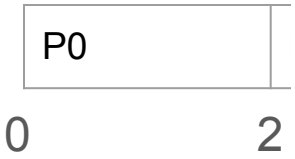


- choose another process from the head of the ready queue, and run that process for at most one time slice …

Round-Robin (RR)

Quantum = 2

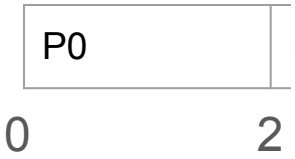
Process	CPU time
P0	5
P1	1
P2	3
P3	2



Round-Robin (RR)

Quantum = 2

Process	CPU time
P0	~~5~~ 3
P1	1
P2	3
P3	2



Round-Robin (RR)

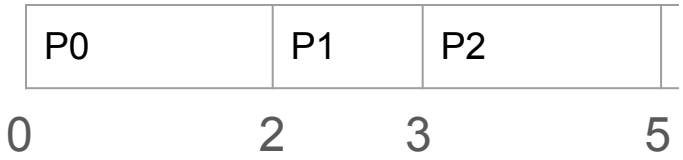
Quantum = 2

Process	CPU time
P0	~~5~~ 3
P1	~~1~~
P2	3
P3	2

Round-Robin (RR)

Quantum = 2

Process	CPU time
P0	~~5~~ 3
P1	~~1~~
P2	~~3~~ 1
P3	2



Round-Robin (RR)

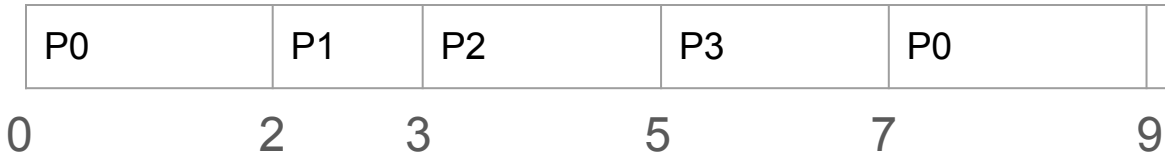
Quantum = 2

Process	CPU time
P0	~~5~~ 3
P1	~~1~~
P2	~~3~~ 1
P3	~~2~~

Round-Robin (RR)

Quantum = 2

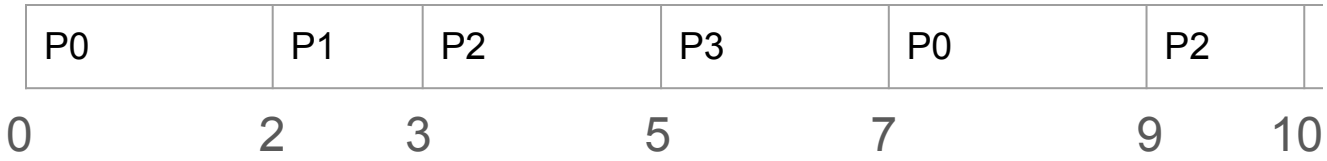
Process	CPU time
P0	~~5~~ ~~3~~ 1
P1	~~1~~
P2	~~3~~ 1
P3	~~2~~



Round-Robin (RR)

Quantum = 2

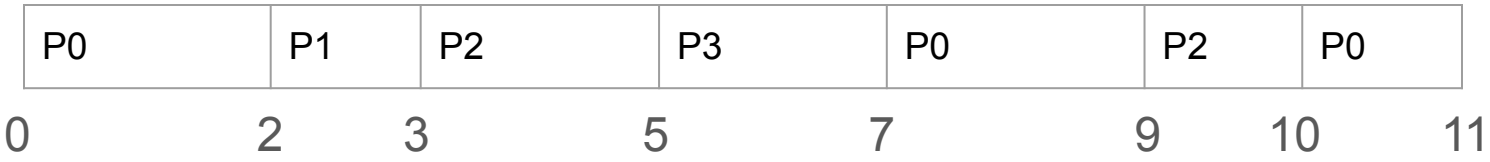
Process	CPU time
P0	~~5~~ ~~3~~ 1
P1	~~1~~
P2	~~3~~ ~~1~~
P3	~~2~~



Round-Robin (RR)

Quantum = 2

Process	CPU time
P0	~~5~~ ~~3~~ ~~1~~
P1	~~1~~
P2	~~3~~ ~~1~~
P3	~~2~~

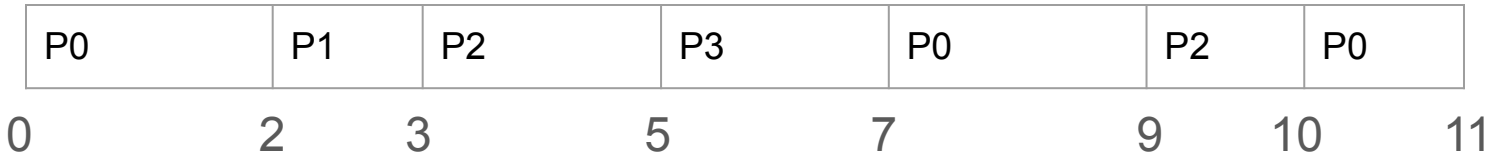


Round-Robin (RR)

Quantum = 2

Process	CPU time
P0	5
P1	1
P2	3
P3	2

	Wait-time	Turnaround-time
P0	start-arrival	
P1		
P2		
P3		

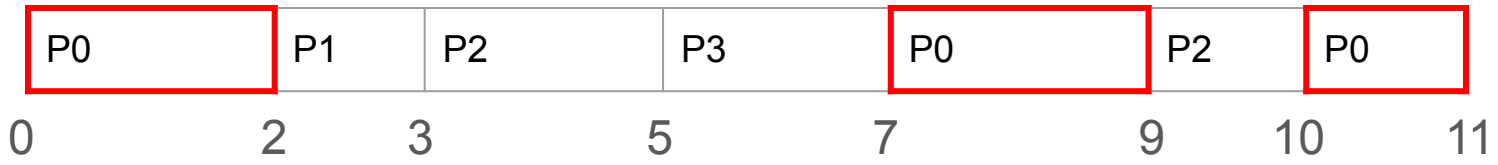


Round-Robin (RR)

Quantum = 2

Process	CPU time
P0	5
P1	1
P2	3
P3	2

	Wait-time	Turnaround-time
P0	start-arrival	
P1		
P2		
P3		

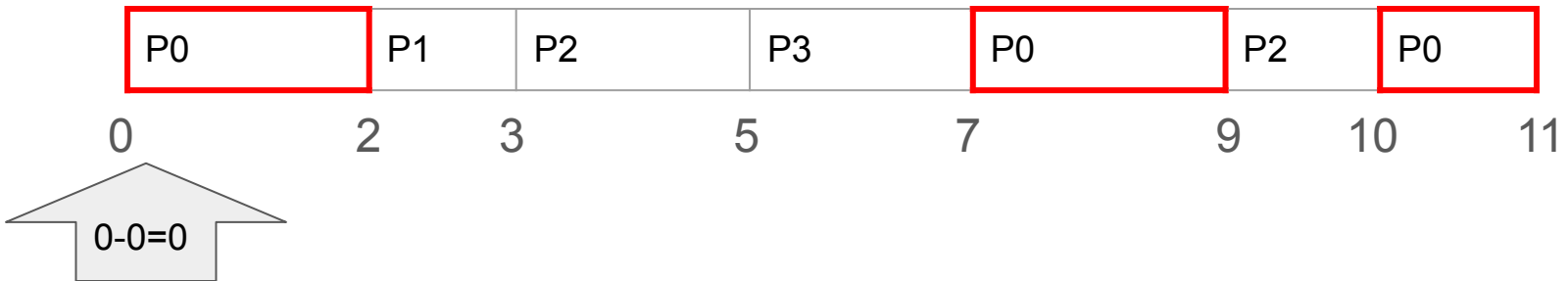


Round-Robin (RR)

Quantum = 2

Process	CPU time
P0	5
P1	1
P2	3
P3	2

	Wait-time	Turnaround-time
P0	start-arrival	
P1		
P2		
P3		

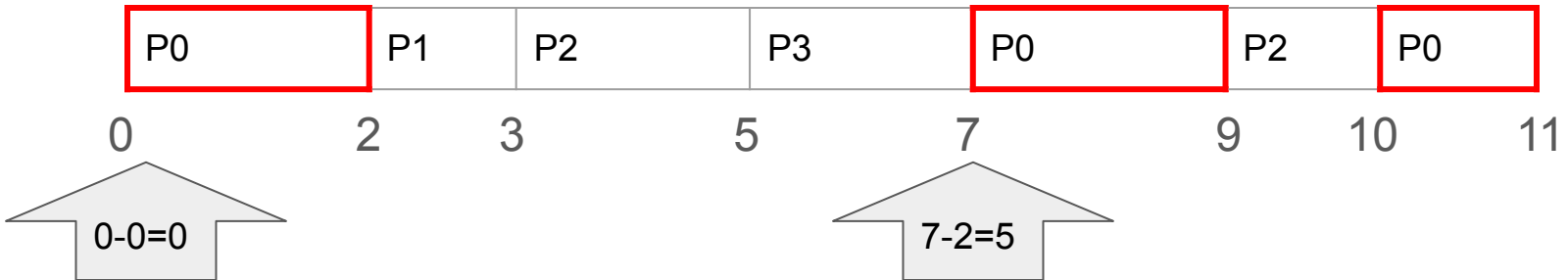


Round-Robin (RR)

Quantum = 2

Process	CPU time
P0	5
P1	1
P2	3
P3	2

	Wait-time	Turnaround-time
P0	start-arrival	
P1		
P2		
P3		

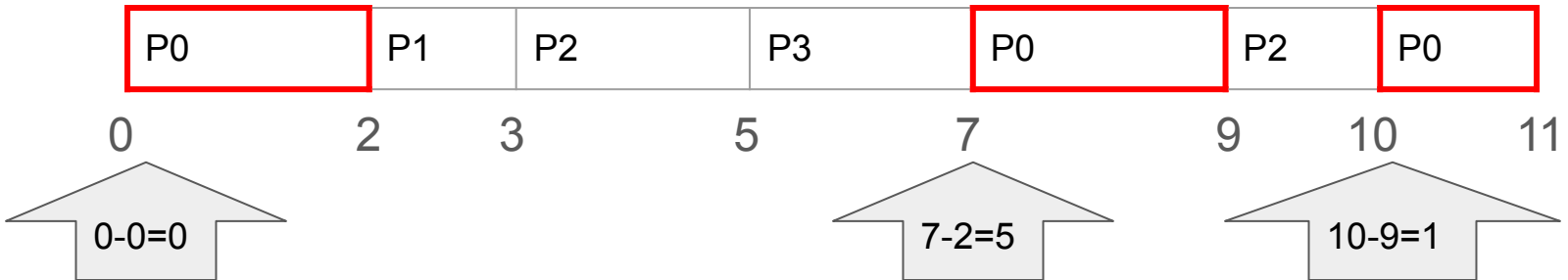


Round-Robin (RR)

Quantum = 2

Process	CPU time
P0	5
P1	1
P2	3
P3	2

	Wait-time	Turnaround-time
P0	start-arrival	
P1		
P2		
P3		

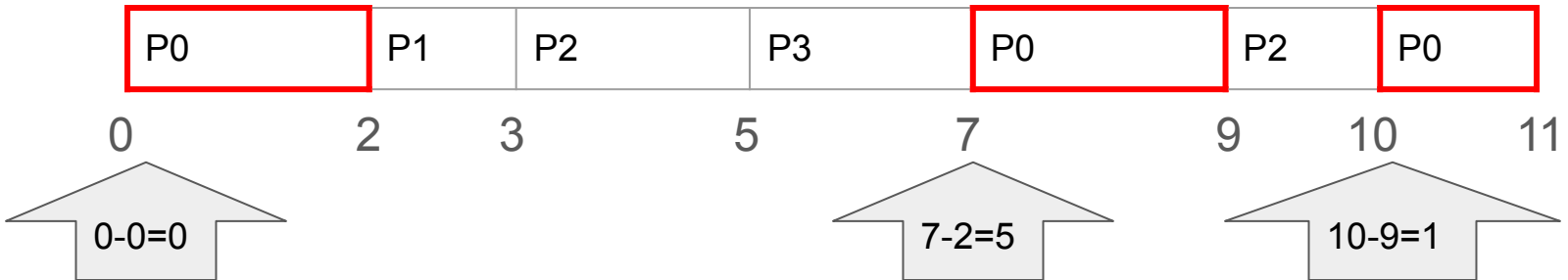


Round-Robin (RR)

Quantum = 2

Process	CPU time
P0	5
P1	1
P2	3
P3	2

	Wait-time	Turnaround-time
P0	0 + 5 + 1= 6	
P1		
P2		
P3		

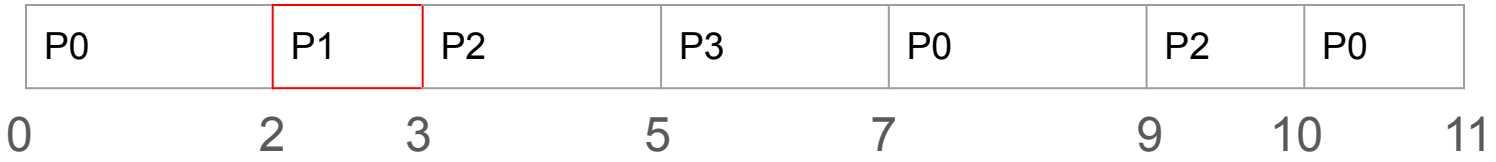


Round-Robin (RR)

Quantum = 2

Process	CPU time
P0	5
P1	1
P2	3
P3	2

	Wait-time	Turnaround-time
P0	0 + 5 + 1= 6	
P1		
P2		
P3		

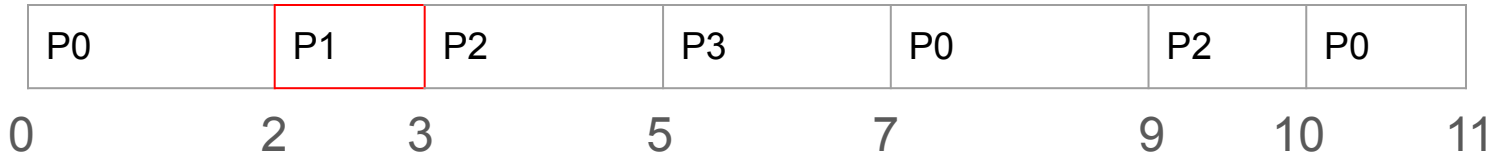


Round-Robin (RR)

Quantum = 2

Process	CPU time
P0	5
P1	1
P2	3
P3	2

	Wait-time	Turnaround-time
P0	0 + 5 + 1= 6	
P1	2 - 0 = 2	
P2		
P3		

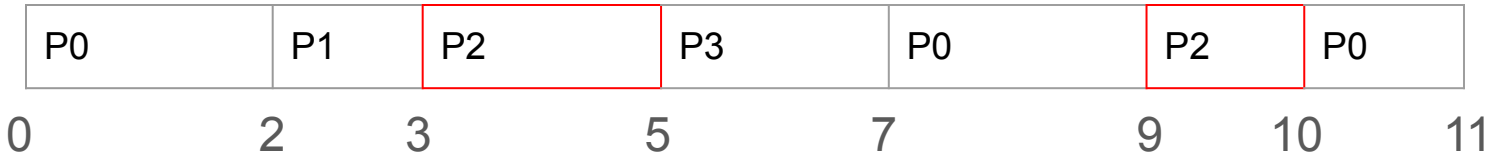


Round-Robin (RR)

Quantum = 2

Process	CPU time
P0	5
P1	1
P2	3
P3	2

	Wait-time	Turnaround-time
P0	0 + 5 + 1= 6	
P1	2 - 0 = 2	
P2		
P3		

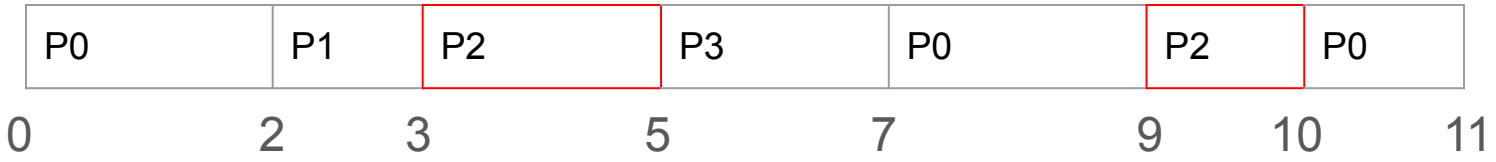


Round-Robin (RR)

Quantum = 2

Process	CPU time
P0	5
P1	1
P2	3
P3	2

	Wait-time	Turnaround-time
P0	0 + 5 + 1= 6	
P1	(2 - 0) = 2	
P2	(3-0)+(9-5)=7	
P3		

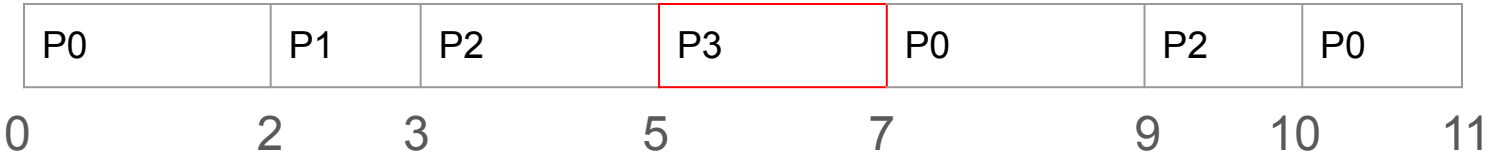


Round-Robin (RR)

Quantum = 2

Process	CPU time
P0	5
P1	1
P2	3
P3	2

	Wait-time	Turnaround-time
P0	0 + 5 + 1= 6	
P1	(2 - 0) = 2	
P2	(3-0)+(9-5)=7	
P3		

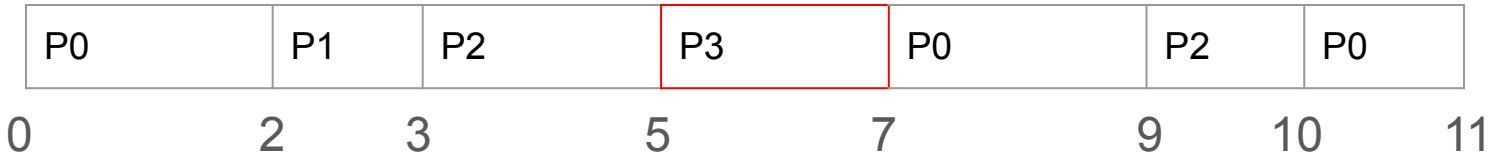


Round-Robin (RR)

Quantum = 2

Process	CPU time
P0	5
P1	1
P2	3
P3	2

	Wait-time	Turnaround-time
P0	0 + 5 + 1= 6	
P1	(2 - 0) = 2	
P2	(3-0)+(9-5)=7	
P3	(5-0)=5	

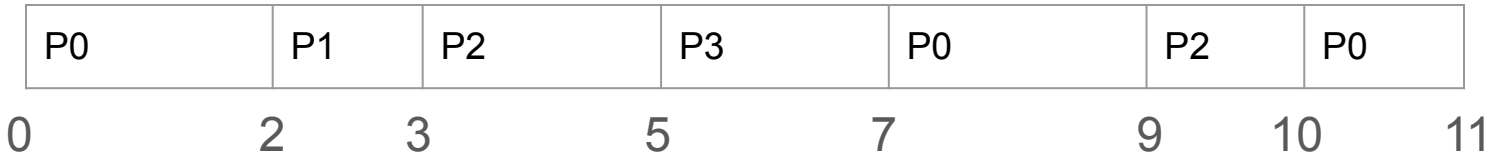


Round-Robin (RR)

Quantum = 2

Process	CPU time
P0	5
P1	1
P2	3
P3	2

	Wait-time	Turnaround-time
P0	0 + 5 + 1= 6	Wait + CPU_time
P1	(2 - 0) = 2	
P2	(3-0)+(9-5)=7	
P3	(5-0)=5	

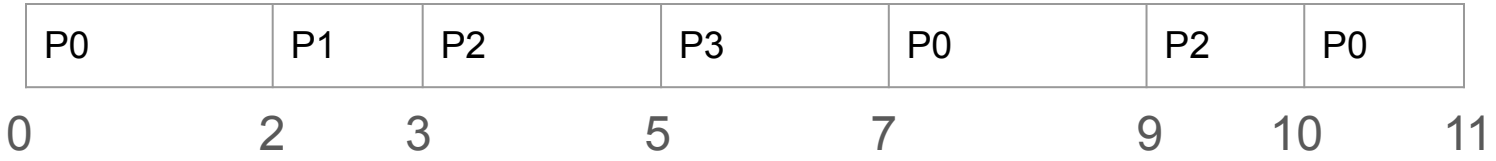


Round-Robin (RR)

Quantum = 2

Process	CPU time
P0	5
P1	1
P2	3
P3	2

	Wait-time	Turnaround-time
P0	0 + 5 + 1= 6	6 + 5 = 11
P1	(2 - 0) = 2	2 + 1 = 3
P2	(3-0)+(9-5)=7	7 + 3 = 10
P3	(5-0)=5	5 + 2 = 7



Round-Robin (RR)

- preemptive **(at end of time slice)**
- Long processes may have to wait n*q time units for another time slice
  - n = number of other processes
  - q = length of time slice

	Wait-time	**Turnaround-time**
P0	0 + 5 + 1= 6	6 + 5 = 11
P1	(2 - 0) = 2	2 + 1 = 3
P2	(3-0)+(9-5)=7	7 + 3 = 10
P3	(5-0)=5	5 + 2 = 7
AVG	5	7.75

Round-Robin (RR)

- preemptive **(at end of time slice)**
- Long processes may have to wait n*q time units for another time slice
  - n = number of other processes
  - q = length of time slice
- Throughput — depends on time slice
  - too small — too many context switches
  - too large — approximates FCFS

	Wait-time	**Turnaround-time**
P0	0 + 5 + 1= 6	6 + 5 = 11
P1	(2 - 0) = 2	2 + 1 = 3
P2	(3-0)+(9-5)=7	7 + 3 = 10
P3	(5-0)=5	5 + 2 = 7
AVG	5	7.75

Round-Robin (RR)

- preemptive **(at end of time slice)**
- Long processes may have to wait n*q time units for another time slice
  - n = number of other processes
  - q = length of time slice
- Throughput — depends on time slice
  - too small — too many context switches
  - too large — approximates FCFS
- fairness — penalizes I/O-bound processes (may not use full time slice)

	Wait-time	**Turnaround-time**
P0	0 + 5 + 1= 6	6 + 5 = 11
P1	(2 - 0) = 2	2 + 1 = 3
P2	(3-0)+(9-5)=7	7 + 3 = 10
P3	(5-0)=5	5 + 2 = 7
AVG	5	7.75

Round-Robin (RR)

- preemptive **(at end of time slice)**
- Long processes may have to wait n*q time units for another time slice
  - n = number of other processes
  - q = length of time slice
- Throughput — depends on time slice
  - too small — too many context switches
  - too large — approximates FCFS
- fairness — penalizes I/O-bound processes (may not use full time slice)
- starvation — not possible
- overhead — somewhat low

	Wait-time	Turnaround-time
P0	0 + 5 + 1= 6	6 + 5 = 11
P1	(2 - 0) = 2	2 + 1 = 3
P2	(3-0)+(9-5)=7	7 + 3 = 10
P3	(5-0)=5	5 + 2 = 7
AVG	5	7.75

RR with arrival time

Round-Robin (RR) (with arrival time)

Quantum = 2

Process	CPU time	Arrival
P0	6	0
P1	4	3
P2	3	5
P3	2	7

Can you calculate avg. WT & TT?

small trick in this question

Round-Robin (RR) (with arrival time)

Quantum = 2

Process	CPU time	Arrival
P0	6	0
P1	4	3
P2	3	5
P3	2	7

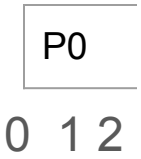
0

Round-Robin (RR) (with arrival time)

Quantum = 2

Process	CPU time	Arrival
P0	6	0
P1	4	3
P2	3	5
P3	2	7

ready



Time slice is over, what should the scheduler do?

Round-Robin (RR) (with arrival time)

Quantum = 2

Process	CPU time	Arrival
P0	6	0
P1	4	3
P2	3	5
P3	2	7

ready	P1			

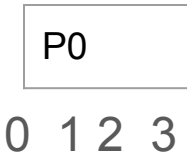
P0	

0 1 2 3

Round-Robin (RR) (with arrival time)

Quantum = 2

Process	CPU time	Arrival
P0	6	0
P1	4	3
P2	3	5
P3	2	7

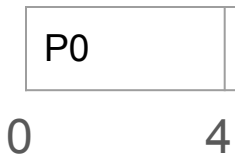


Preemption occurs only at the end of the time slice!

Round-Robin (RR) (with arrival time)

Quantum = 2

Process	CPU time	Arrival
P0	6	0
P1	4	3
P2	3	5
P3	2	7

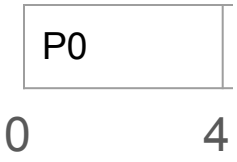


Round-Robin (RR) (with arrival time)

Quantum = 2

Process	CPU time	Arrival
P0	6	0
P1	4	3
P2	3	5
P3	2	7

ready	P1	P0		



At the end of the time slice the process returns to the ready Queue

Round-Robin (RR) (with arrival time)

Quantum = 2

Process	CPU time	Arrival
P0	~~6~~ 2	~~0~~ 4
P1	4	3
P2	3	5
P3	2	7

ready	P1	P0		

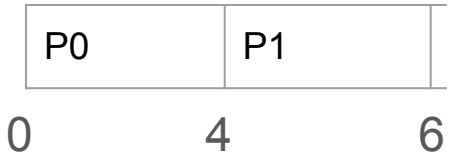
At the end of the time slice the process returns to the ready Queue. Therefore, we should update arrival time.

Round-Robin (RR) (with arrival time)

Quantum = 2

Process	CPU time	Arrival
P0	~~6~~ 2	~~0~~ 4
P1	~~4~~ 2	~~3~~ 6
P2	3	5
P3	2	7

ready		P0	P2	P1

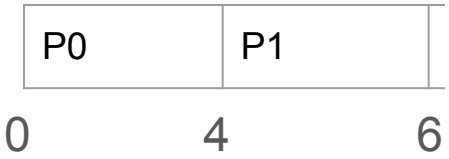


Round-Robin (RR) (with arrival time)

Quantum = 2

Process	CPU time	Arrival
P0	~~6~~ 2	~~0~~ 4
P1	~~4~~ 2	~~3~~ 6
P2	3	5
P3	2	7

ready	P0	P2	P1	

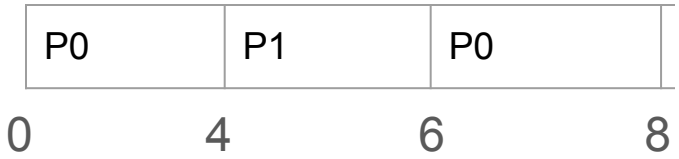


Round-Robin (RR) (with arrival time)

Quantum = 2

Process	CPU time	Arrival
P0	~~6~~ ~~2~~	~~0~~ 4
P1	~~4~~ 2	~~3~~ 6
P2	3	5
P3	2	7

ready		P2	P1	P3

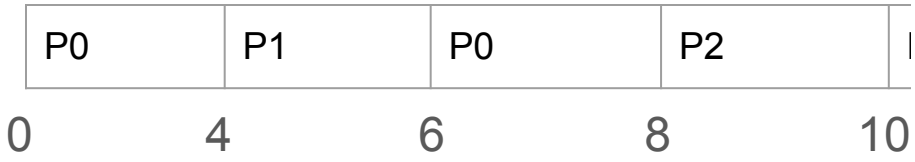


Round-Robin (RR) (with arrival time)

Quantum = 2

Process	CPU time	Arrival
P0	~~6~~ ~~2~~	~~0~~ 4
P1	~~4~~ 2	~~3~~ 6
P2	~~3~~ 1	~~5~~ 10
P3	2	7

ready			P1	P3

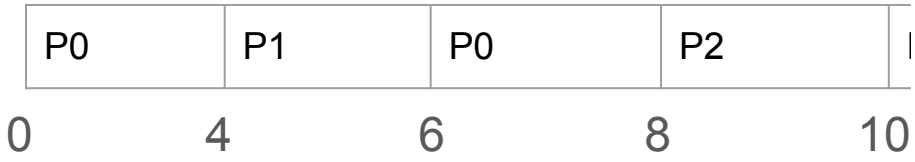


Round-Robin (RR) (with arrival time)

Quantum = 2

Process	CPU time	Arrival
P0	~~6~~ ~~2~~	~~0~~ 4
P1	~~4~~ 2	~~3~~ 6
P2	~~3~~ 1	~~5~~ 10
P3	2	7

ready	P1	P3	P2	

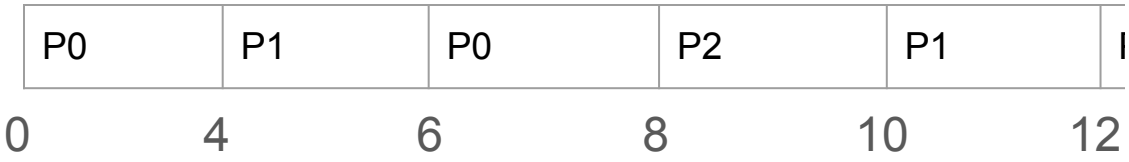


Round-Robin (RR) (with arrival time)

Quantum = 2

Process	CPU time	Arrival
P0	~~6~~ ~~2~~	~~0~~ 4
P1	~~4~~ ~~2~~	~~3~~ 6
P2	~~3~~ 1	~~5~~ 10
P3	2	7

ready		P3	P2	

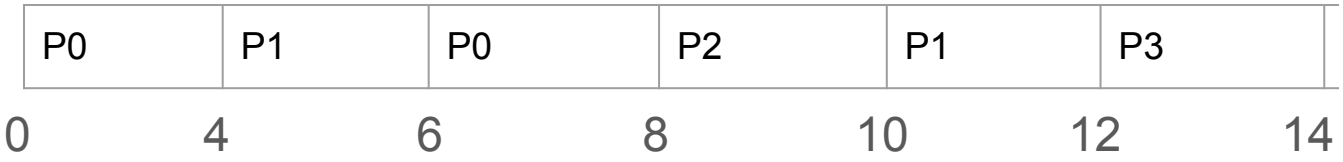


Round-Robin (RR) (with arrival time)

Quantum = 2

Process	CPU time	Arrival
P0	~~6~~ ~~2~~	~~0~~ 4
P1	~~4~~ ~~2~~	~~3~~ 6
P2	~~3~~ 1	~~5~~ 10
P3	~~2~~	7

ready			P2	

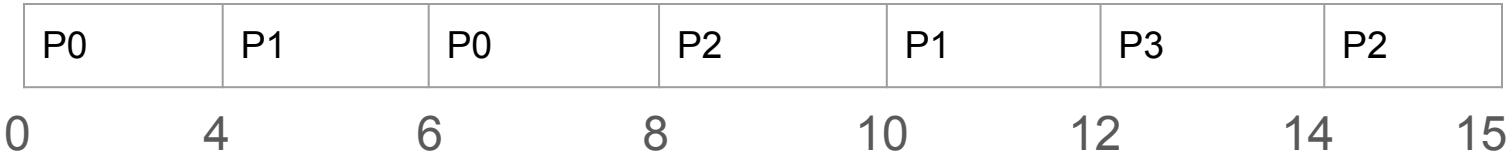


Round-Robin (RR) (with arrival time)

Quantum = 2

Process	CPU time	Arrival
P0	~~6~~ ~~2~~	~~0~~ 4
P1	~~4~~ ~~2~~	~~3~~ 6
P2	~~3~~ ~~1~~	~~5~~ 10
P3	~~2~~	7

ready

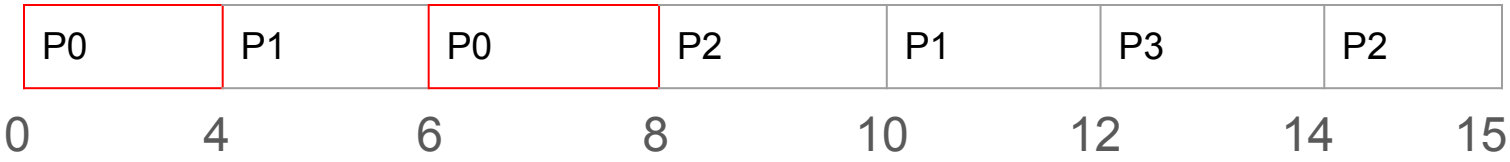


Round-Robin (RR) (with arrival time)

Quantum = 2

Process	CPU time	Arrival
P0	6	0
P1	4	3
P2	3	5
P3	2	7

	Wait-time	Turnaround-time
P0	start-arrival	
P1		
P2		
P3		

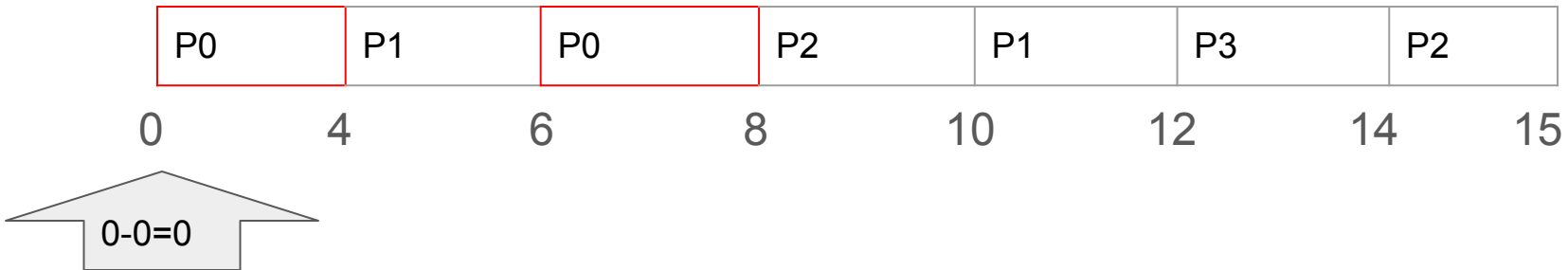


Round-Robin (RR) (with arrival time)

Quantum = 2

Process	CPU time	Arrival
P0	6	0
P1	4	3
P2	3	5
P3	2	7

	Wait-time	Turnaround-time
P0	(0-0)+	
P1		
P2		
P3		

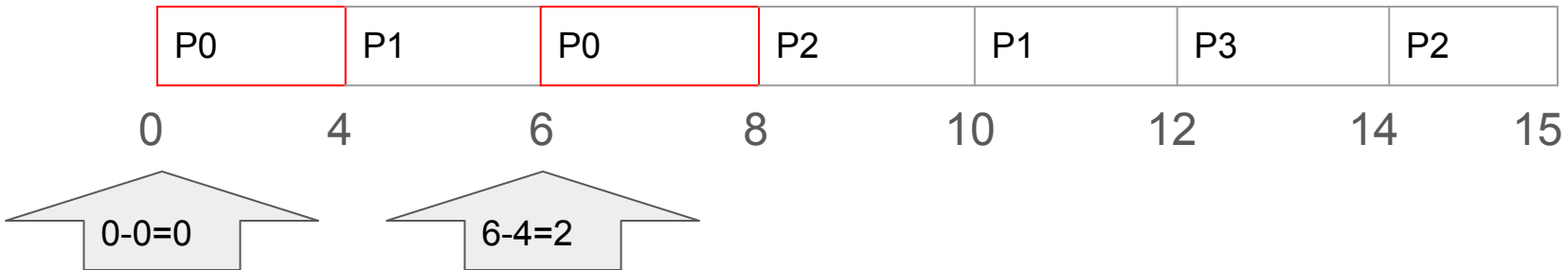


Round-Robin (RR) (with arrival time)

Quantum = 2

Process	CPU time	Arrival
P0	6	0
P1	4	3
P2	3	5
P3	2	7

	Wait-time	Turnaround-time
P0	(0-0)+(6-4)=2	
P1		
P2		
P3		

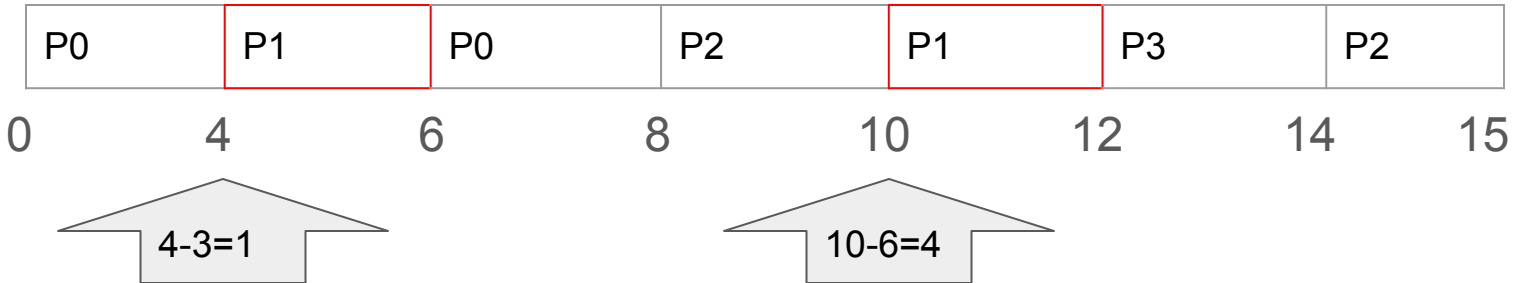


Round-Robin (RR) (with arrival time)

Quantum = 2

Process	CPU time	Arrival
P0	6	0
P1	4	3
P2	3	5
P3	2	7

	Wait-time	Turnaround-time
P0	(0-0)+(6-4)=2	
P1	(4-3)+(10-6)=5	
P2		
P3		

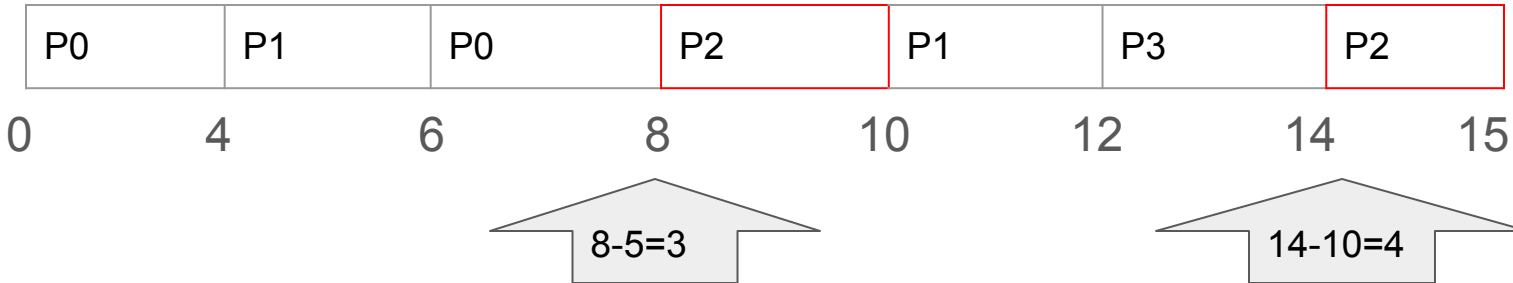


Round-Robin (RR) (with arrival time)

Quantum = 2

Process	CPU time	Arrival
P0	6	0
P1	4	3
P2	3	5
P3	2	7

	Wait-time	Turnaround-time
P0	(0-0)+(6-4)=2	
P1	(4-3)+(10-6)=5	
P2	(8-5)+(14-10)=7	
P3		

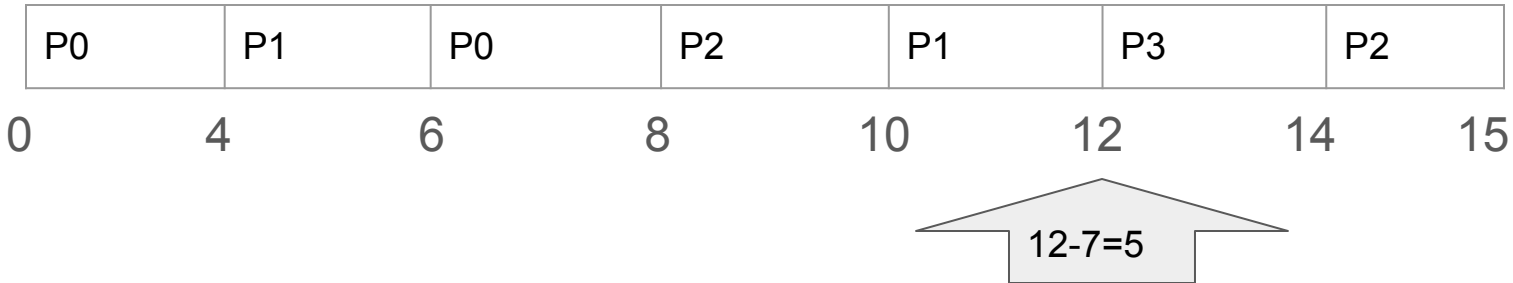


Round-Robin (RR) (with arrival time)

Quantum = 2

Process	CPU time	Arrival
P0	6	0
P1	4	3
P2	3	5
P3	2	7

	Wait-time	Turnaround-time
P0	(0-0)+(6-4)=2	
P1	(4-3)+(10-6)=5	
P2	(8-5)+(14-10)=7	
P3	12-7=5	

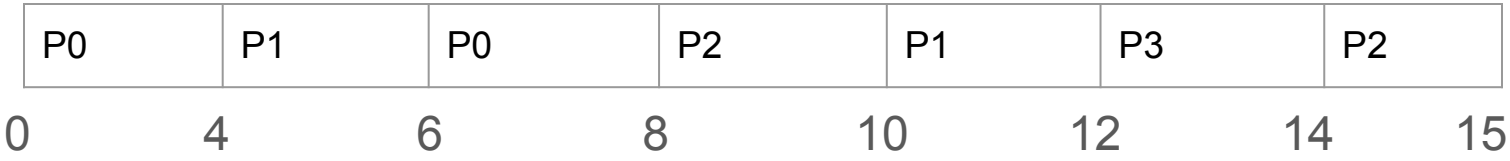


Round-Robin (RR) (with arrival time)

Quantum = 2

Process	CPU time	Arrival
P0	6	0
P1	4	3
P2	3	5
P3	2	7

	Wait-time	Turnaround-time
P0	(0-0)+(6-4)=2	2 + 6 = 8
P1	(4-3)+(10-6)=5	5 + 4 = 9
P2	(8-5)+(14-10)=7	7 + 3 = 10
P3	12-7=5	5 + 2 = 7



CPU Scheduling Algorithms

Preemptive:

- **Round Robin (RR)**
- Shortest Remaining Time (SRT)
- Priority (PP)

Non-Preemptive:

- **Shortest Job First (SJF)**
- **First Come First Serve (FCFS)**
- **Priority (P)**

Shortest Remaining Time (SRT)

- Also Known As Shortest-Job-First (SJF) with preemption

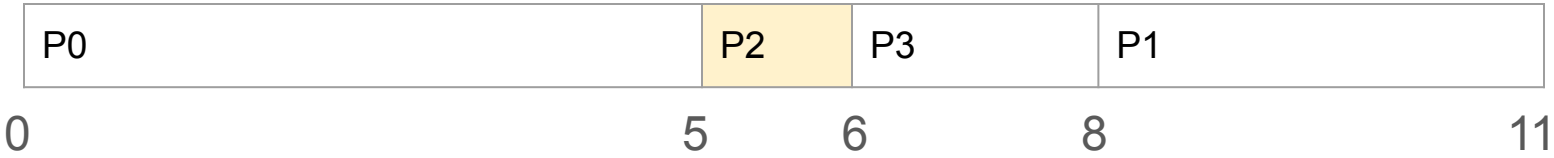
- Associate with each process the length of its next CPU burst
- Use these lengths to schedule the process with the shortest time

- SRT is optimal – gives minimum average waiting time for a given set of processes
- The difficulty is knowing the length of the next CPU request
- preemptive (**at arrival of process into ready queue**)

Flashback: Shortest Job First (non-preemptive)

Process	CPU time	Arrival time
P0	5	0
P1	3	1
P2	1	3
P3	2	4

	Wait-time	Turnaround-time
P0	0	0 + 5 = 5
P1	8 - 1 = 7	7 + 3 = 10
P2	5 - 3 = 2	2 + 1 = 3
P3	6 - 4 = 2	2 + 2 = 4



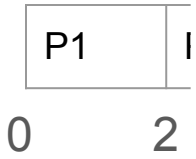
P2 arrived at time 3 with a burst-time of 1, yet it had to wait for 2 time slots to run.

Shortest Remaining Time (SRT) (with arrival time)

Process	CPU time	Arrival
P1	7	0
P2	4	2
P3	1	4
P4	4	5

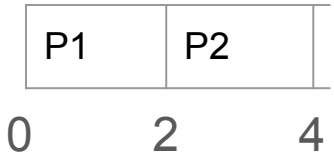
Shortest Remaining Time (SRT) (with arrival time)

Process	CPU time	Arrival
P1	~~7~~ 5	0
P2	4	2
P3	1	4
P4	4	5



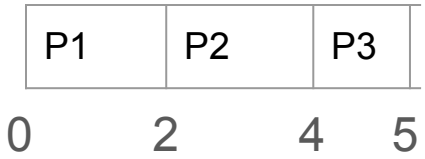
Shortest Remaining Time (SRT) (with arrival time)

Process	CPU time	Arrival
P1	~~7~~ 5	0
P2	~~4~~ 2	2
P3	1	4
P4	4	5



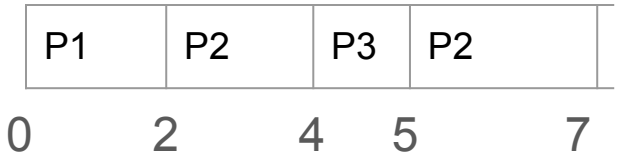
Shortest Remaining Time (SRT) (with arrival time)

Process	CPU time	Arrival
P1	~~7~~ 5	0
P2	~~4~~ 2	2
P3	~~1~~	4
P4	4	5



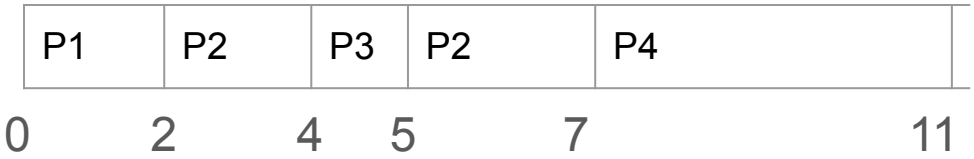
Shortest Remaining Time (SRT) (with arrival time)

Process	CPU time	Arrival
P1	~~7~~ 5	0
P2	~~4~~ ~~2~~	2
P3	~~1~~	4
P4	4	5



Shortest Remaining Time (SRT) (with arrival time)

Process	CPU time	Arrival
P1	~~7~~ 5	0
P2	~~4~~ ~~2~~	2
P3	~~1~~	4
P4	~~4~~	5



Shortest Remaining Time (SRT) (with arrival time)

Process	CPU time	Arrival
P1	~~7~~ ~~5~~	0
P2	~~4~~ ~~2~~	2
P3	~~1~~	4
P4	~~4~~	5

Shortest Remaining Time (SRT) (with arrival time)

Process	CPU time	Arrival
P1	7	0
P2	4	2
P3	1	4
P4	4	5

	Wait-time	Turnaround-time
P1	0 + 9 = 9	9 + 7 = 16
P2	1	1 + 4 = 5
P3	0	0 + 1 = 1
P4	2	2 + 4 = 6

Shortest Remaining Time (SRT) (with arrival time)

- preemptive **(at arrival of process into ready queue)**
- provably optimal waiting time
- fairness?

	Wait-time	**Turnaround-time**
P1	0 + 9 = 9	9 + 7 = 16
P2	1	1 + 4 = 5
P3	0	0 + 1 = 1
P4	2	2 + 4 = 6
AVG	3	7

Shortest Remaining Time (SRT) (with arrival time)

- preemptive **(at arrival of process into ready queue)**
- provably optimal waiting time
- fairness — penalizes long processes
- note: long processes eventually become short processes
- starvation?

	Wait-time	**Turnaround-time**
P1	0 + 9 = 9	9 + 7 = 16
P2	1	1 + 4 = 5
P3	0	0 + 1 = 1
P4	2	2 + 4 = 6
AVG	3	7

Shortest Remaining Time (SRT) (with arrival time)

- preemptive **(at arrival of process into ready queue)**
- provably optimal waiting time
- fairness — penalizes long processes
- note: long processes eventually become short processes
- starvation — possible
- overhead?

	Wait-time	**Turnaround-time**
P1	0 + 9 = 9	9 + 7 = 16
P2	1	1 + 4 = 5
P3	0	0 + 1 = 1
P4	2	2 + 4 = 6
AVG	3	7

Shortest Remaining Time (SRT) (with arrival time)

- preemptive **(at arrival of process into ready queue)**
- provably optimal waiting time
- fairness — penalizes long processes
- note: long processes eventually become short processes
- starvation — possible
- overhead — can be high
    - (requires recording and estimating CPU burst times)

	Wait-time	**Turnaround-time**
P1	0 + 9 = 9	9 + 7 = 16
P2	1	1 + 4 = 5
P3	0	0 + 1 = 1
P4	2	2 + 4 = 6
AVG	3	7

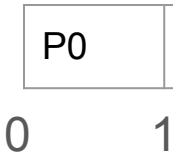
Shortest Remaining Time (SRT) (with arrival time)

Process	CPU time	Arrival
P0	5	0
P1	3	1
P2	1	9
P3	3	10

Can you calculate avg. WT & TT?

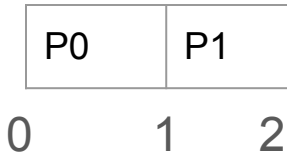
Shortest Remaining Time (SRT) (with arrival time)

Process	CPU time	Arrival
P0	~~5~~ 4	0
P1	3	1
P2	1	9
P3	3	10



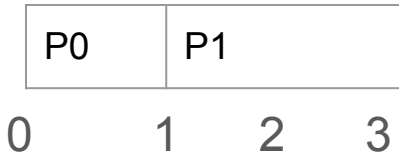
Shortest Remaining Time (SRT) (with arrival time)

Process	CPU time	Arrival
P0	~~5~~ 4	0
P1	3	1
P2	1	9
P3	3	10



Shortest Remaining Time (SRT) (with arrival time)

Process	CPU time	Arrival
P0	~~5~~ 4	0
P1	3	1
P2	1	9
P3	3	10



Shortest Remaining Time (SRT) (with arrival time)

Process	CPU time	Arrival
P0	~~5~~ 4	0
P1	~~3~~	1
P2	1	9
P3	3	10

Shortest Remaining Time (SRT) (with arrival time)

Process	CPU time	Arrival
P0	~~5~~ 4	0
P1	~~3~~	1
P2	1	9
P3	3	10

Shortest Remaining Time (SRT) (with arrival time)

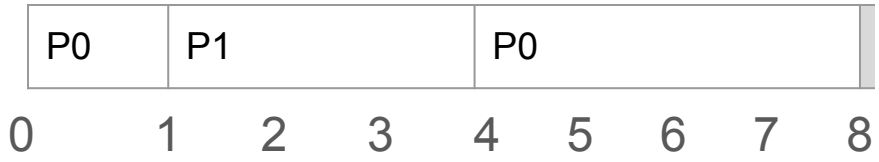
Process	CPU time	Arrival
P0	~~5~~ 4	0
P1	~~3~~	1
P2	1	9
P3	3	10

Shortest Remaining Time (SRT) (with arrival time)

Process	CPU time	Arrival
P0	~~5~~ 4	0
P1	~~3~~	1
P2	1	9
P3	3	10

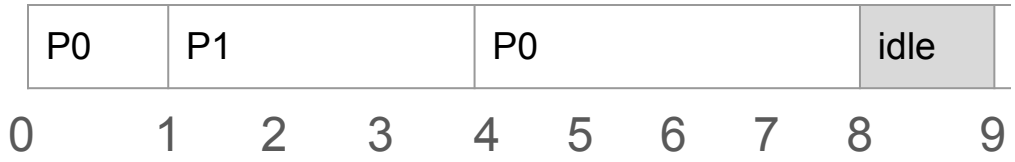
Shortest Remaining Time (SRT) (with arrival time)

Process	CPU time	Arrival
P0	~~5~~ 4	0
P1	~~3~~	1
P2	1	9
P3	3	10



Shortest Remaining Time (SRT) (with arrival time)

Process	CPU time	Arrival
P0	~~5~~ 4	0
P1	~~3~~	1
P2	1	9
P3	3	10



Shortest Remaining Time (SRT) (with arrival time)

Process	CPU time	Arrival
P0	5	0
P1	3	1
P2	1	9
P3	3	10

Shortest Remaining Time (SRT) (with arrival time)

Process	CPU time	Arrival
P0	~~5~~	0
P1	~~3~~	1
P2	~~1~~	9
P3	~~3~~	10

Shortest Remaining Time (SRT) (with arrival time)

Process	CPU time	Arrival
P0	5	0
P1	3	1
P2	1	9
P3	3	10

	Wait-time	Turnaround-time
P0	0 + 3 = 3	
P1	0	
P2	0	
P3	0	

Shortest Remaining Time (SRT) (with arrival time)

Process	CPU time	Arrival
P0	5	0
P1	3	1
P2	1	9
P3	3	10

	Wait-time	Turnaround-time
P0	0 + 3 = 3	3 + 5 = 8
P1	0	0 + 3 = 3
P2	0	0 + 1 = 1
P3	0	3 + 0 = 3

Shortest Remaining Time (SRT) (with arrival time)

- SRT gives optimal performance
- Really difficult to predict burst time in a real-time system
- Used to assess the performance of other algorithms

	Wait-time	Turnaround-time
P0	0 + 3 = 3	3 + 5 = 8
P1	0	0 + 3 = 3
P2	0	0 + 1 = 1
P3	0	3 + 0 = 3
AVG	0.75	3.75

Preemptive Priority

CPU Scheduling Algorithms

Preemptive:

- **Round Robin (RR)**
- **Shortest Remaining Time (SRT)**
- Preemptive Priority (PP)

Non-Preemptive:

- **Shortest Job First (SJF)**
- **First Come First Serve (FCFS)**
- **Priority (P)**

Preemptive Priority (PP)

- Associate a priority with each process
- Run the process with the highest priority
- **Preemptive at process arrival**

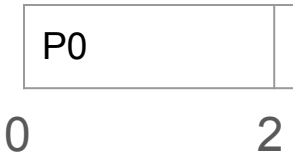
Preemptive Priority (PP)

Process	CPU time	Arrival	Priority
P0	5	0	30
P1	4	2	35
P2	1	5	36
P3	6	6	20

0

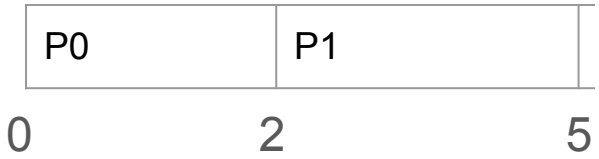
Preemptive Priority (PP)

Process	CPU time	Arrival	Priority
P0	~~5~~ 3	0	30
P1	4	2	35
P2	1	5	36
P3	6	6	20



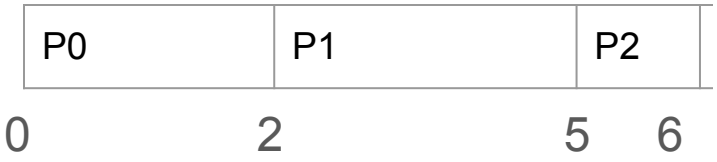
Preemptive Priority (PP)

Process	CPU time	Arrival	Priority
P0	~~5~~ 3	0	30
P1	~~4~~ 1	2	35
P2	1	5	36
P3	6	6	20



Preemptive Priority (PP)

Process	CPU time	Arrival	Priority
P0	~~5~~ 3	0	30
P1	~~4~~ 1	2	35
P2	~~1~~	5	36
P3	6	6	20

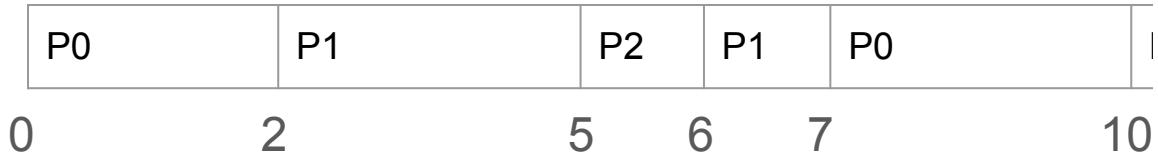


Preemptive Priority (PP)

Process	CPU time	Arrival	Priority
P0	~~5~~ 3	0	30
P1	~~4~~ ~~1~~	2	35
P2	~~1~~	5	36
P3	6	6	20

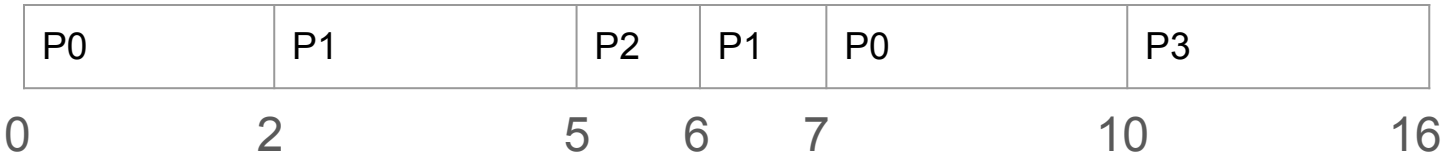
Preemptive Priority (PP)

Process	CPU time	Arrival	Priority
P0	~~5~~ ~~3~~	0	30
P1	~~4~~ ~~1~~	2	35
P2	~~1~~	5	36
P3	6	6	20



Preemptive Priority (PP)

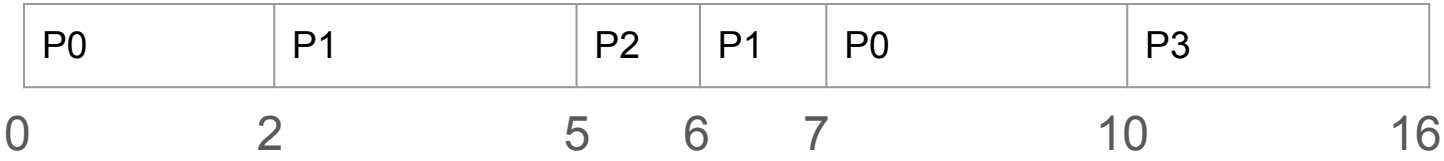
Process	CPU time	Arrival	Priority
P0	~~5~~ ~~3~~	0	30
P1	~~4~~ ~~1~~	2	35
P2	~~1~~	5	36
P3	6	6	20



Preemptive Priority (PP)

Process	CPU time	Arrival	Priority
P0	5	0	30
P1	4	2	35
P2	1	5	36
P3	6	6	20

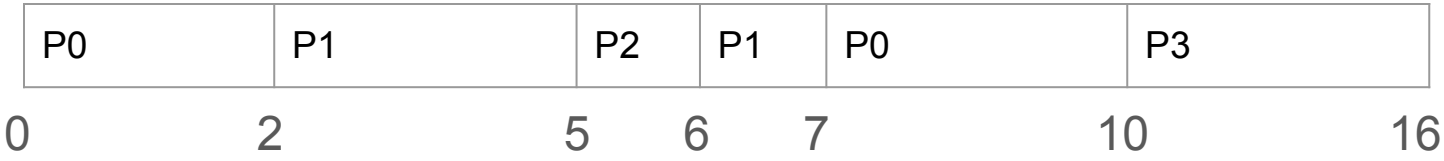
	Wait-time	Turnaround-time
P0	(0-0)+(7-2)=5	
P1		
P2		
P3		



Preemptive Priority (PP)

Process	CPU time	Arrival	Priority
P0	5	0	30
P1	4	2	35
P2	1	5	36
P3	6	6	20

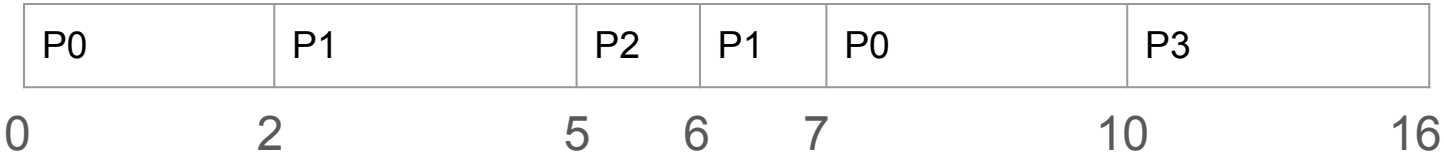
	Wait-time	Turnaround-time
P0	(0-0)+(7-2)=5	
P1	(2-2)+(6-5)=1	
P2		
P3		



Preemptive Priority (PP)

Process	CPU time	Arrival	Priority
P0	5	0	30
P1	4	2	35
P2	1	5	36
P3	6	6	20

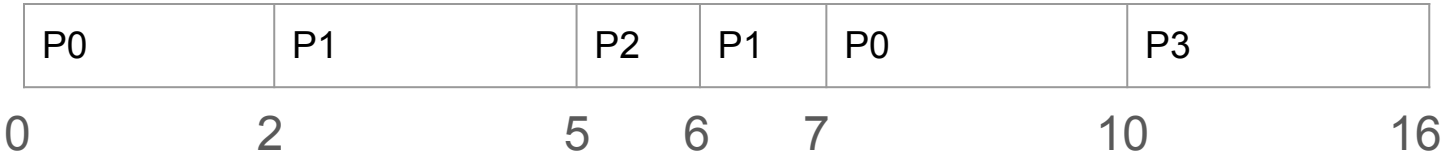
	Wait-time	Turnaround-time
P0	(0-0)+(7-2)=5	
P1	(2-2)+(6-5)=1	
P2	(5-5)=0	
P3		



Preemptive Priority (PP)

Process	CPU time	Arrival	Priority
P0	5	0	30
P1	4	2	35
P2	1	5	36
P3	6	6	20

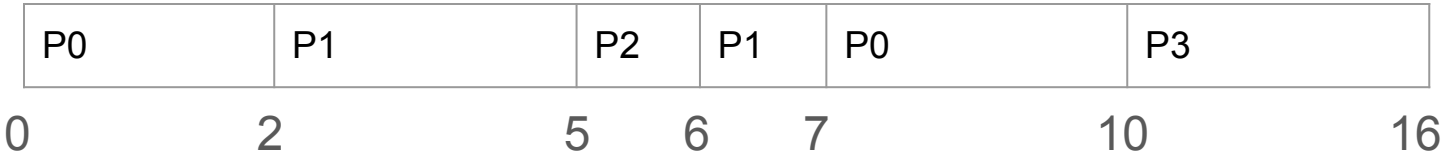
	Wait-time	Turnaround-time
P0	(0-0)+(7-2)=5	
P1	(2-2)+(6-5)=1	
P2	(5-5)=0	
P3	(10-6)=4	



Preemptive Priority (PP)

Process	CPU time	Arrival	Priority
P0	5	0	30
P1	4	2	35
P2	1	5	36
P3	6	6	20

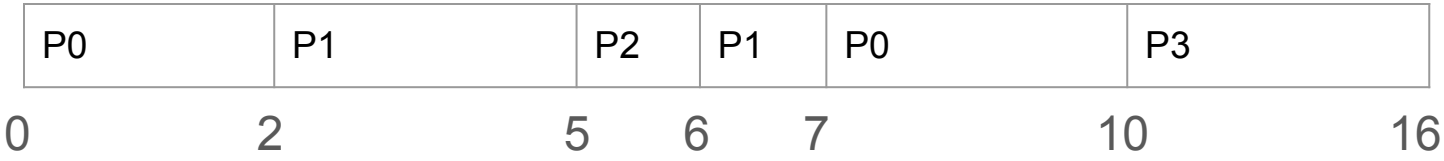
	Wait-time	Turnaround-time
P0	(0-0)+(7-2)=5	5 + 5 = 10
P1	(2-2)+(6-5)=1	
P2	(5-5)=0	
P3	(10-6)=4	



Preemptive Priority (PP)

Process	CPU time	Arrival	Priority
P0	5	0	30
P1	4	2	35
P2	1	5	36
P3	6	6	20

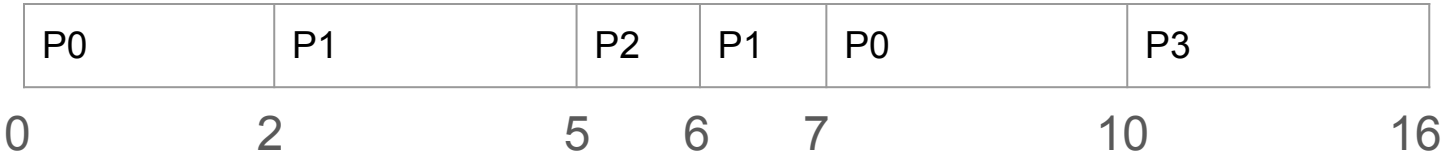
	Wait-time	Turnaround-time
P0	(0-0)+(7-2)=5	5 + 5 = 10
P1	(2-2)+(6-5)=1	1 + 4 = 5
P2	(5-5)=0	
P3	(10-6)=4	



Preemptive Priority (PP)

Process	CPU time	Arrival	Priority
P0	5	0	30
P1	4	2	35
P2	1	5	36
P3	6	6	20

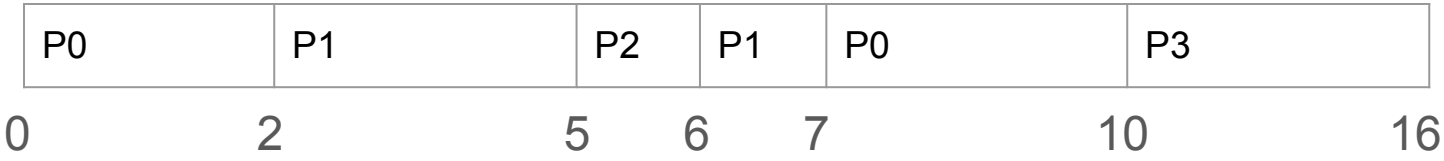
	Wait-time	Turnaround-time
P0	(0-0)+(7-2)=5	5 + 5 = 10
P1	(2-2)+(6-5)=1	1 + 4 = 5
P2	(5-5)=0	0 + 1 = 1
P3	(10-6)=4	



Preemptive Priority (PP)

Process	CPU time	Arrival	Priority
P0	5	0	30
P1	4	2	35
P2	1	5	36
P3	6	6	20

	Wait-time	Turnaround-time
P0	(0-0)+(7-2)=5	5 + 5 = 10
P1	(2-2)+(6-5)=1	1 + 4 = 5
P2	(5-5)=0	0 + 1 = 1
P3	(10-6)=4	4 + 6 = 10



Preemptive Priority (PP) Evaluation

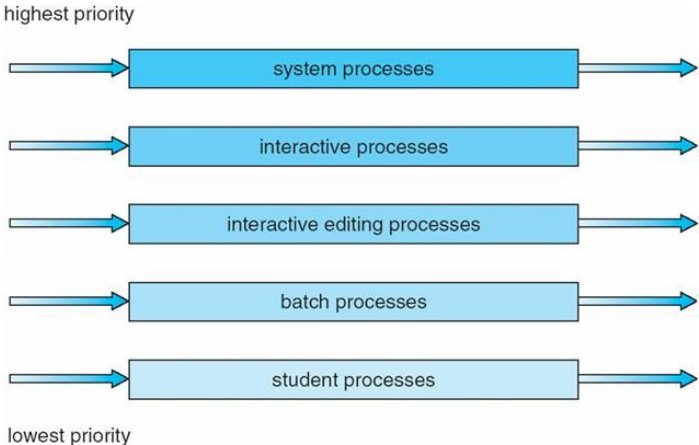
- preemptive **(at arrival of process into ready queue)**
- fairness — penalizes low priority processes
- starvation — possible
- overhead — minimal

	Wait-time	**Turnaround-time**
P0	(0-0)+(7-2)=5	5 + 5 = 10
P1	(2-2)+(6-5)=1	1 + 4 = 5
P2	(5-5)=0	0 + 1 = 1
P3	(10-6)=4	4 + 6 = 10
AVG	2.25	6.5

Multilevel Queues

Multilevel Queue

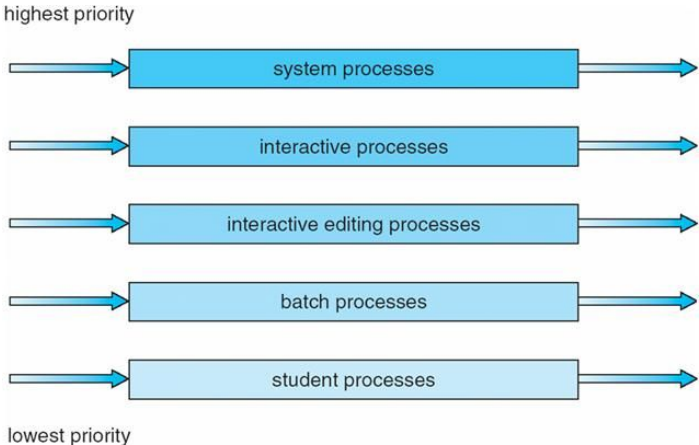
- use several ready queues, and associate a different priority with each queue
- schedule either
  - the highest priority occupied queue. problem: processes at low-level queues may starve
  - each queue gets a certain amount of CPU time.
- assign new processes permanently to a particular queue
  - foreground, background system, interactive, editing, computing.
- each queue has its own scheduling discipline
  - example: two queues:
    - foreground using RR
    - background using FCFS



Multilevel Queue

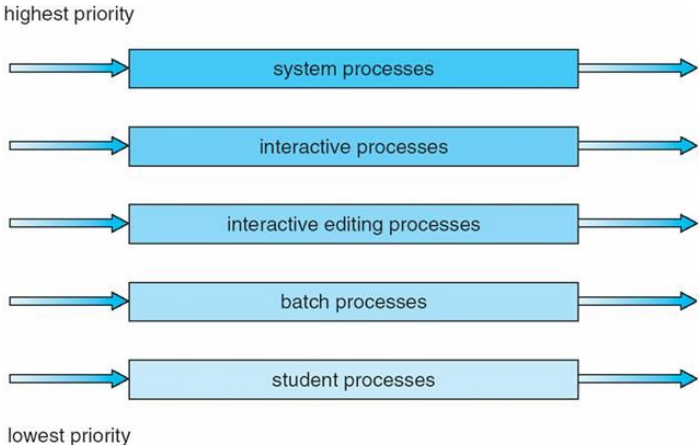
- use several ready queues, and associate a different priority with each queue
- schedule either
  - the highest priority occupied queue. problem: processes at low-level queues may starve
  - each queue gets a certain amount of CPU time.
- assign new processes permanently to a particular queue
  - foreground, background system, interactive, editing, computing.
- each queue has its own scheduling discipline
  - example: two queues:
    - foreground using RR
    - background using FCFS

What would be a possible problem?



Multilevel Queue

- use several ready queues, and associate a different priority with each queue
- schedule either
  - the highest priority occupied queue. problem: processes at low-level queues may starve
  - each queue gets a certain amount of CPU time.
- assign new processes permanently to a particular queue
  - foreground, background system, interactive, editing, computing.
- each queue has its own scheduling discipline
  - example: two queues:
    - foreground using RR
    - background using FCFS

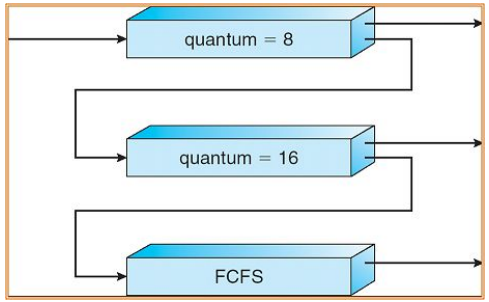


What would be a possible problem? Fairness

Multilevel Feedback Queue

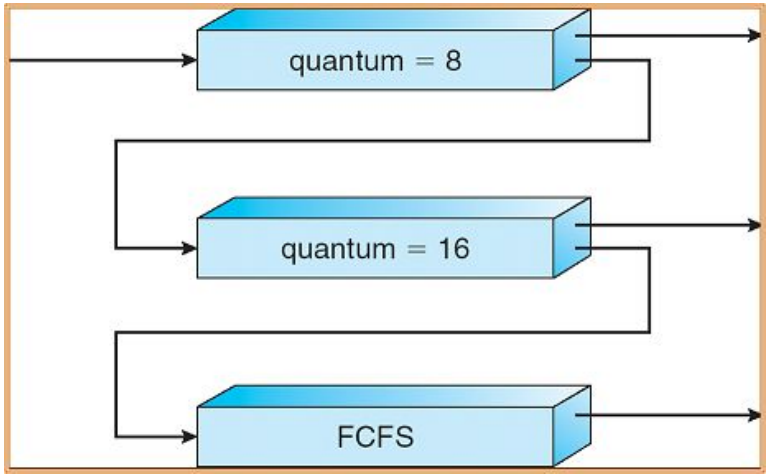
- Feedback – allow scheduler to move processes between queues to ensure fairness
  - aging – moving older processes to higher-priority queue
  - decay – moving older processes to lower-priority queue

- Example: three queues, feedback with process decay
  - Q0 – RR with time slice of 8 milliseconds
  - Q1 – RR with time slice of 16 milliseconds
  - Q2 – FCFS



Multilevel Feedback Queue Example

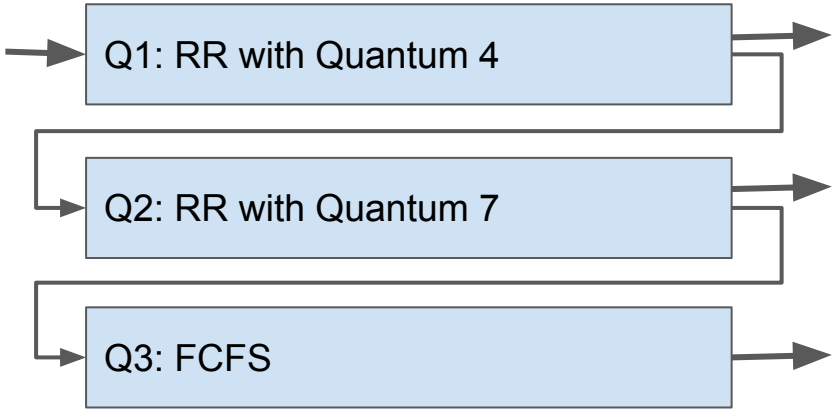
- new job enters queue **Q0**
- job receives 8 milliseconds.
- if it does not finish in 8 milliseconds, job is moved to queue **Q1**.
- Q1 job is again served RR and receives 16 additional milliseconds.
- if it still does not complete, it is preempted and moved to queue **Q2**.



- Q0 – RR with time slice of 8 milliseconds
- Q1 – RR with time slice of 16 milliseconds
- Q2 – FCFS

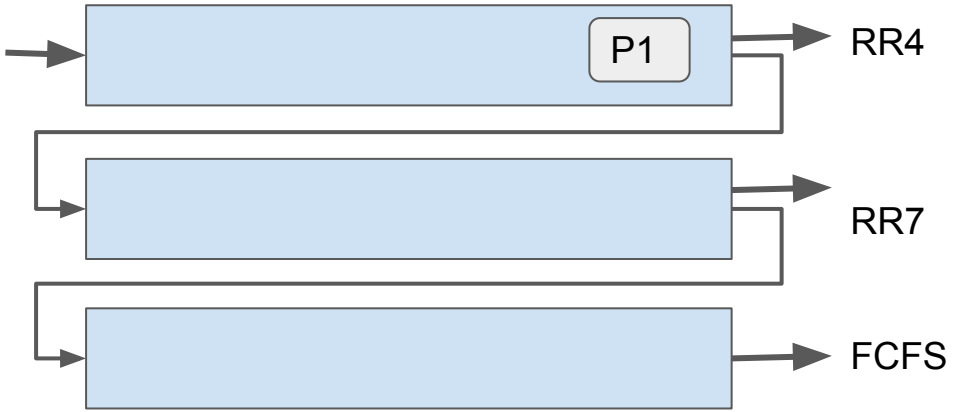
Multilevel Feedback Queue (MLFQ)

Process	CPU,I/O,CPU	Arrival	Queue
P1	5, 6, 7	0	1
P2	4, 2, 3	3	1
P3	2, 3, 4	4	1
P4	5, 2, 7	7	1



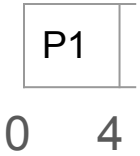
Multilevel Feedback Queue (MLFQ)

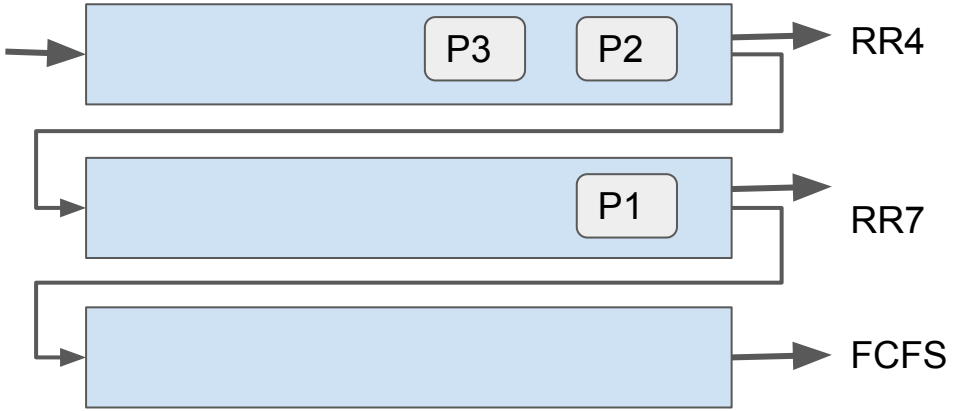
Process	CPU,I/O,CPU	Arrival	Queue
P1	5, 6, 7	0	1
P2	4, 2, 3	3	1
P3	2, 3, 4	4	1
P4	5, 2, 7	7	1



Multilevel Feedback Queue (MLFQ)

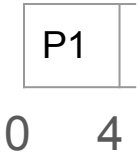
Process	CPU,I/O,CPU	Arrival	Queue
P1	5, 6, 7	0	1
P2	4, 2, 3	3	1
P3	2, 3, 4	4	1
P4	5, 2, 7	7	1

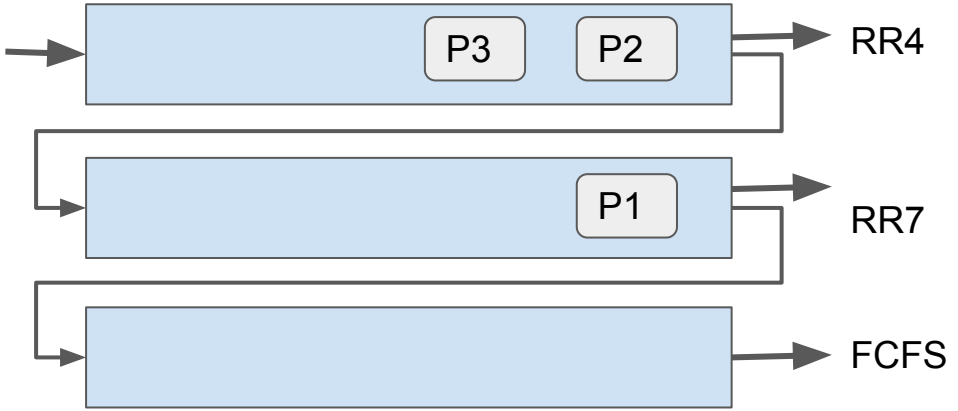




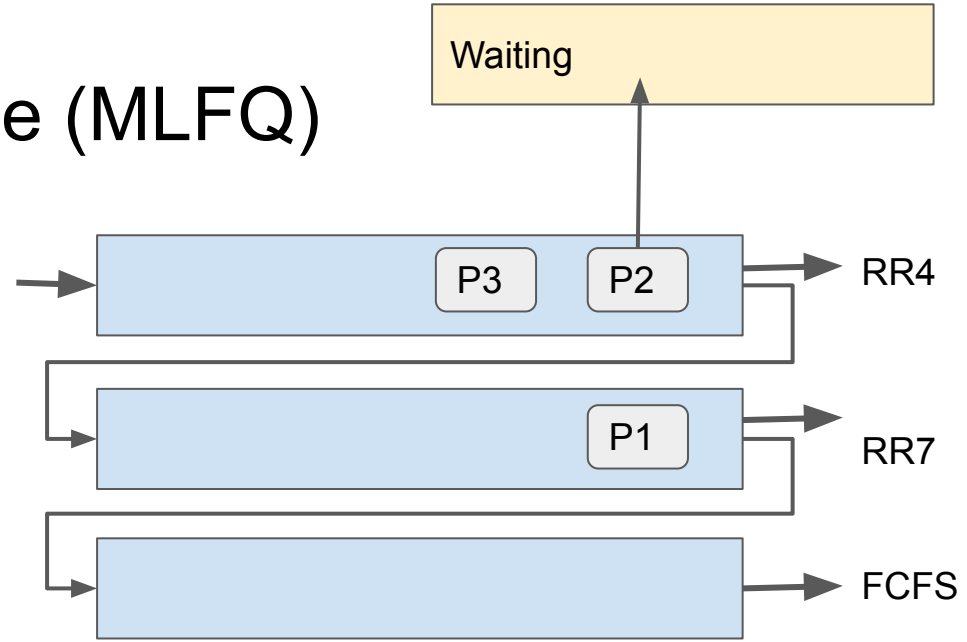
Multilevel Feedback Queue (MLFQ)

Process	CPU,I/O,CPU	Arrival	Queue
P1	~~5~~ 1 , 6, 7	~~0~~ 4	~~1~~ 2
P2	4, 2, 3	3	1
P3	2, 3, 4	4	1
P4	5, 2, 7	7	1

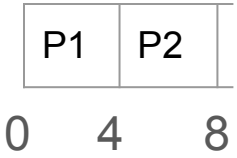




Multilevel Feedback Queue (MLFQ)



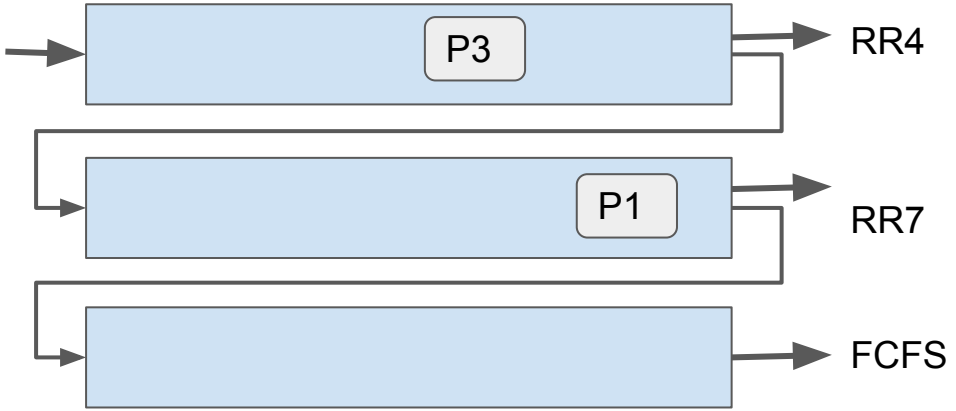
Process	CPU,I/O,CPU	Arrival	Queue
P1	~~5~~ 1 , 6, 7	~~0~~ 4	~~1~~ 2
P2	~~4~~, 2, 3	3	1
P3	2, 3, 4	4	1
P4	5, 2, 7	7	1

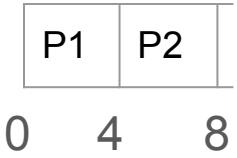


Multilevel Feedback Queue (MLFQ)

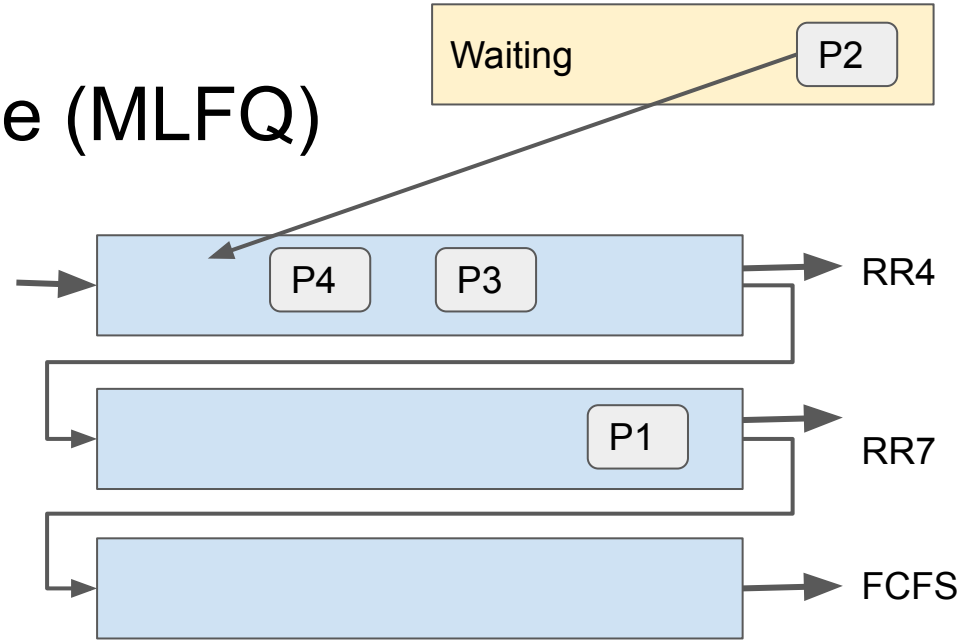
Waiting P2

Process	CPU,I/O,CPU	Arrival	Queue
P1	~~5~~ 1 , 6, 7	~~0~~ 4	~~1~~ 2
P2	~~4~~, 2, 3	3	1
P3	2, 3, 4	4	1
P4	5, 2, 7	7	1





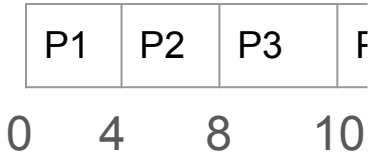
Multilevel Feedback Queue (MLFQ)

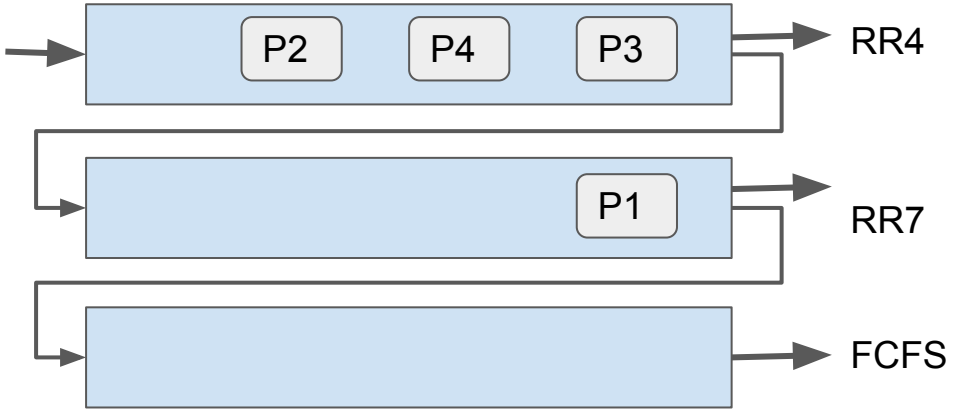


Process	CPU,I/O,CPU	Arrival	Queue
P1	~~5~~ 1 , 6, 7	~~0~~ 4	~~1~~ 2
P2	~~4~~, ~~2~~, 3	~~3~~ 10	1
P3	~~2~~, 3, 4	4	1
P4	5, 2, 7	7	1

Multilevel Feedback Queue (MLFQ)

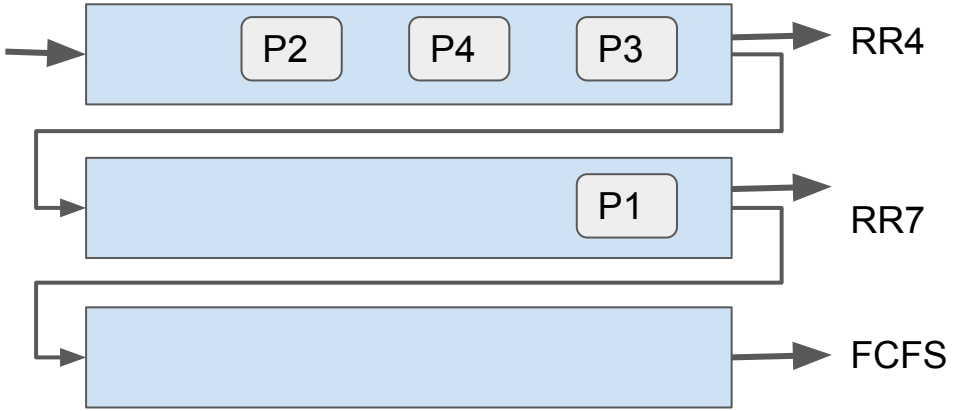
Process	CPU,I/O,CPU	Arrival	Queue
P1	~~5~~ 1 , 6, 7	~~0~~ 4	~~1~~ 2
P2	~~4~~, ~~2~~, 3	~~3~~ 10	1
P3	~~2~~, 3, 4	4	1
P4	5, 2, 7	7	1





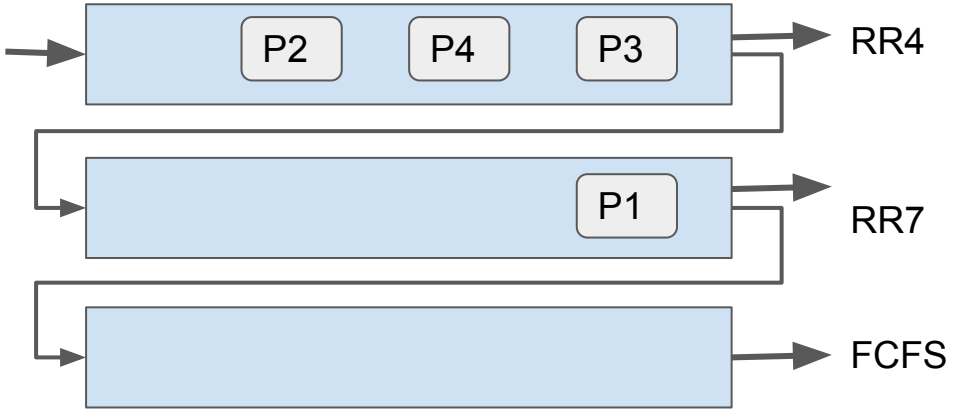
Multilevel Feedback Queue (MLFQ)

Process	CPU,I/O,CPU	Arrival	Queue
P1	~~5~~ 1 , 6, 7	~~0~~ 4	~~1~~ 2
P2	~~4~~, ~~2~~, 3	~~3~~ 10	1
P3	~~2~~, ~~3~~, 4	4	1
P4	5, 2, 7	7	1



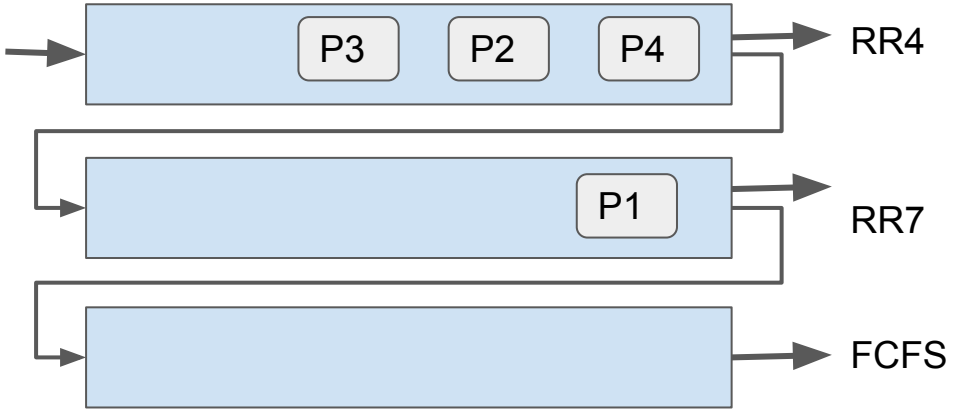
Multilevel Feedback Queue (MLFQ)

Process	CPU,I/O,CPU	Arrival	Queue
P1	~~5~~ 1 , 6, 7	~~0~~ 4	~~1~~ 2
P2	~~4~~, ~~2~~, 3	~~3~~ 10	1
P3	~~2~~, ~~3~~, 4	~~4~~ 13	1
P4	5, 2, 7	7	1



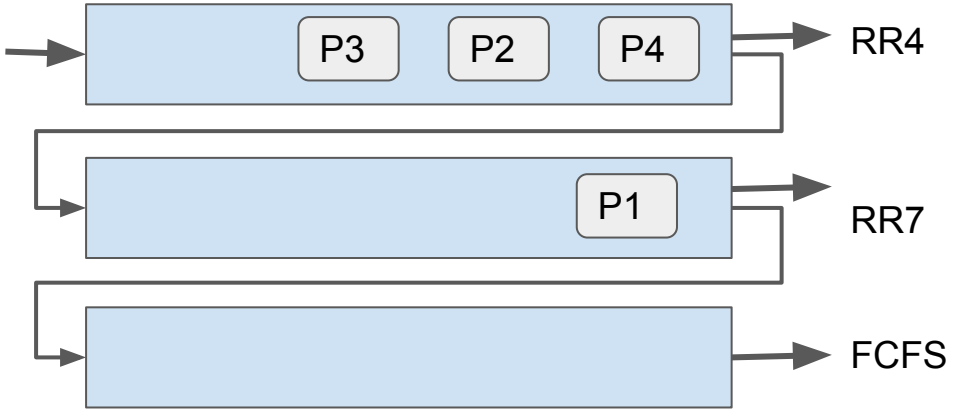
Multilevel Feedback Queue (MLFQ)

Process	CPU,I/O,CPU	Arrival	Queue
P1	~~5~~ 1 , 6, 7	~~0~~ 4	~~1~~ 2
P2	~~4~~, ~~2~~, 3	~~3~~ 10	1
P3	~~2~~, ~~3~~, 4	~~4~~ 13	1
P4	5, 2, 7	7	1



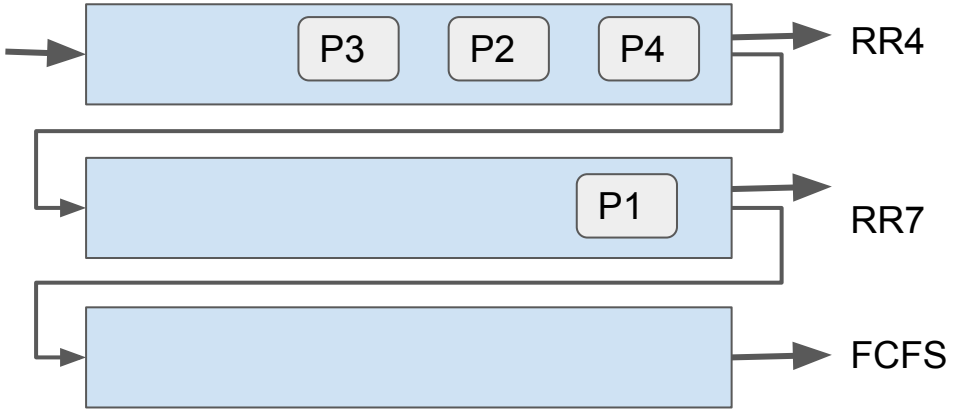
Multilevel Feedback Queue (MLFQ)

Process	CPU,I/O,CPU	Arrival	Queue
P1	~~5~~ 1 , 6, 7	~~0~~ 4	~~1~~ 2
P2	~~4~~, ~~2~~, 3	~~3~~ 10	1
P3	~~2~~, ~~3~~, 4	~~4~~ 13	1
P4	5, 2, 7	7	1



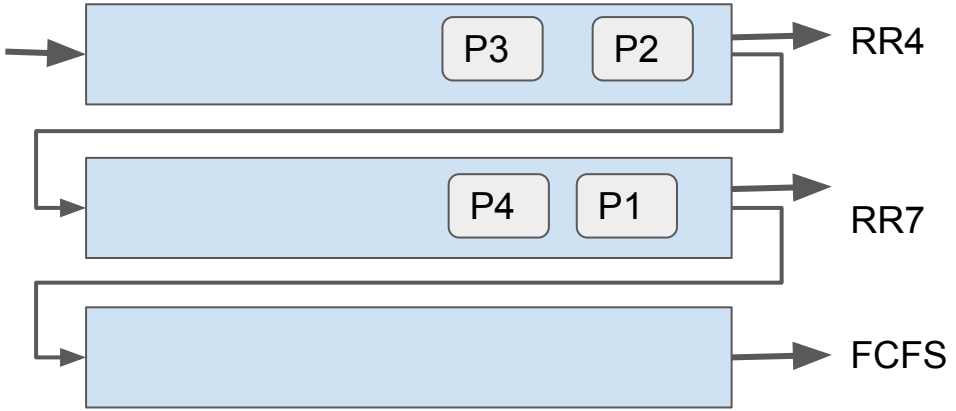
Multilevel Feedback Queue (MLFQ)

Process	CPU,I/O,CPU	Arrival	Queue
P1	~~5~~ 1 , 6, 7	~~0~~ 4	~~1~~ 2
P2	~~4~~, ~~2~~, 3	~~3~~ 10	1
P3	~~2~~, ~~3~~, 4	~~4~~ 13	1
P4	~~5~~ 1 , 2, 7	~~7~~ 14	~~1~~ 2



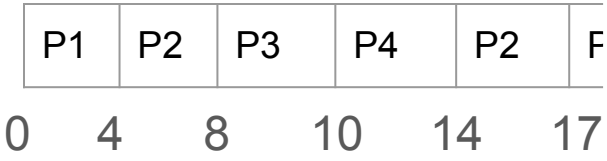
Multilevel Feedback Queue (MLFQ)

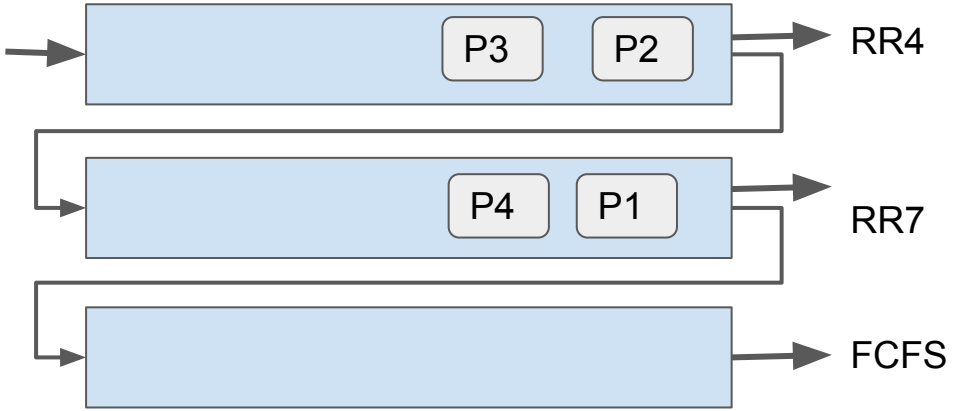
Process	CPU,I/O,CPU	Arrival	Queue
P1	~~5~~ 1 , 6, 7	~~0~~ 4	~~1~~ 2
P2	~~4~~, ~~2~~, 3	~~3~~ 10	1
P3	~~2~~, ~~3~~, 4	~~4~~ 13	1
P4	~~5~~ 1 , 2, 7	~~7~~ 14	~~1~~ 2



Multilevel Feedback Queue (MLFQ)

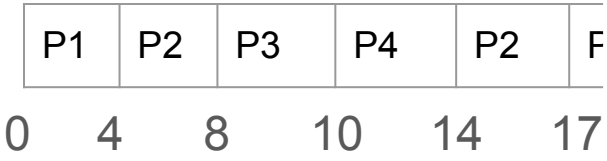
Process	CPU,I/O,CPU	Arrival	Queue
P1	~~5~~ 1 , 6, 7	~~0~~ 4	~~1~~ 2
P2	~~4~~, ~~2~~, 3	~~3~~ 10	1
P3	~~2~~, ~~3~~, 4	~~4~~ 13	1
P4	~~5~~ 1 , 2, 7	~~7~~ 14	~~1~~ 2

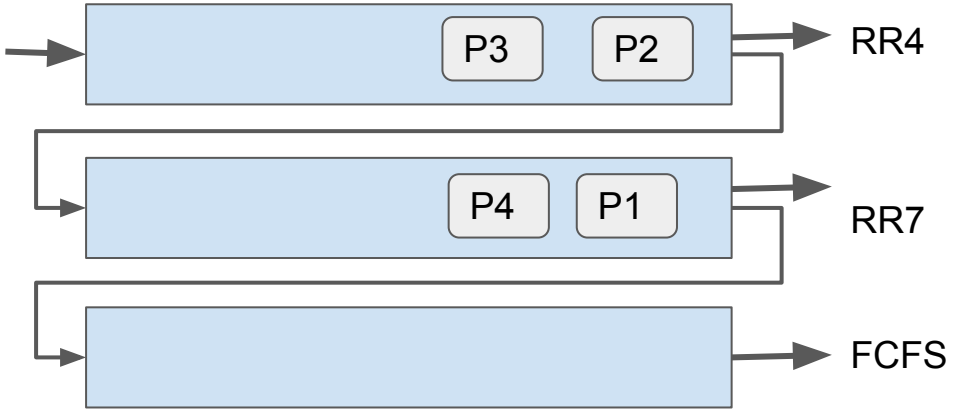




Multilevel Feedback Queue (MLFQ)

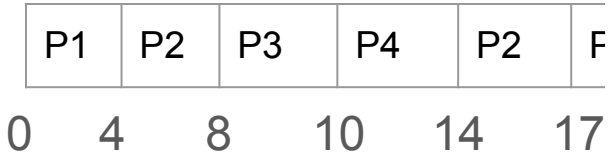
Process	CPU,I/O,CPU	Arrival	Queue
P1	~~5~~ 1 , 6, 7	~~0~~ 4	~~1~~ 2
P2	~~4~~, ~~2~~, ~~3~~	~~3~~ 10	1
P3	~~2~~, ~~3~~, 4	~~4~~ 13	1
P4	~~5~~ 1 , 2, 7	~~7~~ 14	~~1~~ 2

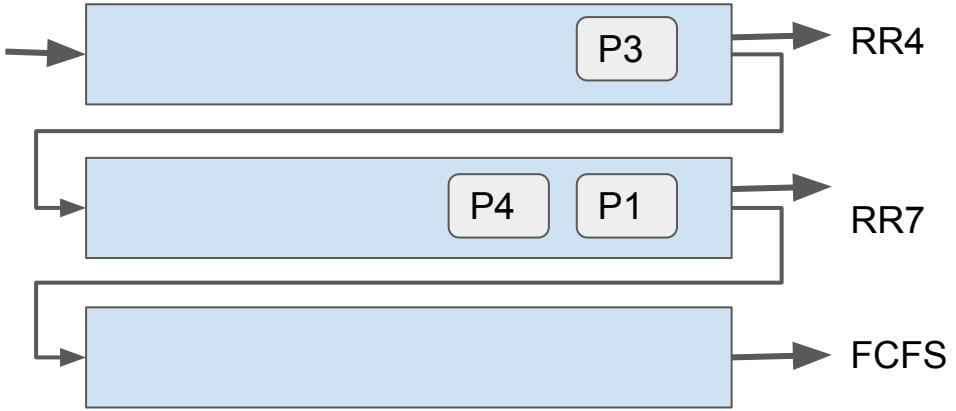




Multilevel Feedback Queue (MLFQ)

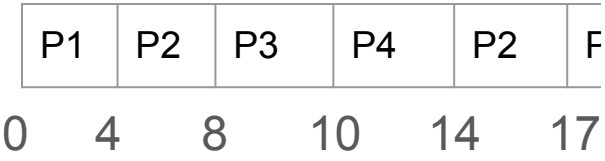
Process	CPU,I/O,CPU	Arrival	Queue
P1	~~5~~ 1 , 6, 7	~~0~~ 4	~~1~~ 2
P2	~~4~~, ~~2~~, ~~3~~	~~3~~ 10	1
P3	~~2~~, ~~3~~, 4	~~4~~ 13	1
P4	~~5~~ 1 , 2, 7	~~7~~ 14	~~1~~ 2

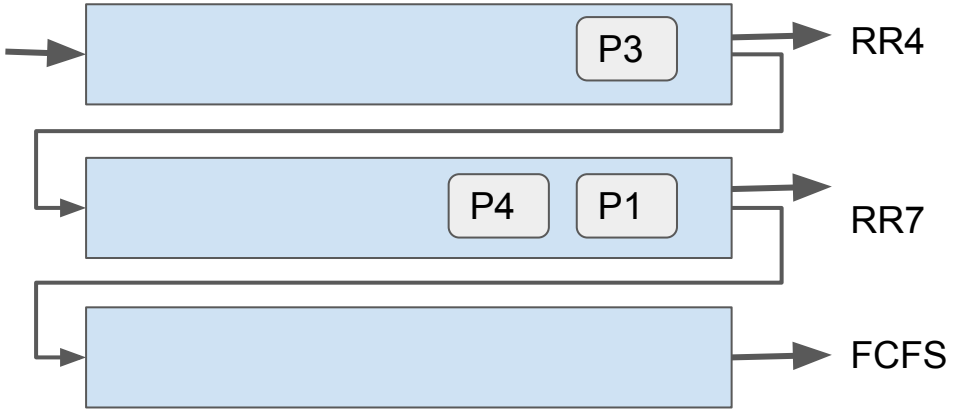




Multilevel Feedback Queue (MLFQ)

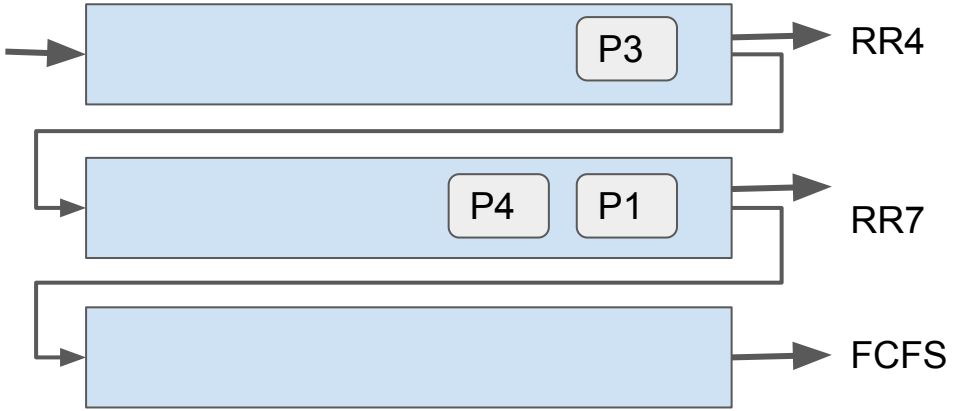
Process	CPU,I/O,CPU	Arrival	Queue
P1	~~5~~ 1 , 6, 7	~~0~~ 4	~~1~~ 2
~~P2~~	~~4, 2, 3~~	~~3~~ 10	~~1~~
P3	~~2~~, ~~3~~, 4	~~4~~ 13	1
P4	~~5~~ 1 , 2, 7	~~7~~ 14	~~1~~ 2





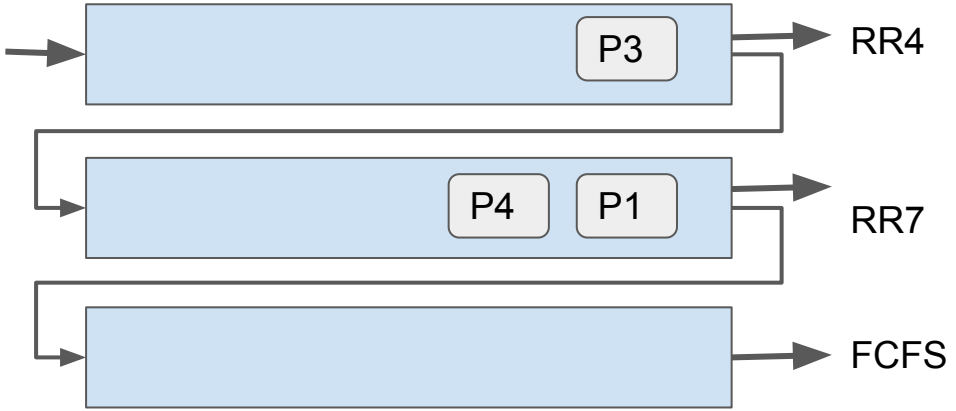
Multilevel Feedback Queue (MLFQ)

Process	CPU,I/O,CPU	Arrival	Queue
P1	~~5~~ 1 , 6, 7	~~0~~ 4	~~1~~ 2
~~P2~~	~~4, 2, 3~~	~~3~~ 10	~~1~~
P3	~~2~~, ~~3~~, 4	~~4~~ 13	1
P4	~~5~~ 1 , 2, 7	~~7~~ 14	~~1~~ 2



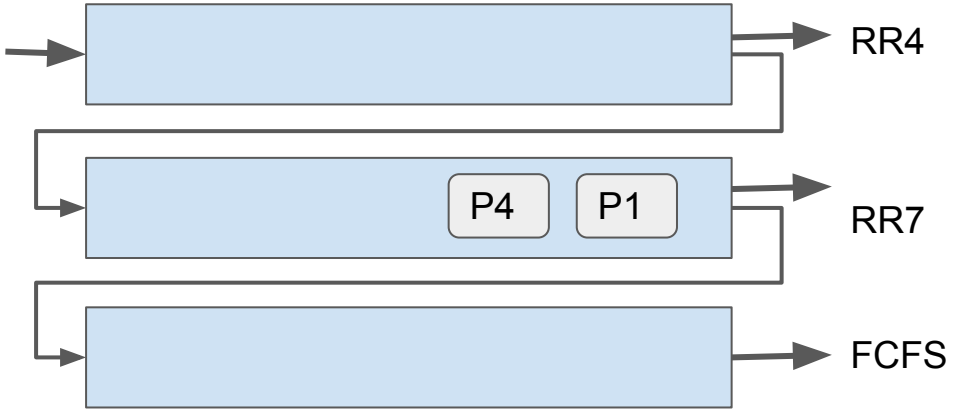
Multilevel Feedback Queue (MLFQ)

Process	CPU,I/O,CPU	Arrival	Queue
P1	~~5~~ 1 , 6, 7	~~0~~ 4	~~1~~ 2
~~P2~~	~~4, 2, 3~~	~~3~~ 10	~~1~~
P3	~~2, 3, 4~~	~~4~~ 13	1
P4	~~5~~ 1 , 2, 7	~~7~~ 14	~~1~~ 2



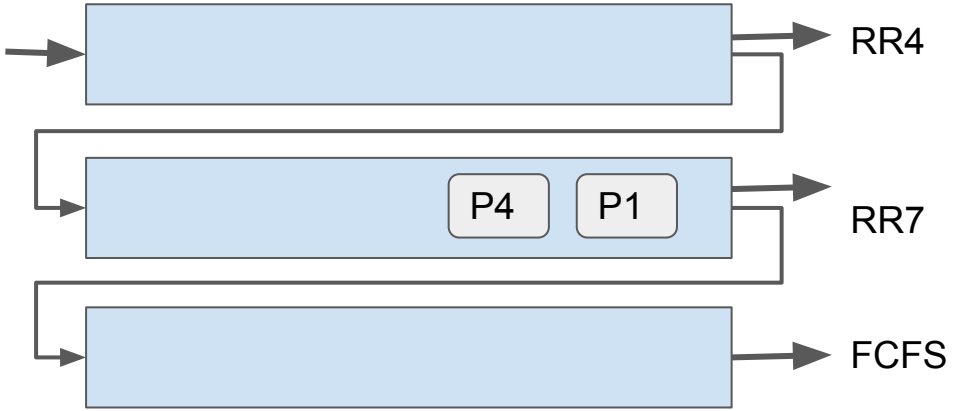
Multilevel Feedback Queue (MLFQ)

Process	CPU,I/O,CPU	Arrival	Queue
P1	~~5~~ 1 , 6, 7	~~0~~ 4	~~1~~ 2
~~P2~~	~~4, 2, 3~~	~~3~~ 10	1
~~P3~~	~~2, 3, 4~~	~~4~~ 13	1
P4	~~5~~ 1 , 2, 7	~~7~~ 14	~~1~~ 2



Multilevel Feedback Queue (MLFQ)

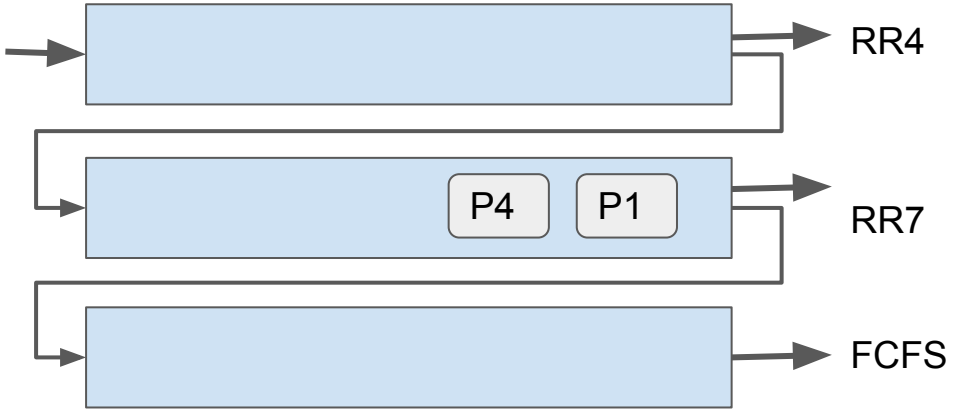
Process	CPU,I/O,CPU	Arrival	Queue
P1	~~5~~ ~~1~~ , 6, 7	~~0~~ 4	~~1~~ 2
~~P2~~	~~4, 2, 3~~	~~3~~ 10	1
~~P3~~	~~2, 3, 4~~	~~4~~ 13	1
P4	~~5~~ 1 , 2, 7	~~7~~ 14	~~1~~ 2



Multilevel Feedback Queue (MLFQ)

Waiting

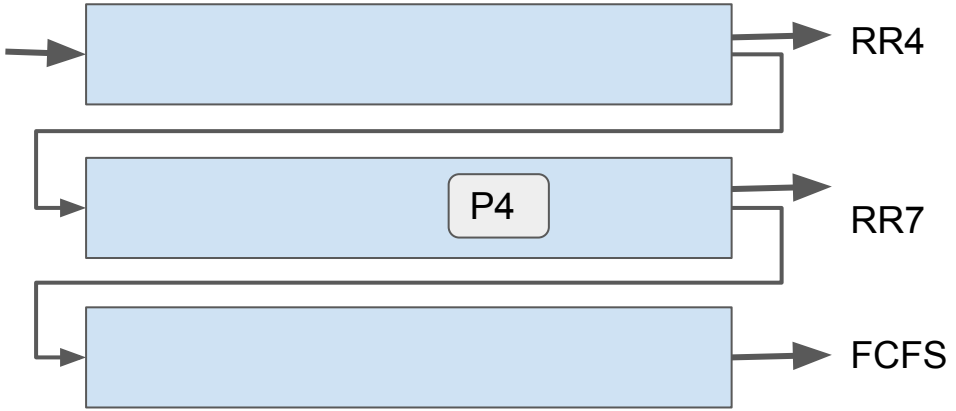
Process	CPU,I/O,CPU	Arrival	Queue
P1	~~5~~ ~~1~~ , 6, 7	~~0~~ 4	~~1~~ 2
~~P2~~	~~4, 2, 3~~	~~3~~ 10	1
~~P3~~	~~2, 3, 4~~	~~4~~ 13	1
P4	~~5~~ 1 , 2, 7	~~7~~ 14	~~1~~ 2



Multilevel Feedback Queue (MLFQ)

Waiting P1

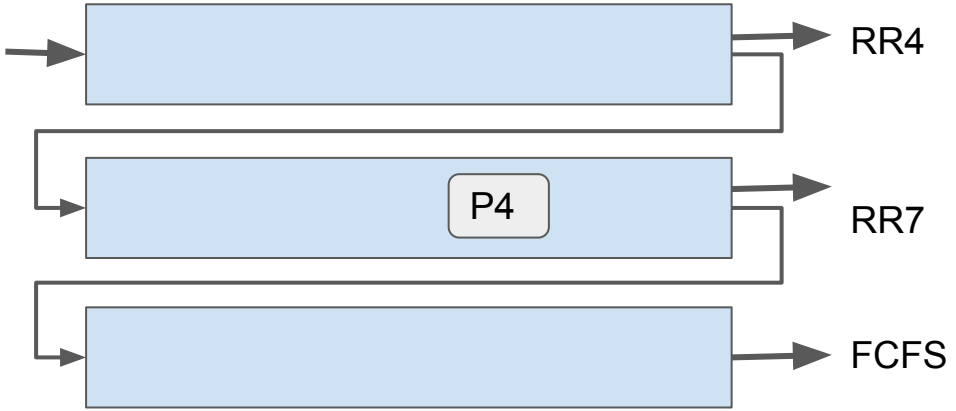
Process	CPU,I/O,CPU	Arrival	Queue
P1	~~5~~ ~~1~~ , 6, 7	~~0~~ 4	~~1~~ 2
~~P2~~	~~4, 2, 3~~	~~3~~ 10	1
~~P3~~	~~2, 3, 4~~	~~4~~ 13	1
P4	~~5~~ 1 , 2, 7	~~7~~ 14	~~1~~ 2

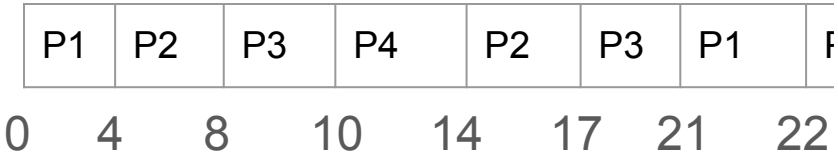


Multilevel Feedback Queue (MLFQ)

Waiting P1

Process	CPU,I/O,CPU	Arrival	Queue
P1	~~5~~ ~~1~~ , 6, 7	~~0~~ 4	~~1~~ 2
~~P2~~	~~4, 2, 3~~	~~3~~ 10	~~1~~
~~P3~~	~~2, 3, 4~~	~~4~~ 13	~~1~~
P4	~~5~~ 1 , 2, 7	~~7~~ 14	~~1~~ 2

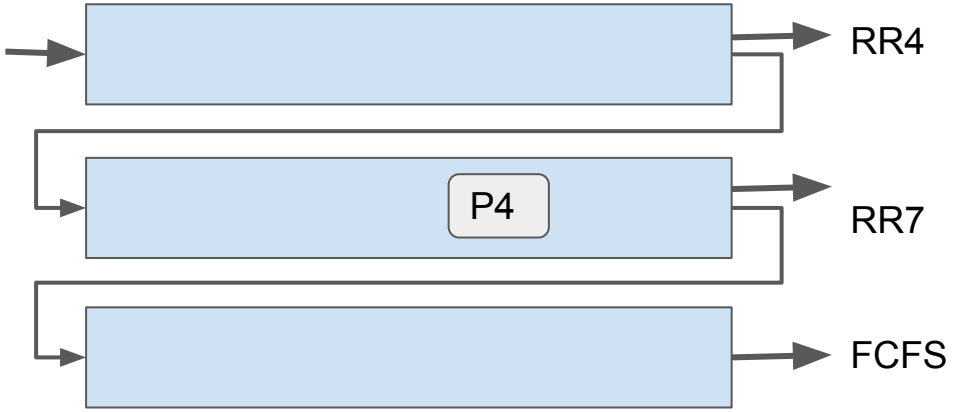




Multilevel Feedback Queue (MLFQ)

Waiting P1

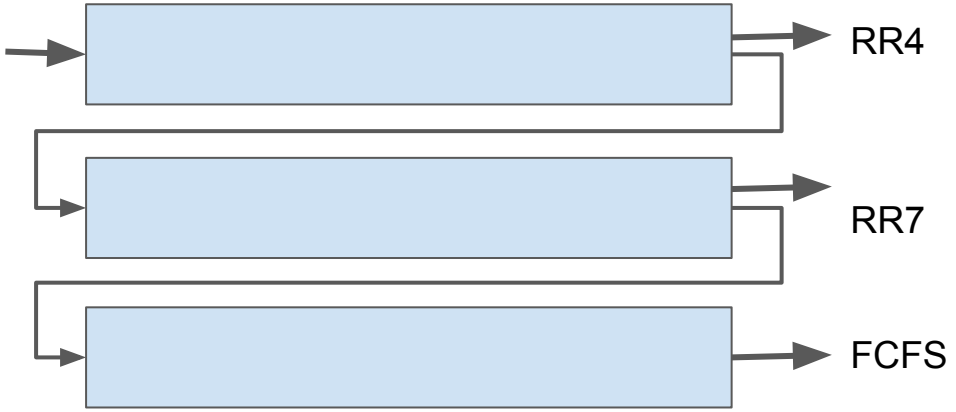
Process	CPU,I/O,CPU	Arrival	Queue
P1	~~5~~ ~~1~~ , 6, 7	~~0~~ 4	~~1~~ 2
~~P2~~	~~4, 2, 3~~	~~3~~ 10	~~1~~
~~P3~~	~~2, 3, 4~~	~~4~~ 13	~~1~~
P4	~~5~~ ~~1~~ , 2, 7	~~7~~ 14	~~1~~ 2



Multilevel Feedback Queue (MLFQ)

Waiting P4 P1

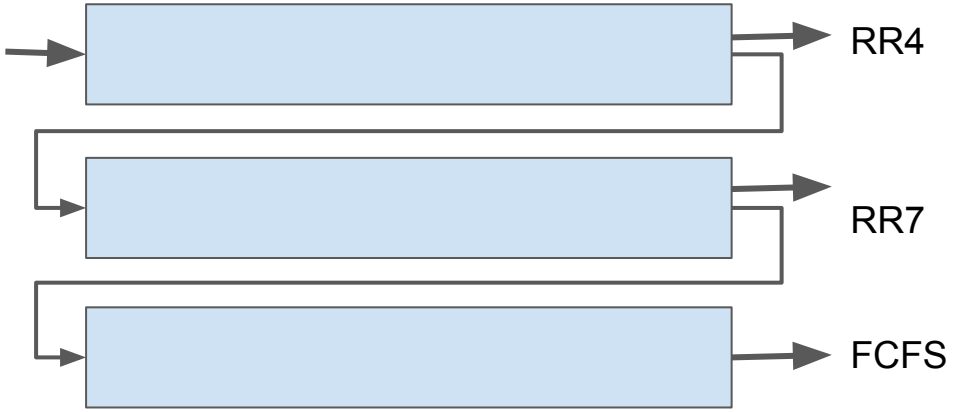
Process	CPU,I/O,CPU	Arrival	Queue
P1	~~5~~ ~~1~~ , 6, 7	~~0~~ 4	~~1~~ 2
~~P2~~	~~4, 2, 3~~	~~3~~ 10	1
~~P3~~	~~2, 3, 4~~	~~4~~ 13	1
P4	~~5~~ ~~1~~ , 2, 7	~~7~~ 14	~~1~~ 2



Multilevel Feedback Queue (MLFQ)

Waiting P4 P1

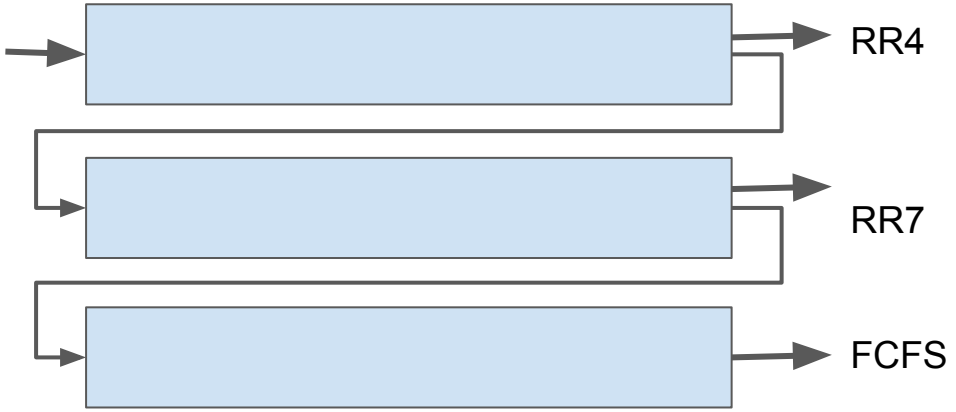
Process	CPU,I/O,CPU	Arrival	Queue
P1	~~5~~ ~~1~~ , 6, 7	~~0~~ 4	~~1~~ 2
~~P2~~	~~4, 2, 3~~	~~3~~ 10	1
~~P3~~	~~2, 3, 4~~	~~4~~ 13	1
P4	~~5~~ ~~1~~ , 2, 7	~~7~~ 14	~~1~~ 2



Multilevel Feedback Queue (MLFQ)

Waiting P4 P1

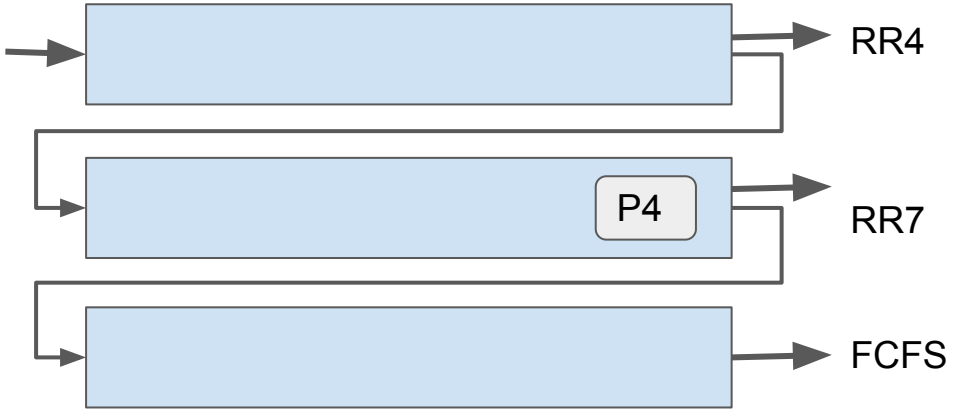
Process	CPU,I/O,CPU	Arrival	Queue
P1	~~5~~ ~~1~~ , 6, 7	~~0~~ 4	~~1~~ 2
~~P2~~	~~4, 2, 3~~	~~3~~ 10	1
~~P3~~	~~2, 3, 4~~	~~4~~ 13	1
P4	~~5~~ ~~1~~ , ~~2~~, 7	~~14~~ 25	~~1~~ 2



Multilevel Feedback Queue (MLFQ)

Waiting P1

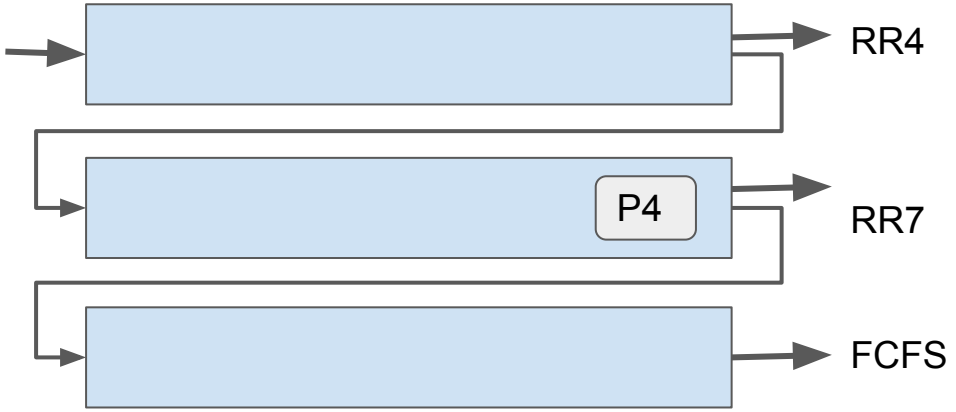
Process	CPU,I/O,CPU	Arrival	Queue
P1	~~5~~ ~~1~~ , 6, 7	~~0~~ 4	~~1~~ 2
~~P2~~	~~4, 2, 3~~	~~3~~ 10	~~1~~
~~P3~~	~~2, 3, 4~~	~~4~~ 13	~~1~~
P4	~~5~~ ~~1~~ , ~~2~~, 7	~~14~~ 25	~~1~~ 2



Multilevel Feedback Queue (MLFQ)

Waiting P1

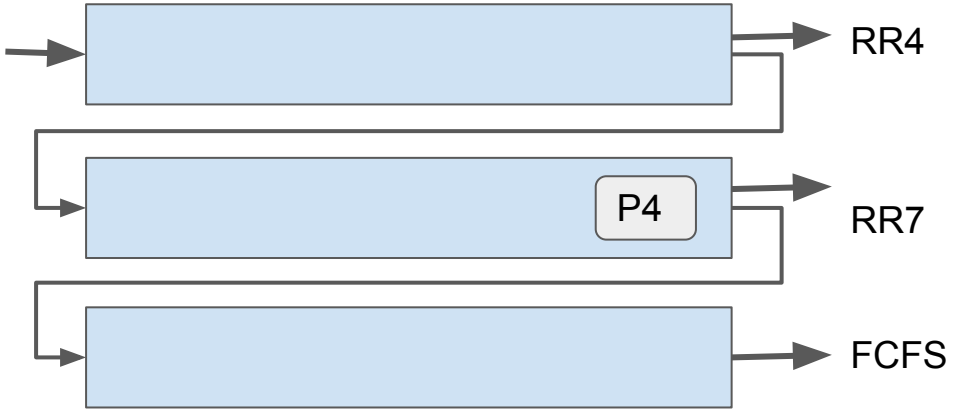
Process	CPU,I/O,CPU	Arrival	Queue
P1	~~5~~ ~~1~~ , 6, 7	~~0~~ 4	~~1~~ 2
~~P2~~	~~4, 2, 3~~	~~3~~ 10	~~1~~
~~P3~~	~~2, 3, 4~~	~~4~~ 13	~~1~~
P4	~~5~~ ~~1~~ , ~~2~~, ~~7~~	~~14~~ 25	~~1~~ 2



Multilevel Feedback Queue (MLFQ)

Waiting P1

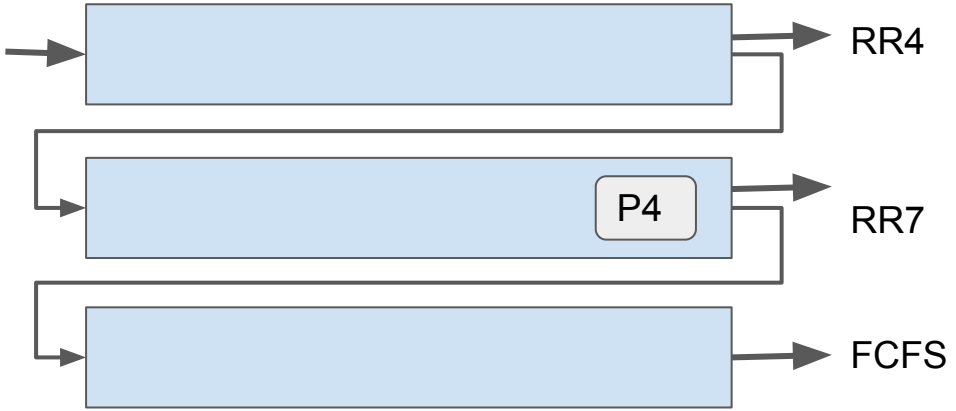
Process	CPU,I/O,CPU	Arrival	Queue
P1	~~5~~ ~~1~~ , 6, 7	~~0~~ 4	~~1~~ 2
~~P2~~	~~4, 2, 3~~	~~3~~ 10	~~1~~
~~P3~~	~~2, 3, 4~~	~~4~~ 13	~~1~~
P4	~~5~~ ~~1~~ , ~~2~~, ~~7~~	~~14~~ 25	~~1~~ 2



Multilevel Feedback Queue (MLFQ)

Waiting P1

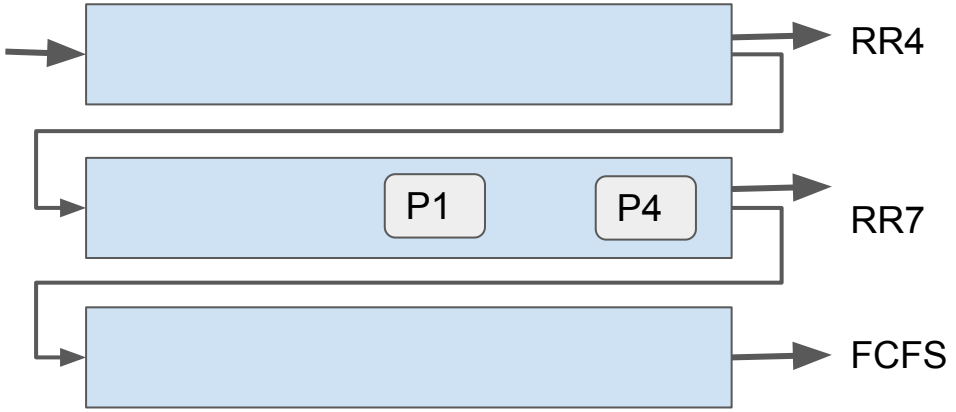
Process	CPU,I/O,CPU	Arrival	Queue
P1	~~5~~ 1 , ~~6~~, 7	~~4~~ 28	~~1~~ 2
~~P2~~	~~4, 2, 3~~	~~3~~ 10	~~1~~
~~P3~~	~~2, 3, 4~~	~~4~~ 13	~~1~~
P4	~~5~~ 1 , ~~2~~, ~~7~~	~~14~~ 25	~~1~~ 2



Multilevel Feedback Queue (MLFQ)

Waiting

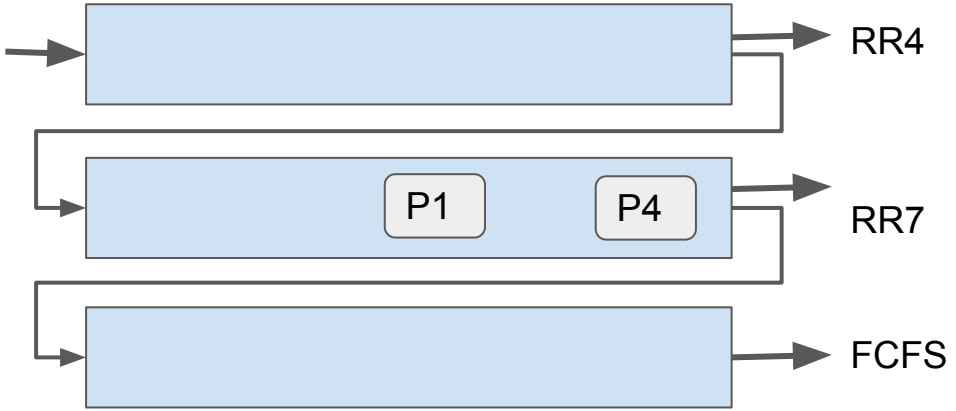
Process	CPU,I/O,CPU	Arrival	Queue
P1	~~5~~ 1 , ~~6~~, 7	~~4~~ 28	~~1~~ 2
~~P2~~	~~4, 2, 3~~	~~3~~ 10	~~1~~
~~P3~~	~~2, 3, 4~~	~~4~~ 13	~~1~~
P4	~~5~~ 1 , ~~2~~, ~~7~~	~~14~~ 25	~~1~~ 2



Multilevel Feedback Queue (MLFQ)

Waiting

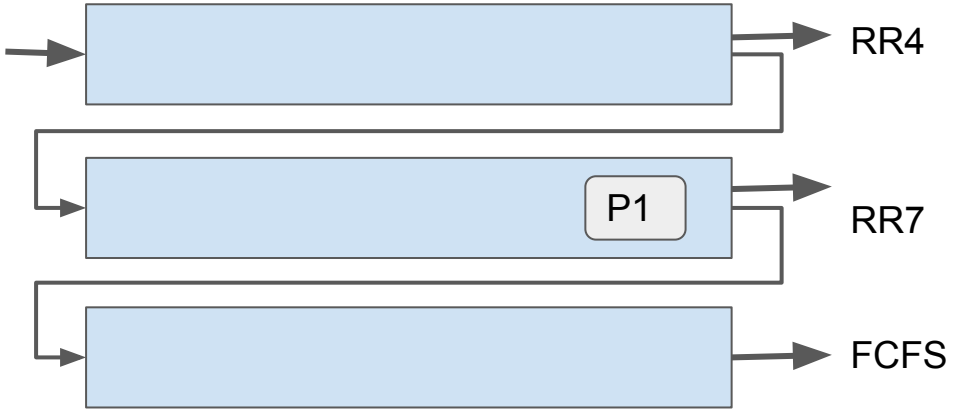
Process	CPU,I/O,CPU	Arrival	Queue
P1	5, 1, 6, 7	4 28	1 2
P2	4, 2, 3	3 10	1
P3	2, 3, 4	4 13	1
P4	5, 1, 2, 7	14 25	1 2



Multilevel Feedback Queue (MLFQ)

Waiting

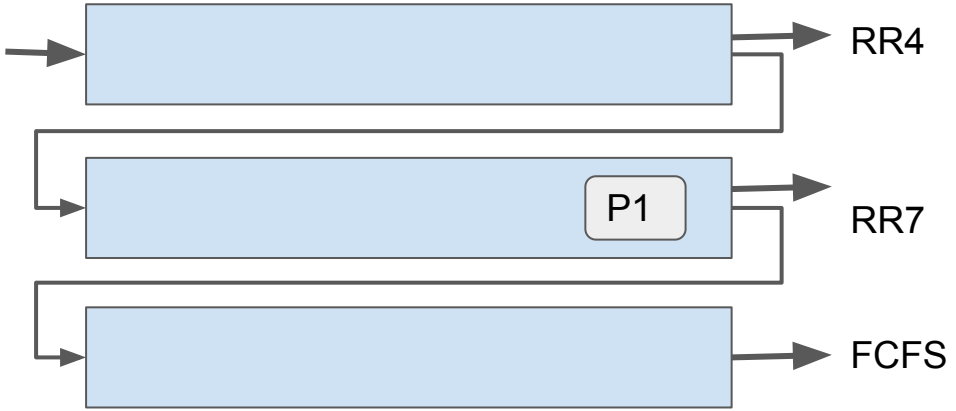
Process	CPU,I/O,CPU	Arrival	Queue
P1	~~5~~ 1 , ~~6~~, 7	~~4~~ 28	~~1~~ 2
~~P2~~	~~4, 2, 3~~	~~3~~ 10	1
~~P3~~	~~2, 3, 4~~	~~4~~ 13	1
~~P4~~	~~5~~ 1 , ~~2, 7~~	~~14~~ 25	~~1~~ 2



Multilevel Feedback Queue (MLFQ)

Waiting

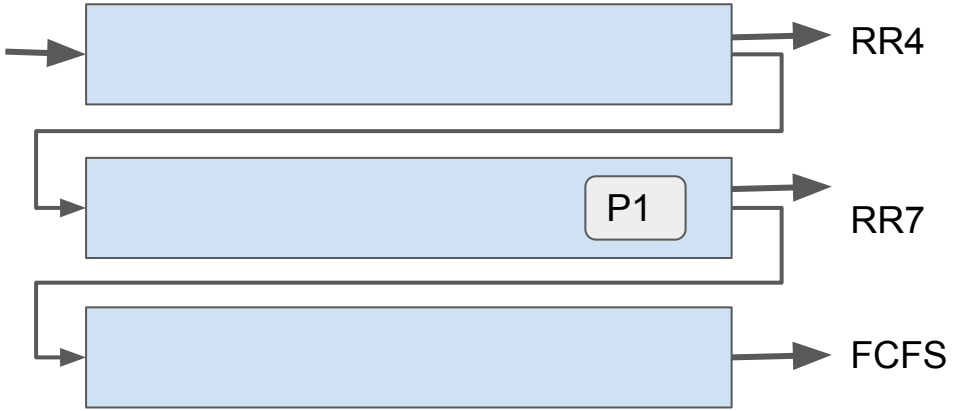
Process	CPU,I/O,CPU	Arrival	Queue
P1	5 1 , 6, 7	4 28	1 2
P2	4, 2, 3	3 10	1
P3	2, 3, 4	4 13	1
P4	5 1 , 2, 7	14 25	1 2



Multilevel Feedback Queue (MLFQ)

Waiting

Process	CPU,I/O,CPU	Arrival	Queue
P1	5, 1, 6, 7	4 28	1 2
P2	4, 2, 3	3 10	1
P3	2, 3, 4	4 13	1
P4	5, 1, 2, 7	14 25	1 2



Waiting

Multilevel Feedback Queue (MLFQ)

Process	CPU,I/O,CPU	Arrival	Queue
P1	5, 1, 6, 7	4 28	1 2
P2	4, 2, 3	3 10	1
P3	2, 3, 4	4 13	1
P4	5, 1, 2, 7	14 25	1 2