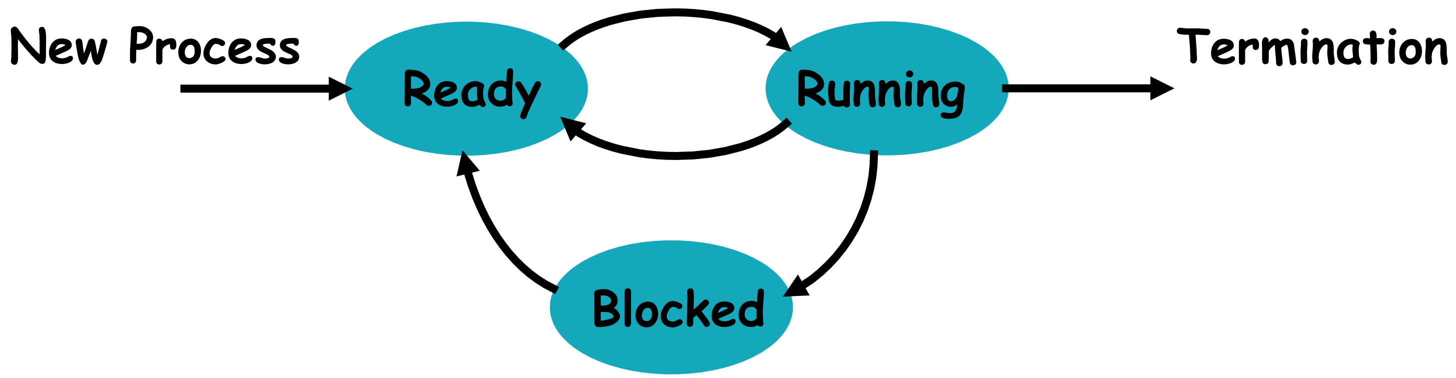
CPU Scheduling

CS 532 Winter 2020

Process State Model



Key Concept:
Separate Mechanism from Policy

Mechanisms: process, process switch

Policy: process scheduling. Which process should run next?

Separating policy from mechanism allows us to make the mechanism solid while allowing varying policy choices. (modularity)

CPU Scheduling Criteria

CPU Utilization
- *how busy is the CPU*?

Throughput
- *how many jobs finished/unit time*?

Turnaround Time
- *how long from job submission to job termination*?

Response Time
- *how long does it take to get a response*

Missed deadlines

Broad Classes of Scheduler

Priority-based?

- Processes have priorities
- Priorities can be assigned statically or dynamically

Preemptive?

- Processes can be switched at any time

Example Scheduling Policies

First-Come, First Served (FIFO)
Shortest Job First (non-preemptive)
Shortest Job First (with preemption)
Round-Robin Scheduling
Priority Scheduling
Real-Time Scheduling

How can we evaluate Scheduling Algorithms?

- Analytic Models
  - Parameters: information about the typical workload such as distribution of job length, interarrival interval
  - Determine: throughput, average queue length, average wait time, etc.
  - Benefits: Can explore different workloads

- Simulation
  - Given a file with a specific workload (captured from a real system, or created)
  - Determine throughput, average queue length, average wait time, etc.
  - Benefits: Fast (see homework 2)

First-Come, First-Served (FIFO)

Start jobs in the order they arrive (FIFO queue)

Run each job until completion

First-Come, First-Served (FIFO)

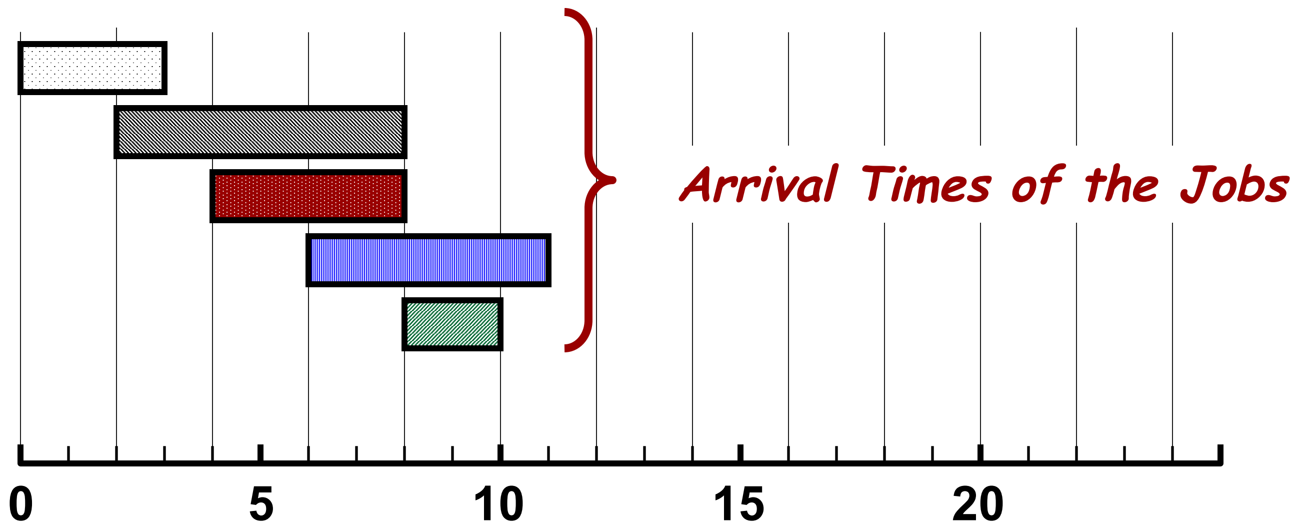
Process	Arrival Time	Processing Time
1	0	3
2	2	6
3	4	4
4	6	5
5	8	2

First-Come, First-Served (FIFO)

Process	Arrival Time	Processing Time	Delay	Turnaround Time
1	0	3		
2	2	6		
3	4	4		
4	6	5		
5	8	2		

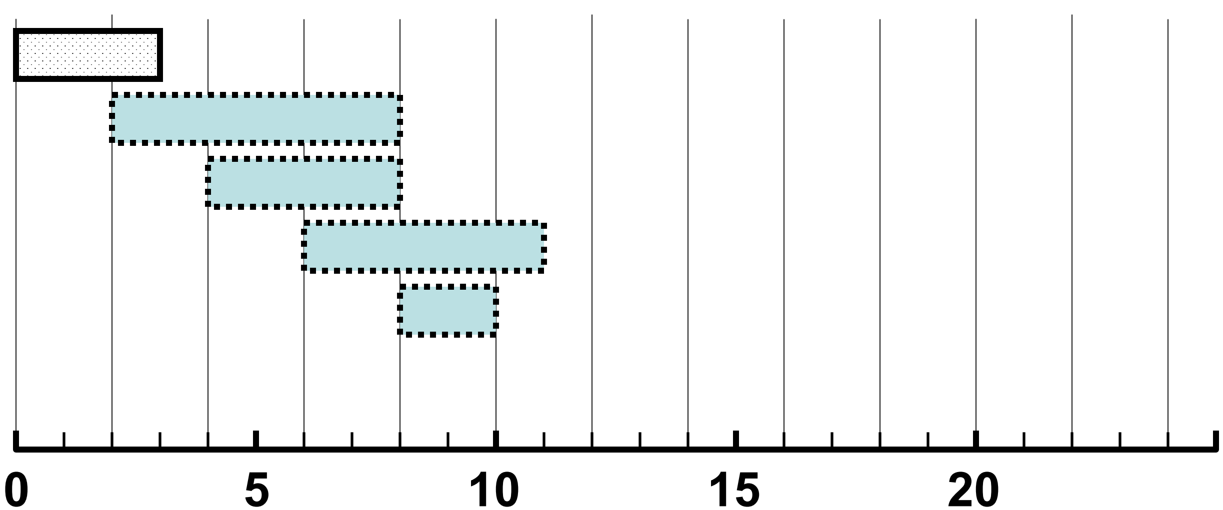
First-Come, First-Served (FIFO)

Process	Arrival Time	Processing Time	Delay	Turnaround Time
1	0	3		
2	2	6		
3	4	4		
4	6	5		
5	8	2		



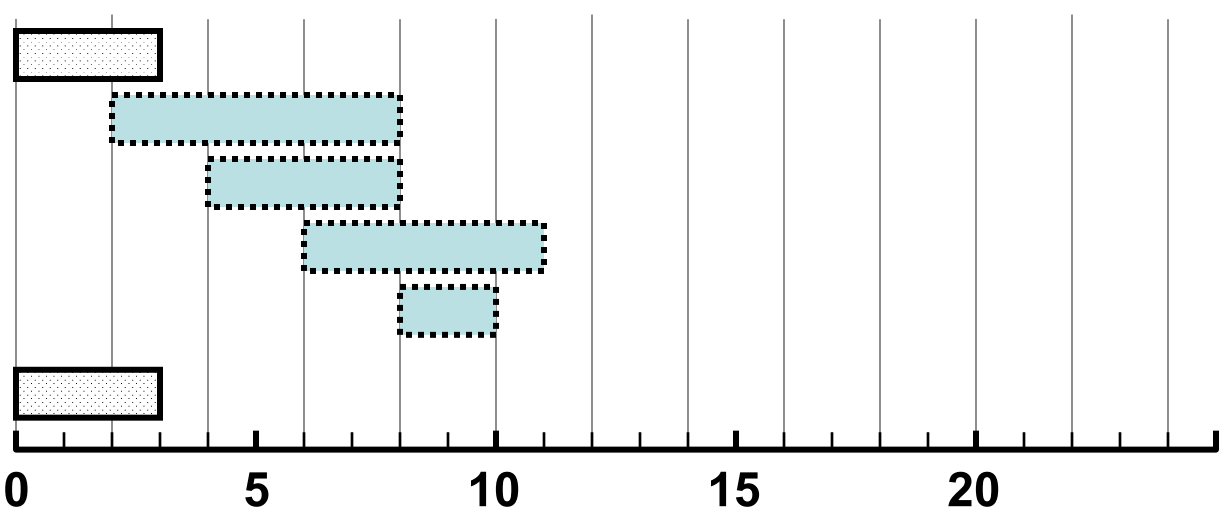
First-Come, First-Served (FIFO)

Process	Arrival Time	Processing Time	Delay	Turnaround Time
1	0	3		
2	2	6		
3	4	4		
4	6	5		
5	8	2		



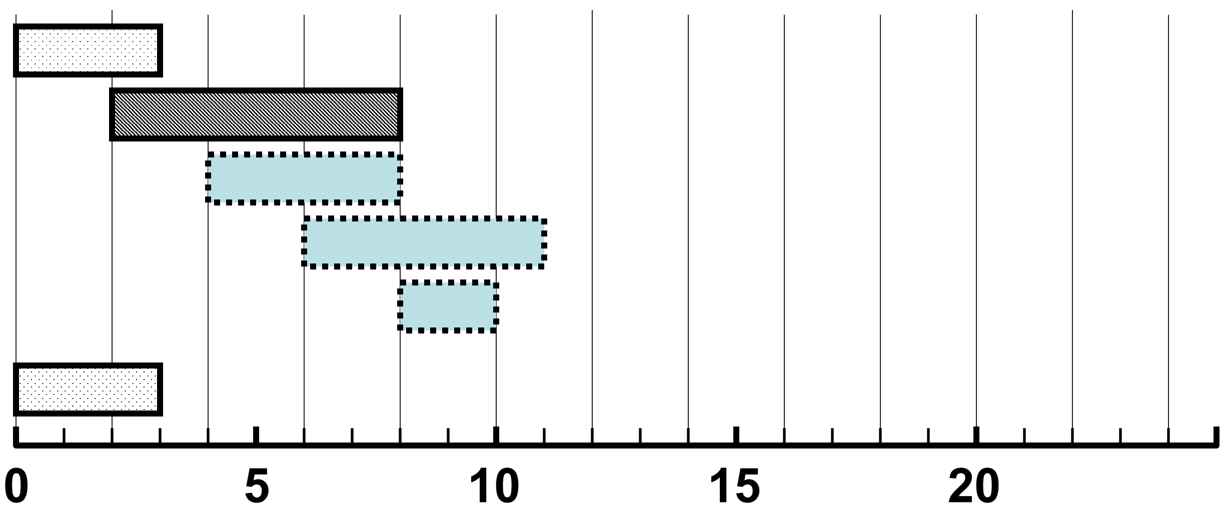
First-Come, First-Served (FIFO)

Process	Arrival Time	Processing Time	Delay	Turnaround Time
1	0	3		
2	2	6		
3	4	4		
4	6	5		
5	8	2		



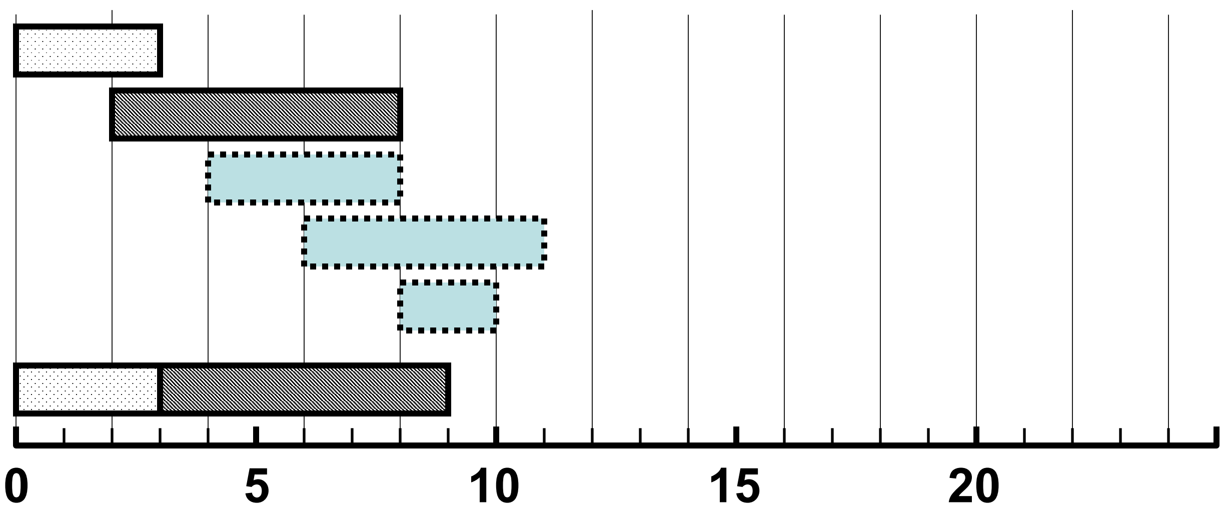
First-Come, First-Served (FIFO)

Process	Arrival Time	Processing Time	Delay	Turnaround Time
1	0	3		
2	2	6		
3	4	4		
4	6	5		
5	8	2		



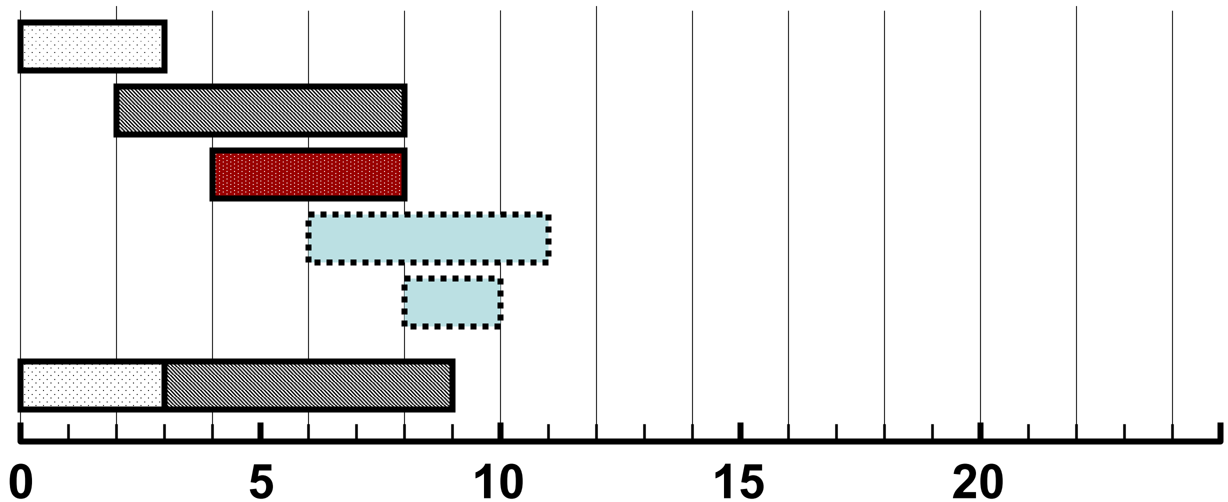
First-Come, First-Served (FIFO)

Process	Arrival Time	Processing Time	Delay	Turnaround Time
1	0	3		
2	2	6		
3	4	4		
4	6	5		
5	8	2		



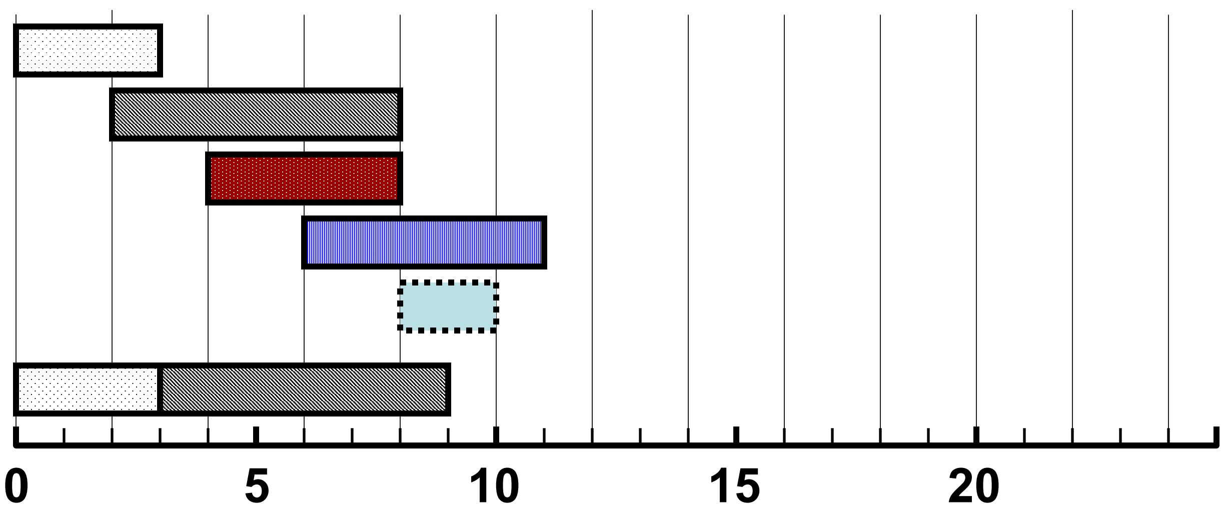
First-Come, First-Served (FIFO)

Process		Arrival Time	Processing Time	Delay	Turnaround Time
	1	0	3		
	2	2	6		
	3	4	4		
	4	6	5		
	5	8	2		



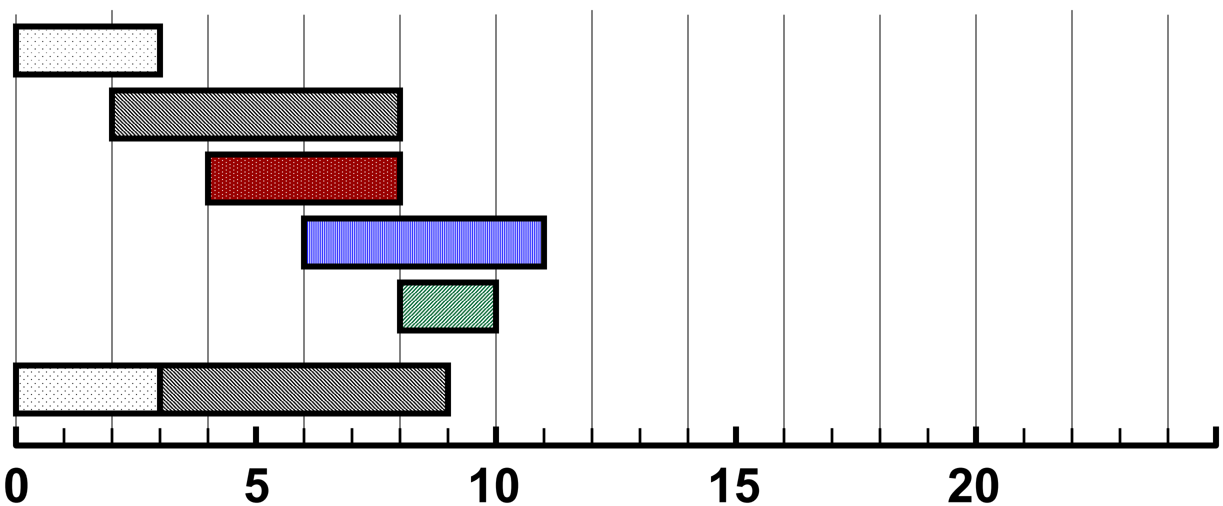
First-Come, First-Served (FIFO)

Process	Arrival Time	Processing Time	Delay	Turnaround Time
1	0	3		
2	2	6		
3	4	4		
4	6	5		
5	8	2		



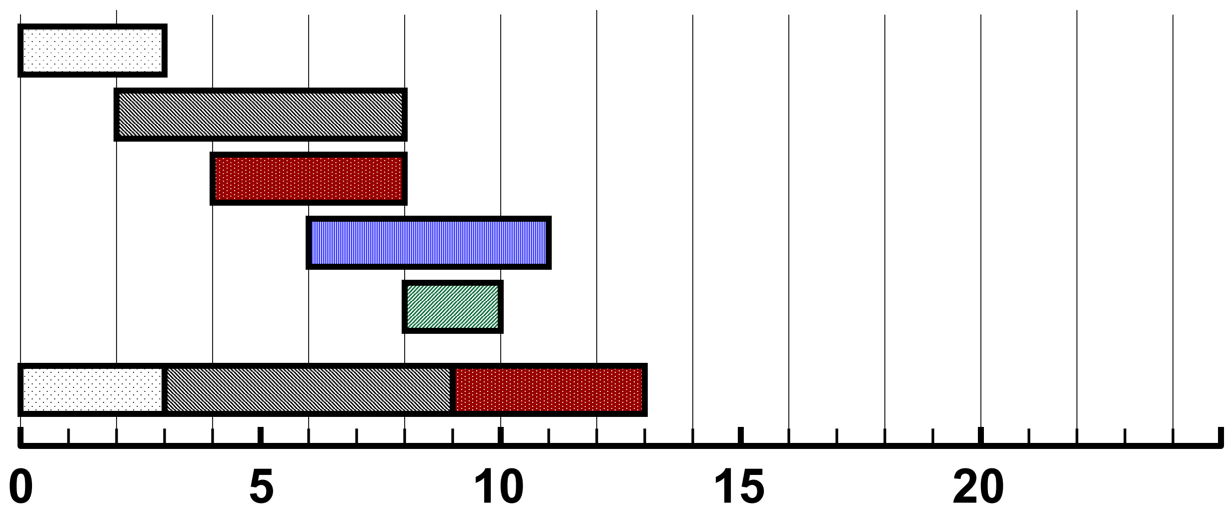
First-Come, First-Served (FIFO)

Process	Arrival Time	Processing Time	Delay	Turnaround Time
1	0	3		
2	2	6		
3	4	4		
4	6	5		
5	8	2		



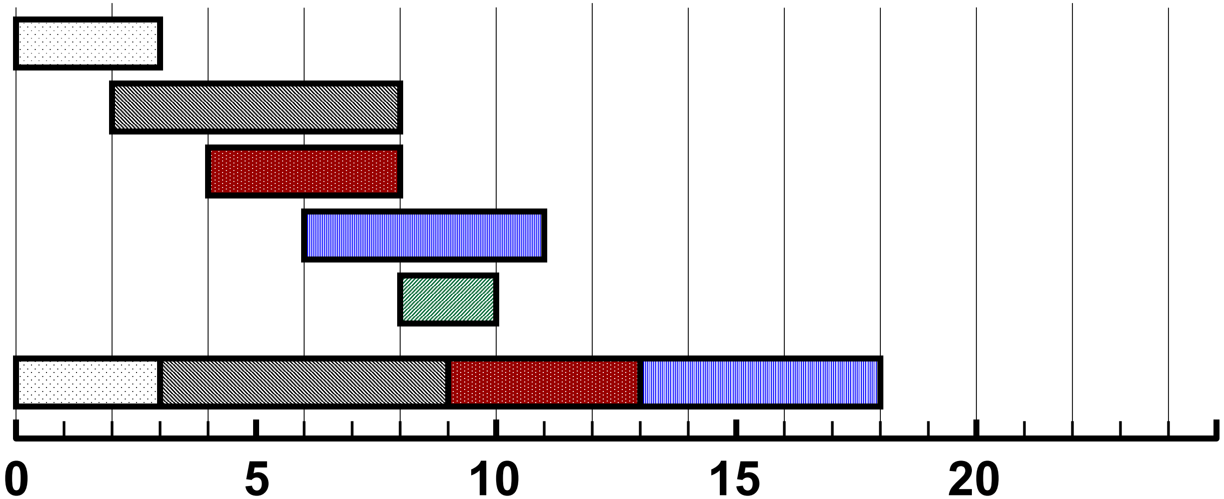
First-Come, First-Served (FIFO)

Process	Arrival Time	Processing Time	Delay	Turnaround Time
1	0	3		
2	2	6		
3	4	4		
4	6	5		
5	8	2		



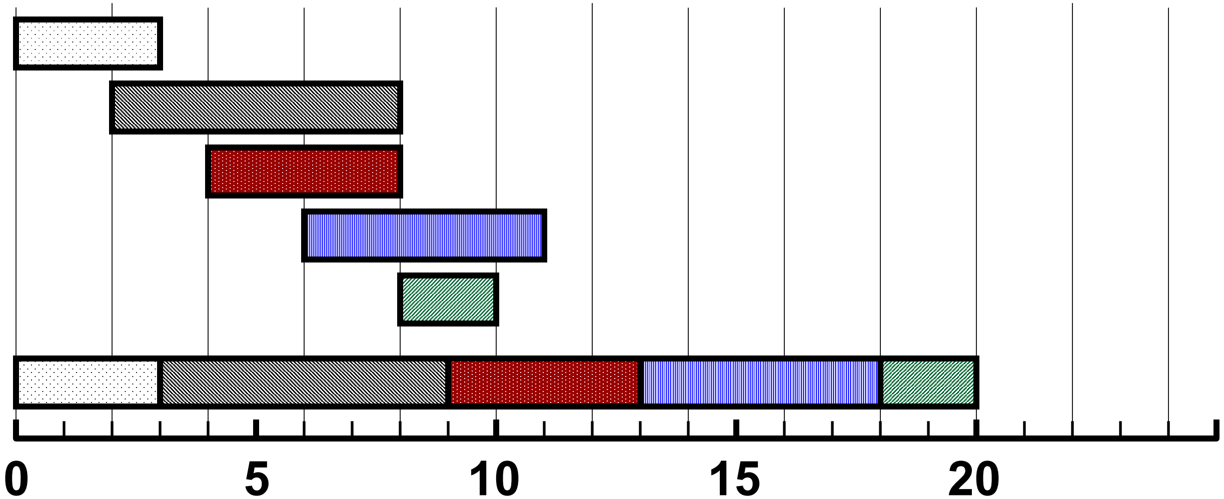
First-Come, First-Served (FIFO)

Process	Arrival Time	Processing Time	Delay	Turnaround Time
1	0	3		
2	2	6		
3	4	4		
4	6	5		
5	8	2		



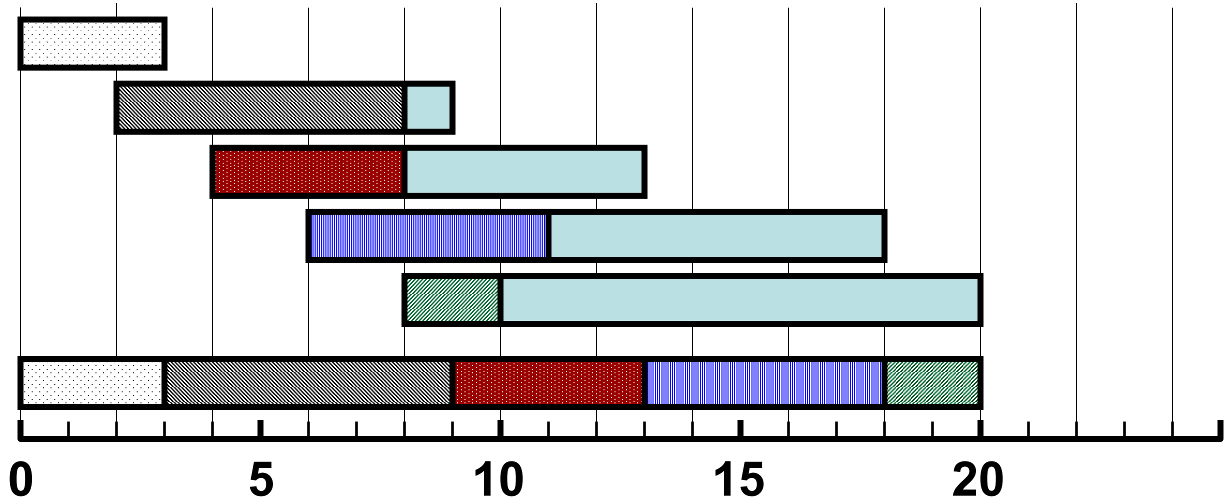
First-Come, First-Served (FIFO)

Process	Arrival Time	Processing Time	Delay	Turnaround Time
1	0	3		
2	2	6		
3	4	4		
4	6	5		
5	8	2		



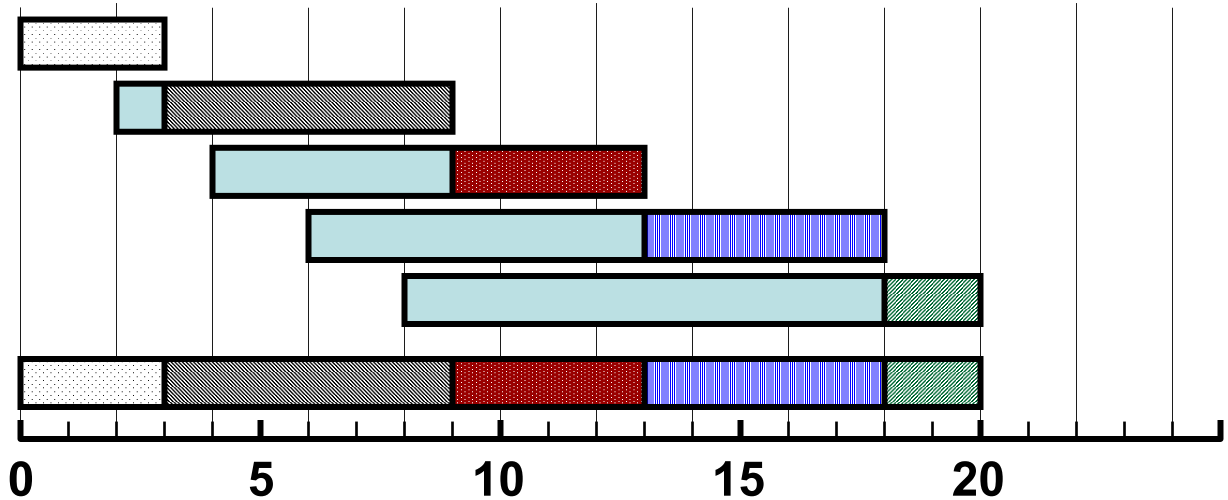
First-Come, First-Served (FIFO)

Process	Arrival Time	Processing Time	Delay	Turnaround Time
1	0	3		
2	2	6		
3	4	4		
4	6	5		
5	8	2		



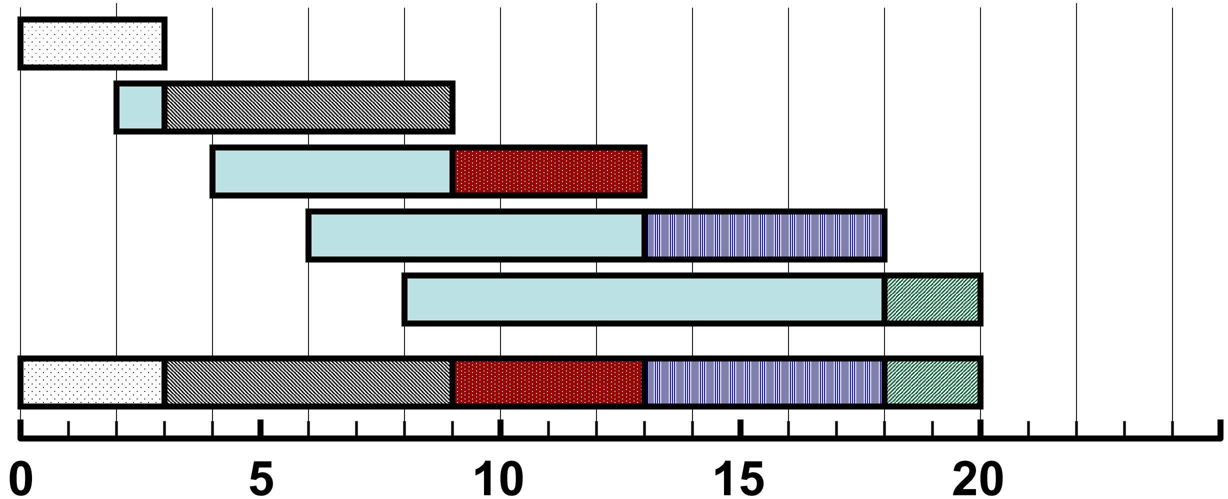
First-Come, First-Served (FIFO)

Process	Arrival Time	Processing Time	Delay	Turnaround Time
1	0	3		
2	2	6		
3	4	4		
4	6	5		
5	8	2		



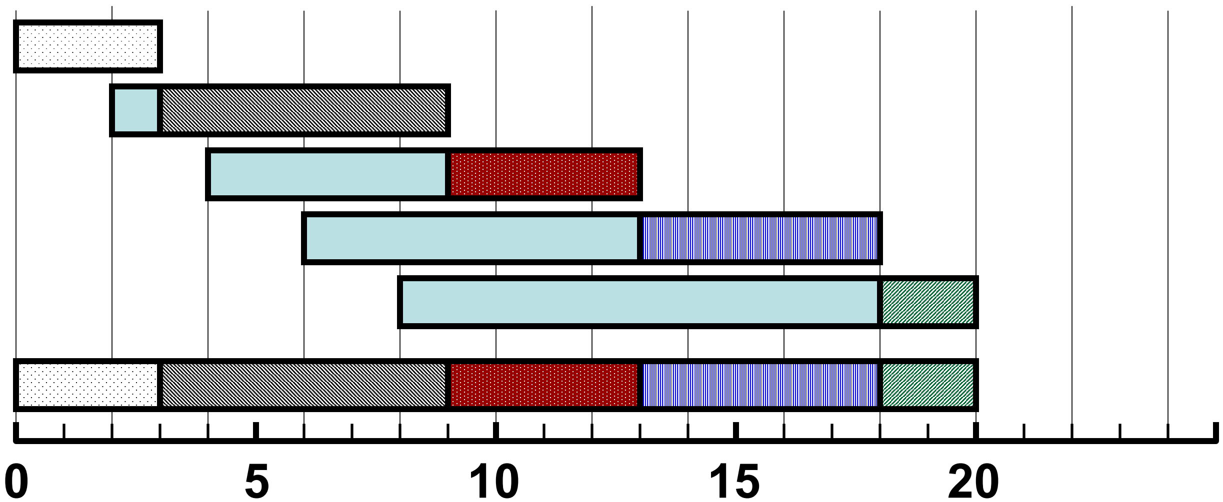
First-Come, First-Served (FIFO)

Process	Arrival Time	Processing Time	Delay	Turnaround Time
1	0	3	0	
2	2	6	1	
3	4	4	5	
4	6	5	7	
5	8	2	10	



First-Come, First-Served (FIFO)

Process		Arrival Time	Processing Time	Delay	Turnaround Time
	1	0	3	0	3
	2	2	6	1	7
	3	4	4	5	9
	4	6	5	7	12
	5	8	2	10	12

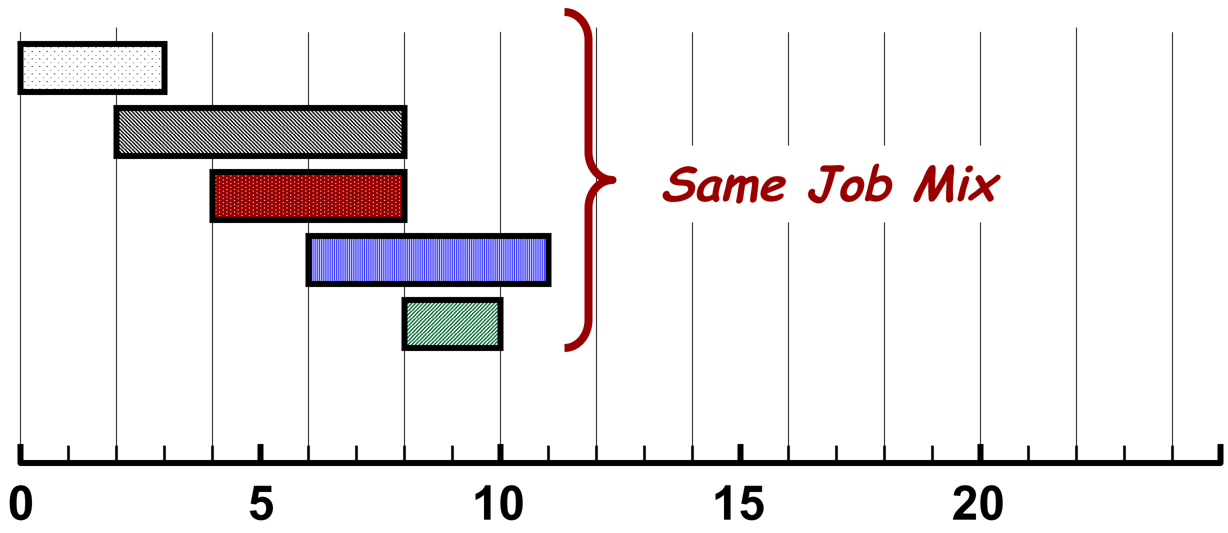


Shortest Job First

- Select the job with the shortest (expected) running time
- Non-Preemptive

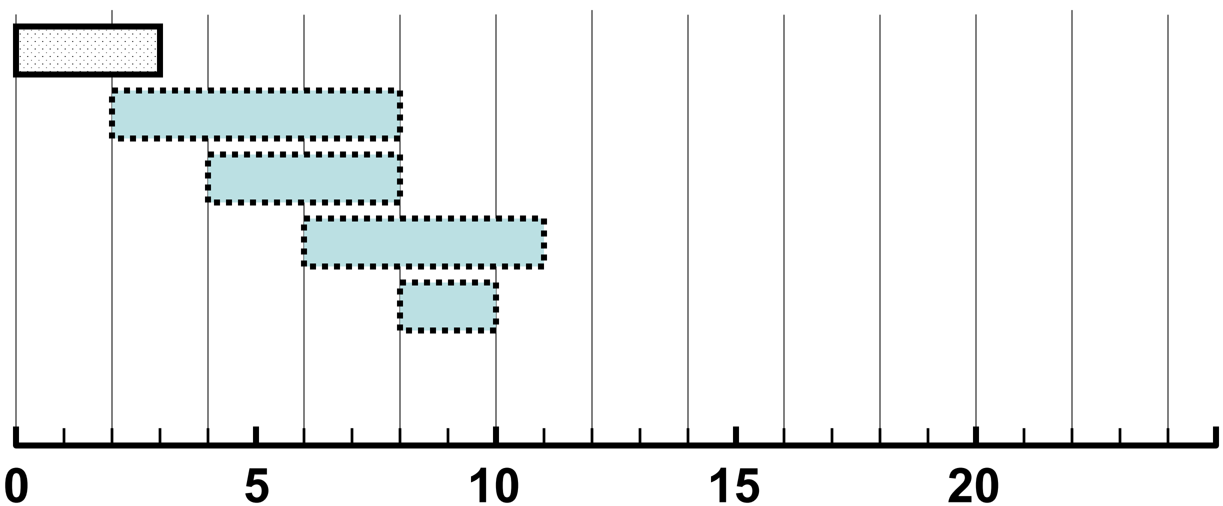
Shortest Job First

Process	Arrival Time	Processing Time	Delay	Turnaround Time
1	0	3		
2	2	6		
3	4	4		
4	6	5		
5	8	2		



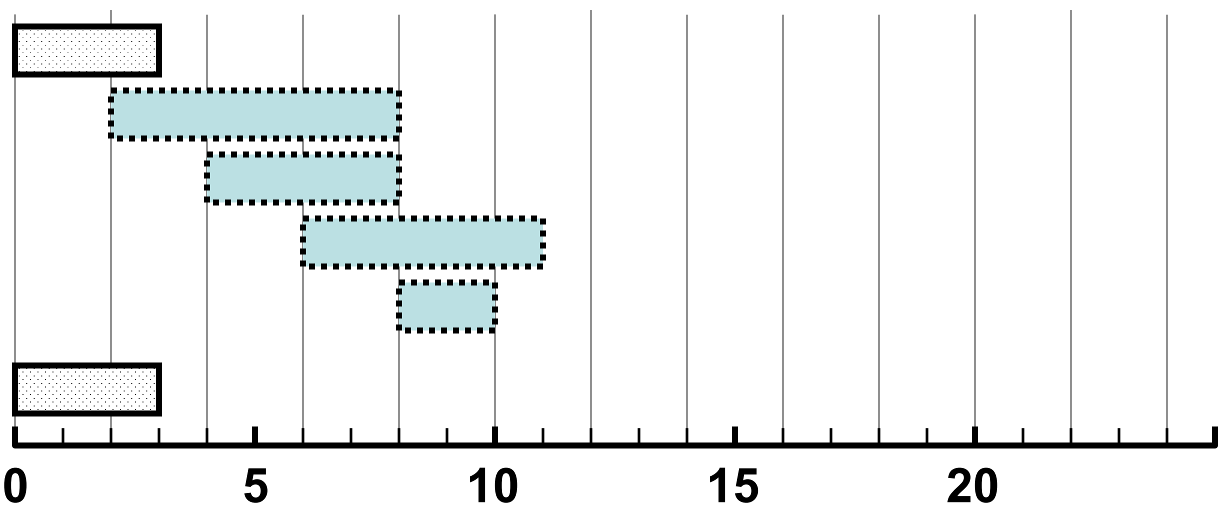
Shortest Job First

Process	Arrival Time	Processing Time	Delay	Turnaround Time
1	0	3		
2	2	6		
3	4	4		
4	6	5		
5	8	2		



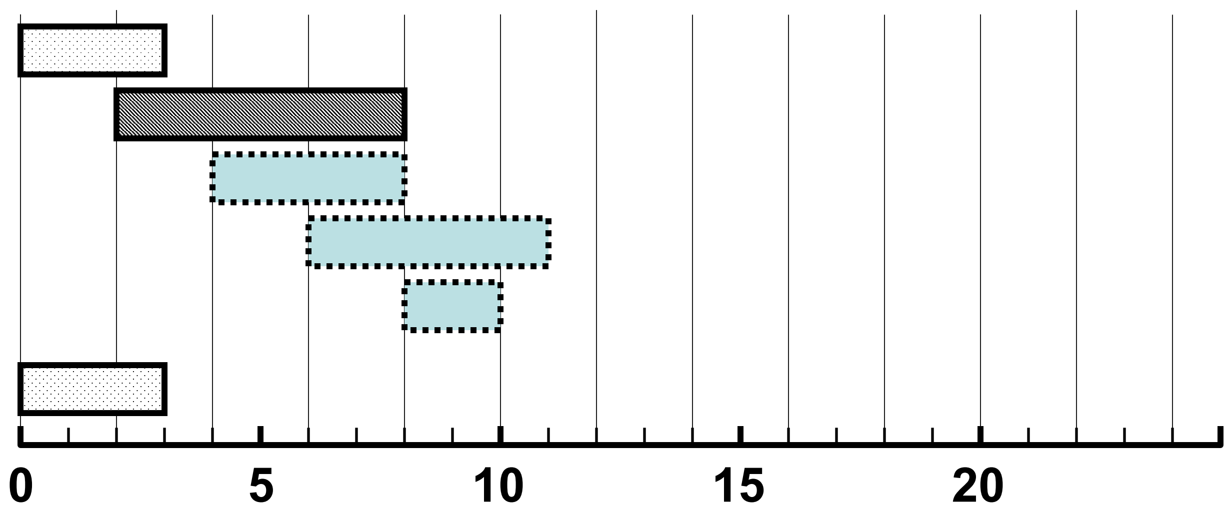
Shortest Job First

Process	Arrival Time	Processing Time	Delay	Turnaround Time
1	0	3		
2	2	6		
3	4	4		
4	6	5		
5	8	2		



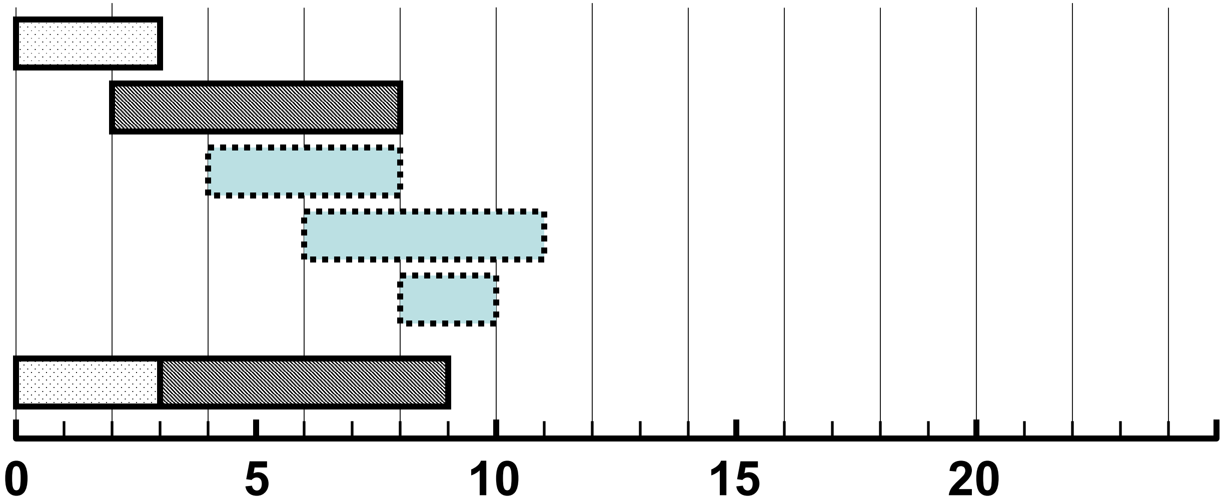
Shortest Job First

Process		Arrival Time	Processing Time	Delay	Turnaround Time
	1	0	3		
	2	2	6		
	3	4	4		
	4	6	5		
	5	8	2		



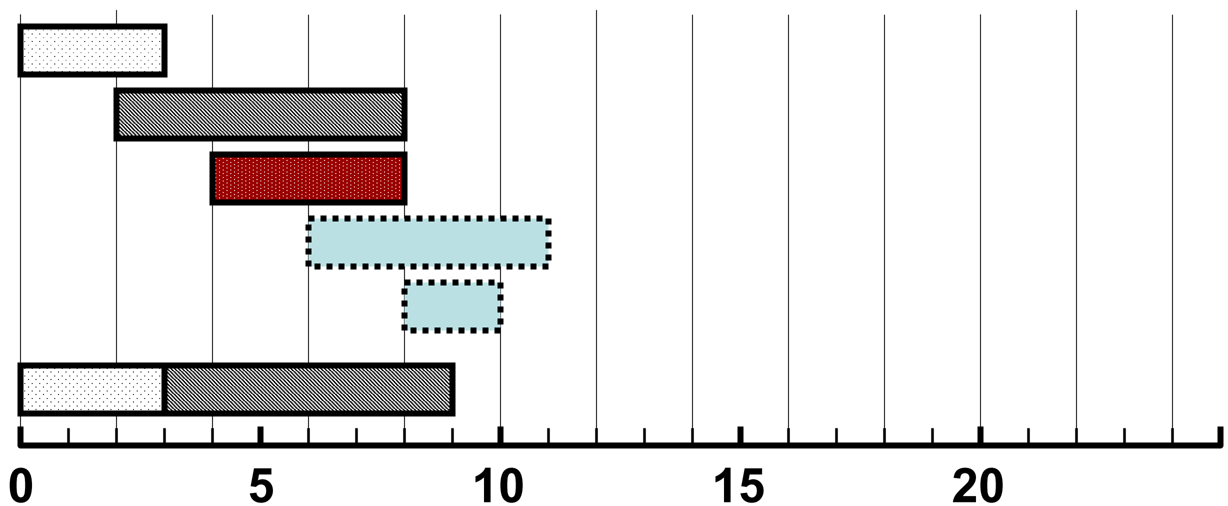
Shortest Job First

Process	Arrival Time	Processing Time	Delay	Turnaround Time
1	0	3		
2	2	6		
3	4	4		
4	6	5		
5	8	2		



Shortest Job First

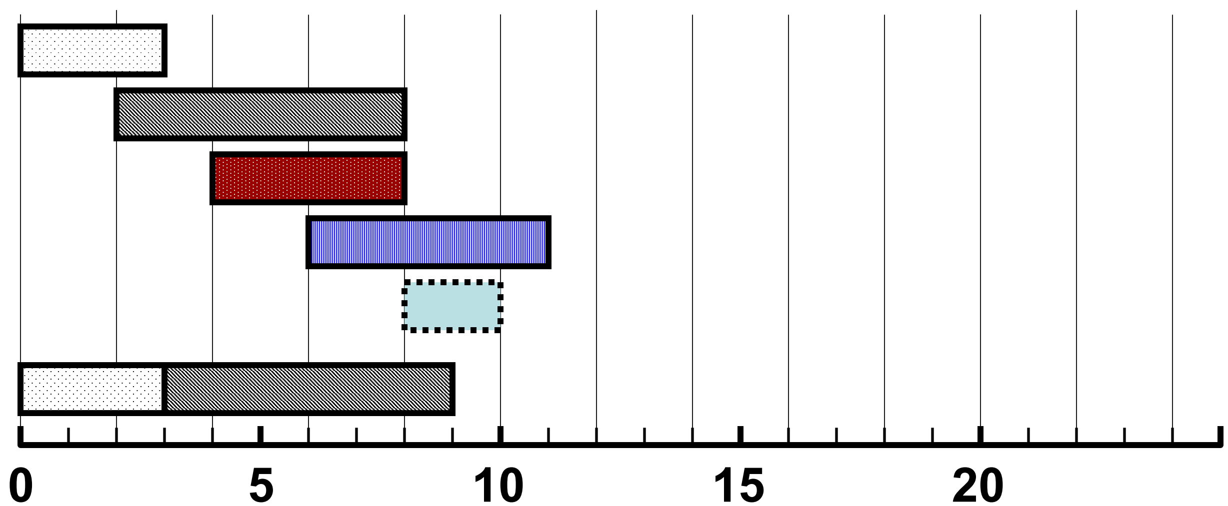
Process	Arrival Time	Processing Time	Delay	Turnaround Time
1	0	3		
2	2	6		
3	4	4		
4	6	5		
5	8	2		



Portland State UNIVERSITY

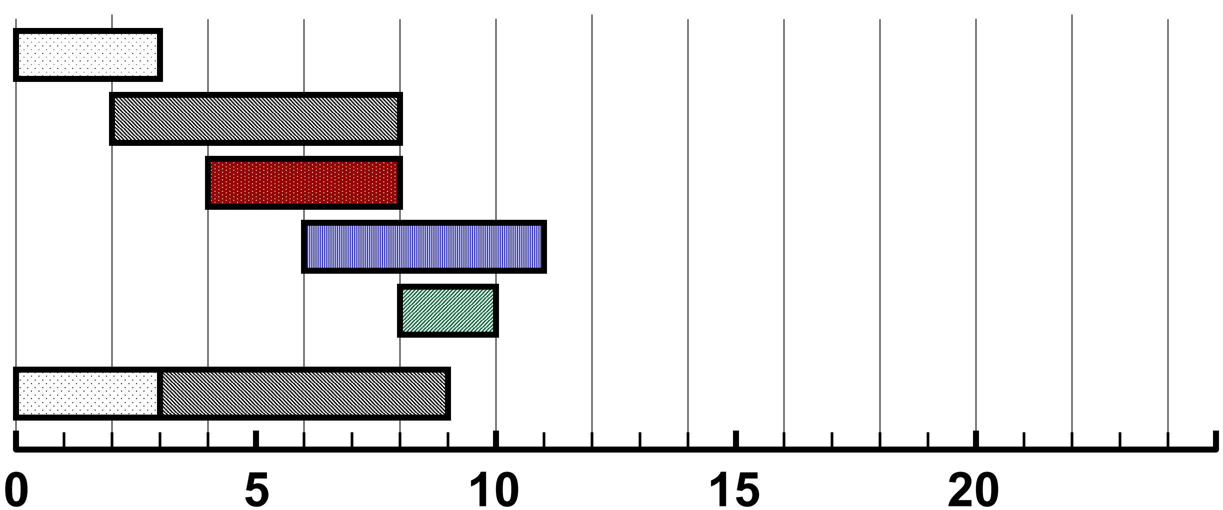
Shortest Job First

Process		Arrival Time	Processing Time	Delay	Turnaround Time
	1	0	3		
	2	2	6		
	3	4	4		
	4	6	5		
	5	8	2		



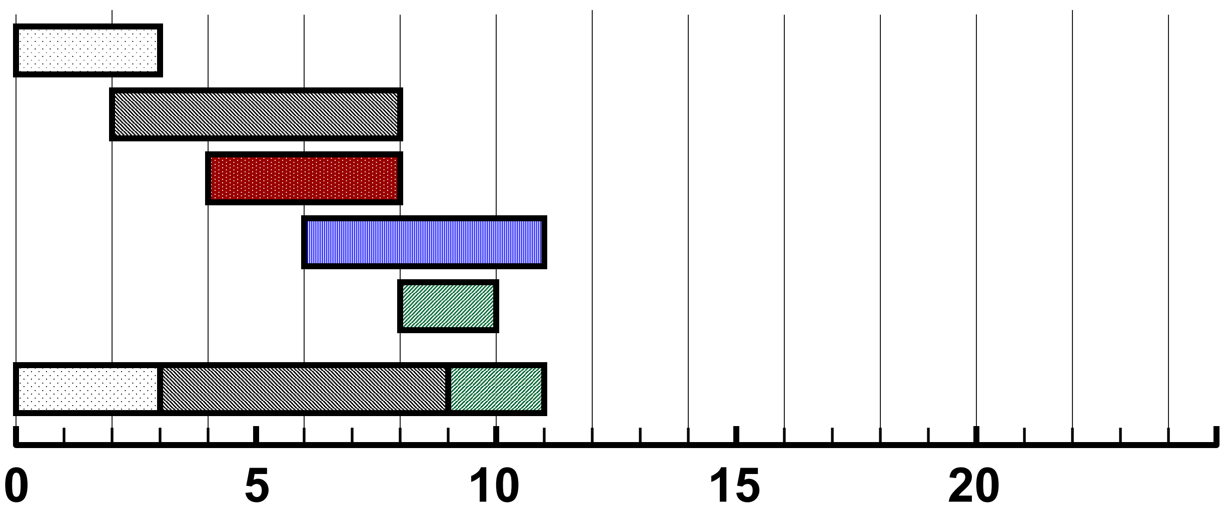
Shortest Job First

Process	Arrival Time	Processing Time	Delay	Turnaround Time
1	0	3		
2	2	6		
3	4	4		
4	6	5		
5	8	2		



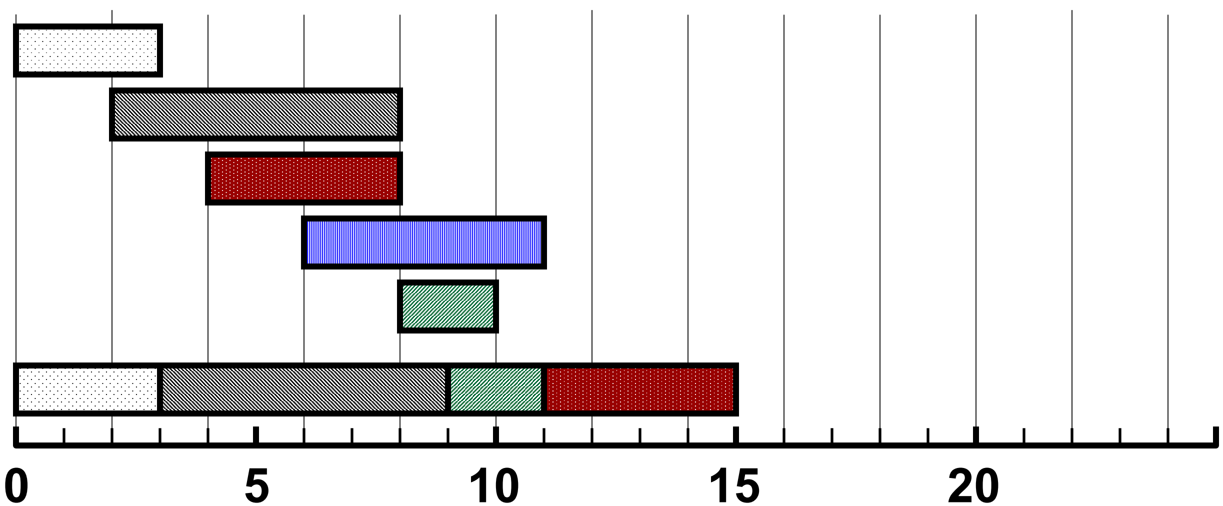
Shortest Job First

Process	Arrival Time	Processing Time	Delay	Turnaround Time
1	0	3		
2	2	6		
3	4	4		
4	6	5		
5	8	2		



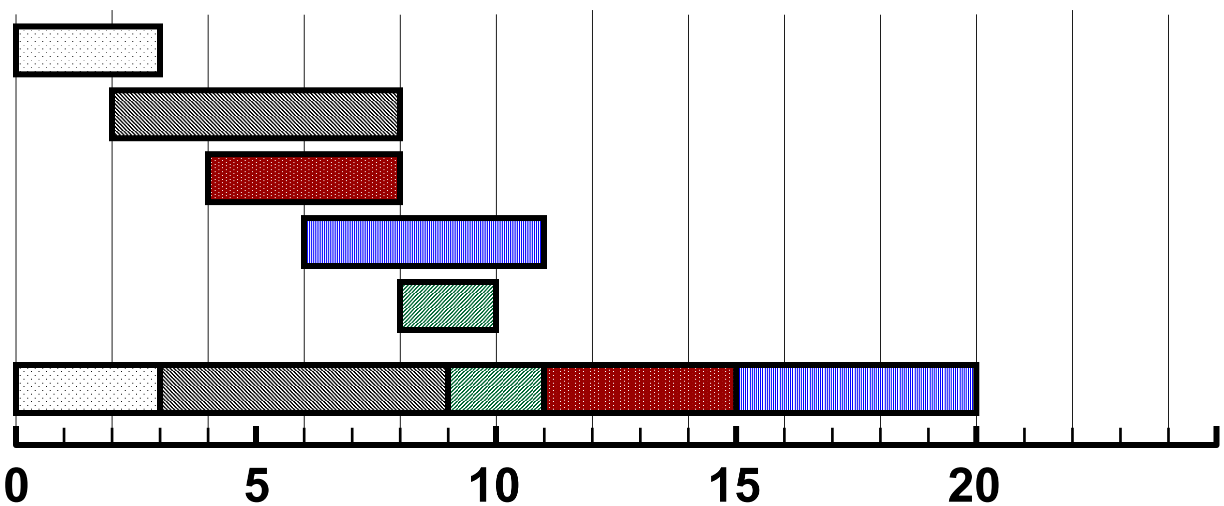
Shortest Job First

Process	Arrival Time	Processing Time	Delay	Turnaround Time
1	0	3		
2	2	6		
3	4	4		
4	6	5		
5	8	2		



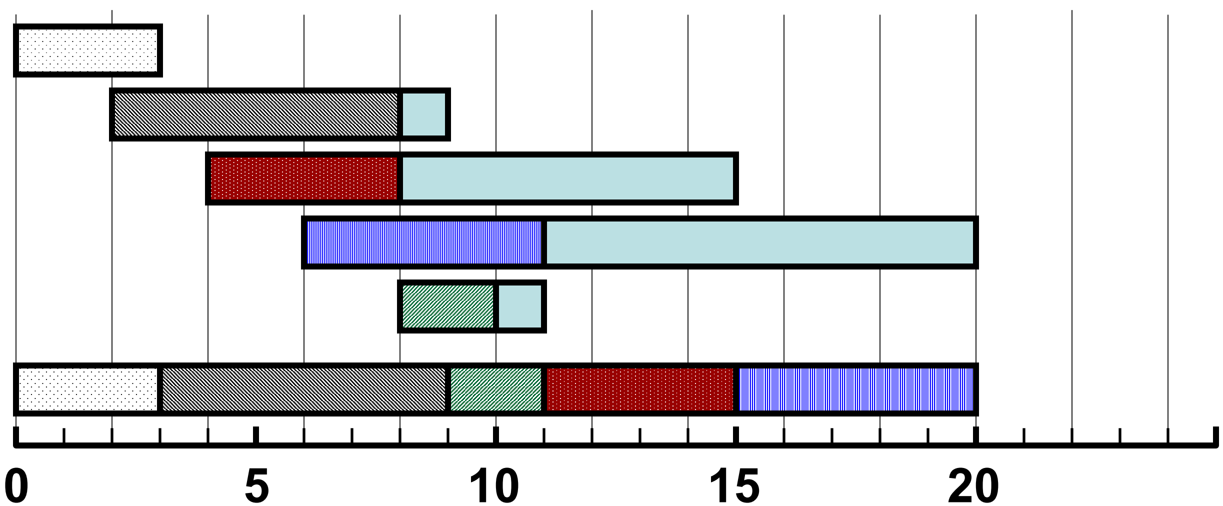
Shortest Job First

Process	Arrival Time	Processing Time	Delay	Turnaround Time
1	0	3		
2	2	6		
3	4	4		
4	6	5		
5	8	2		



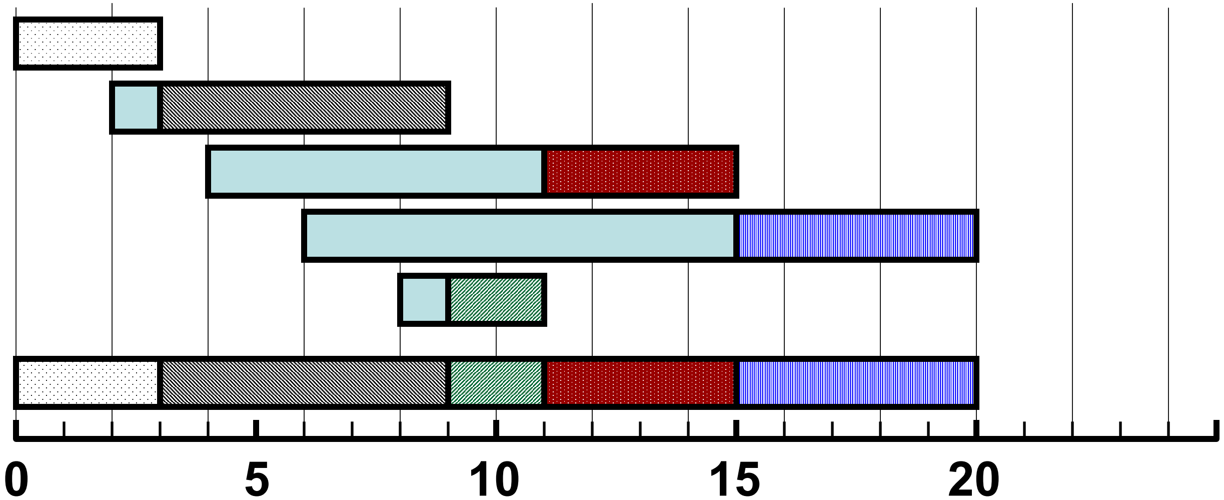
Shortest Job First

Process		Arrival Time	Processing Time	Delay	Turnaround Time
	1	0	3		
	2	2	6		
	3	4	4		
	4	6	5		
	5	8	2		



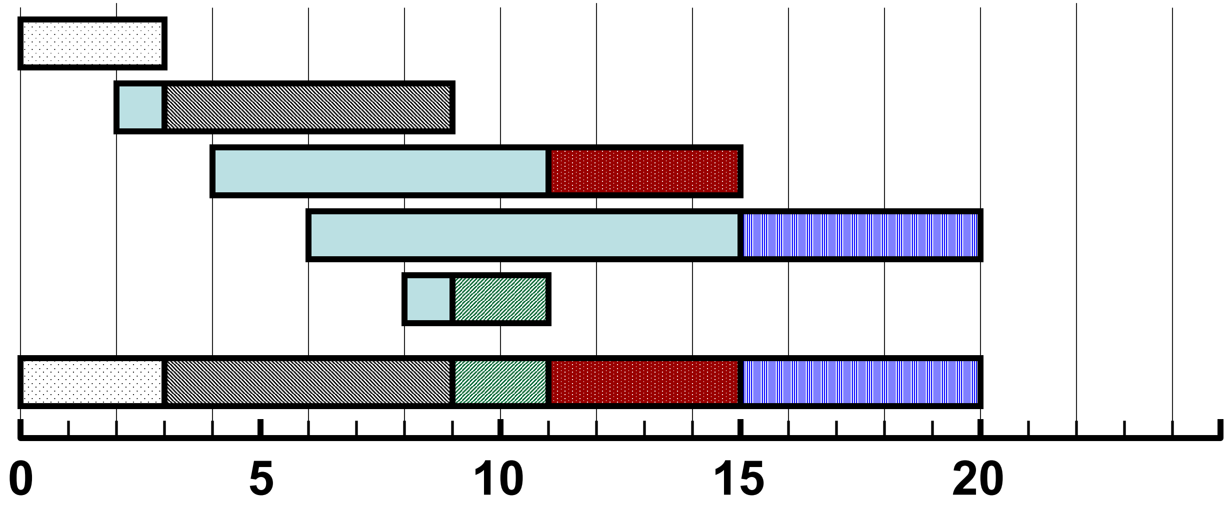
Shortest Job First

Process		Arrival Time	Processing Time	Delay	Turnaround Time
	1	0	3		
	2	2	6		
	3	4	4		
	4	6	5		
	5	8	2		



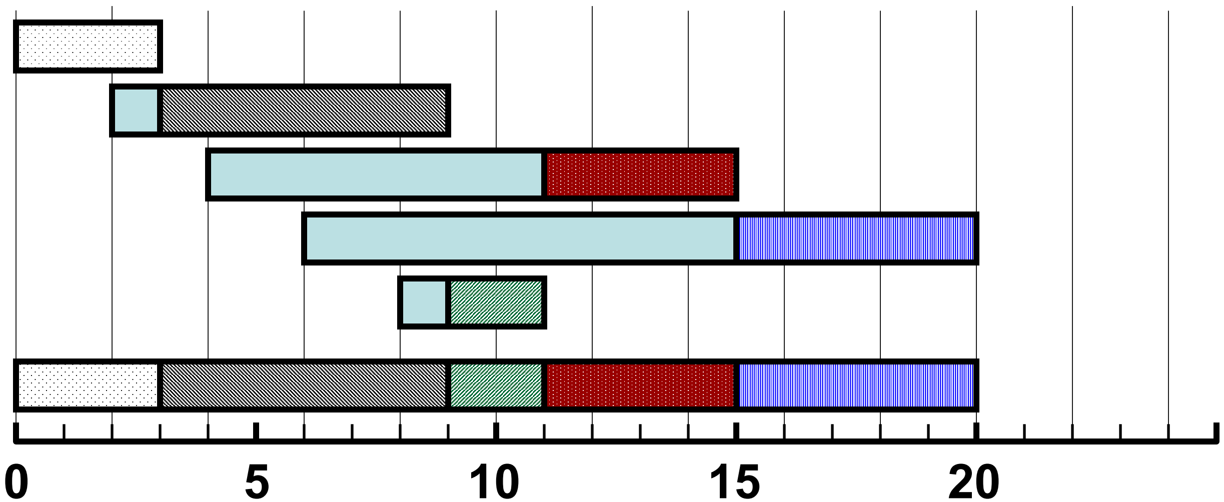
Shortest Job First

Process		Arrival Time	Processing Time	Delay	Turnaround Time
	1	0	3	0	
	2	2	6	1	
	3	4	4	7	
	4	6	5	9	
	5	8	2	1	



Shortest Job First

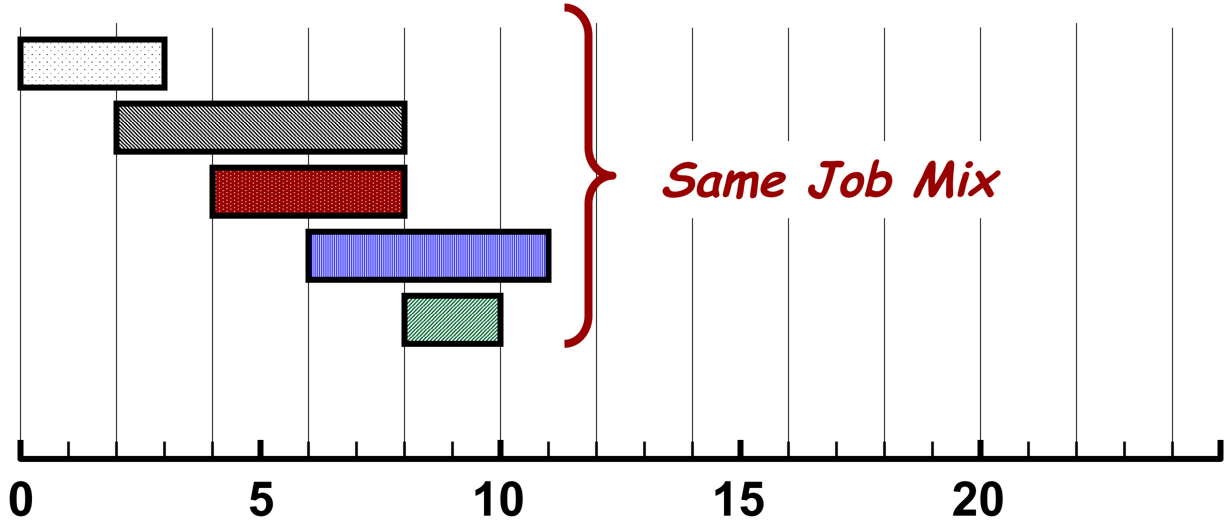
Process	Arrival Time	Processing Time	Delay	Turnaround Time
1	0	3	0	3
2	2	6	1	7
3	4	4	7	11
4	6	5	9	14
5	8	2	1	3



Shortest Remaining Time

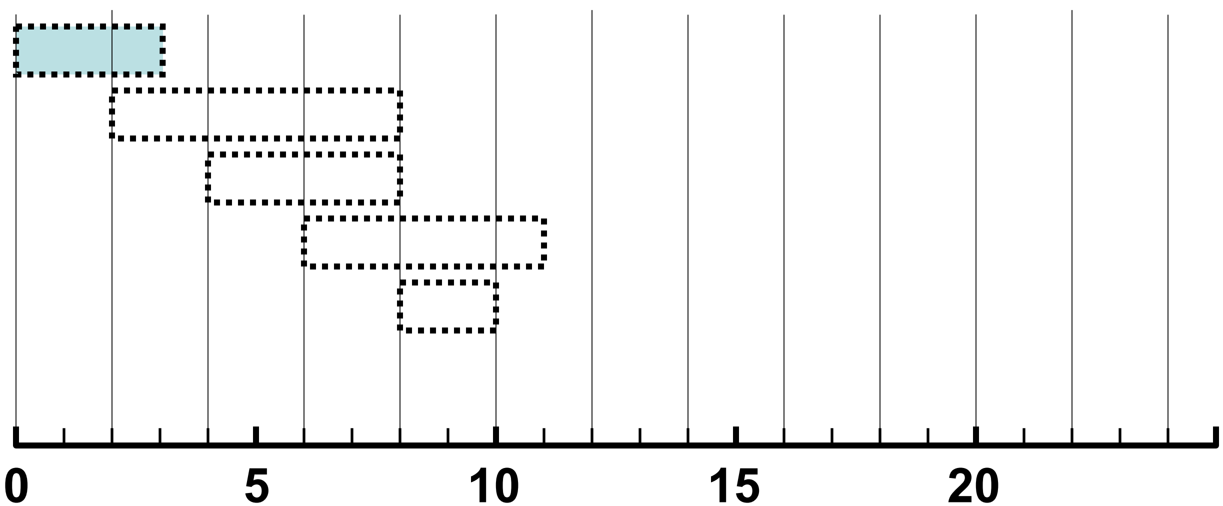
Preemptive version of SJF

Process	Arrival Time	Processing Time	Delay	Turnaround Time
1	0	3		
2	2	6		
3	4	4		
4	6	5		
5	8	2		



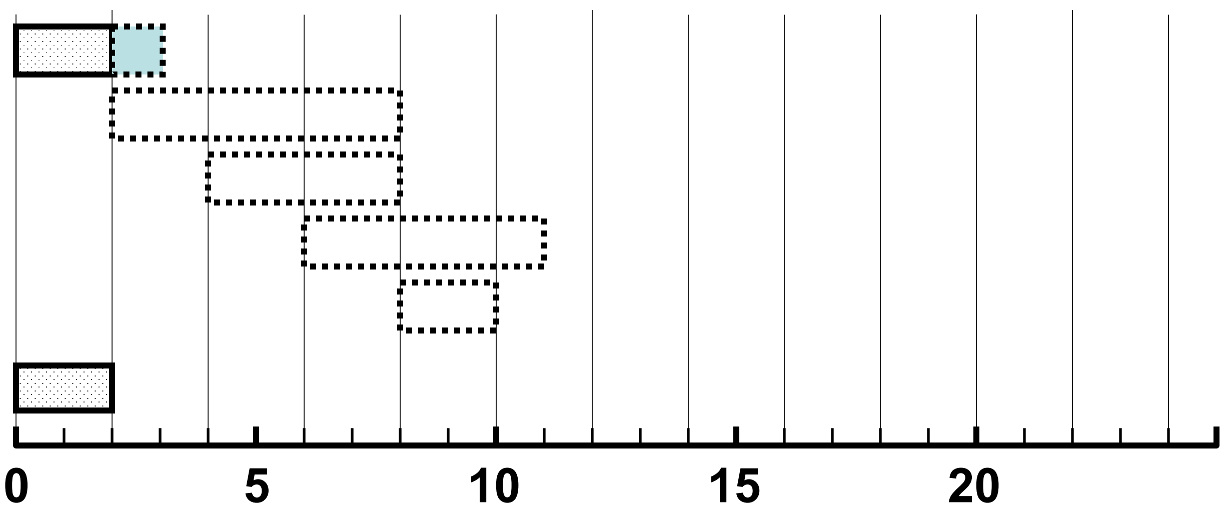
Shortest Remaining Time

Process	Arrival Time	Processing Time	Delay	Turnaround Time
1	0	3		
2	2	6		
3	4	4		
4	6	5		
5	8	2		



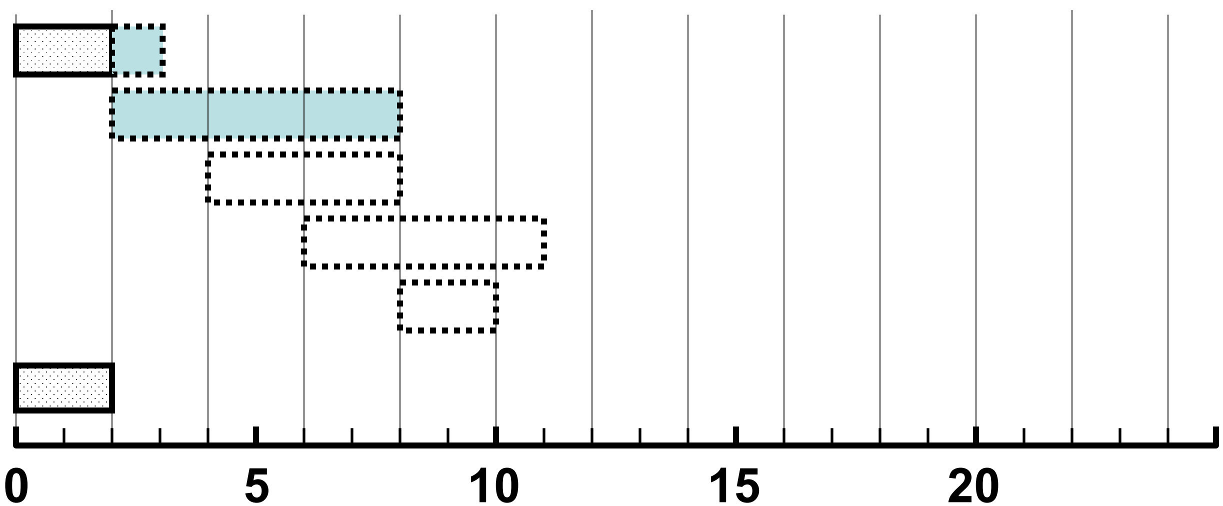
Shortest Remaining Time

Process		Arrival Time	Processing Time	Delay	Turnaround Time
▢	1	0	3		
▢	2	2	6		
▢	3	4	4		
▢	4	6	5		
▢	5	8	2		



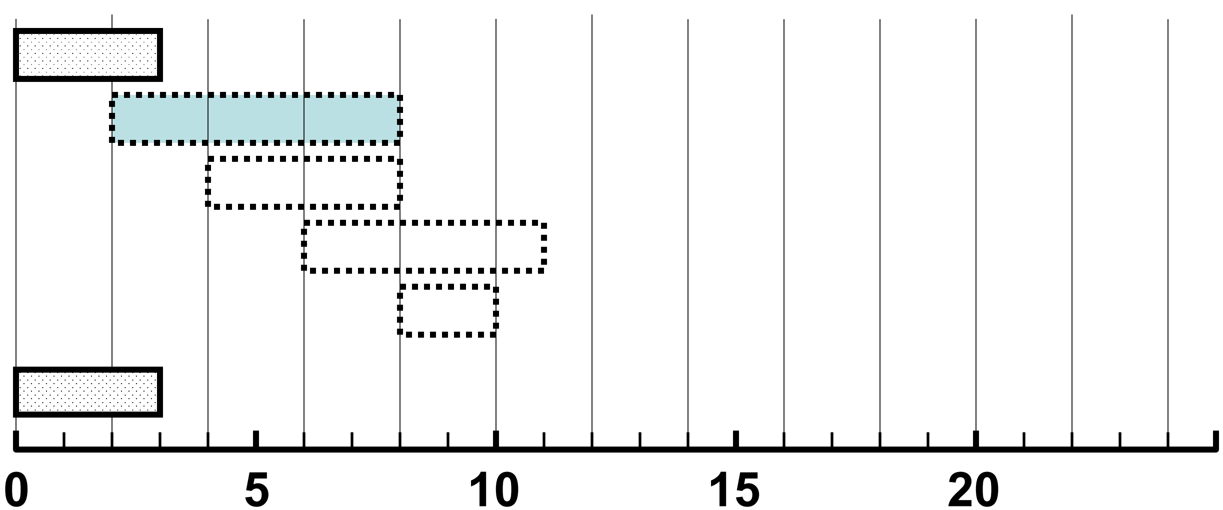
Shortest Remaining Time

Process	Arrival Time	Processing Time	Delay	Turnaround Time
1	0	3		
2	2	6		
3	4	4		
4	6	5		
5	8	2		



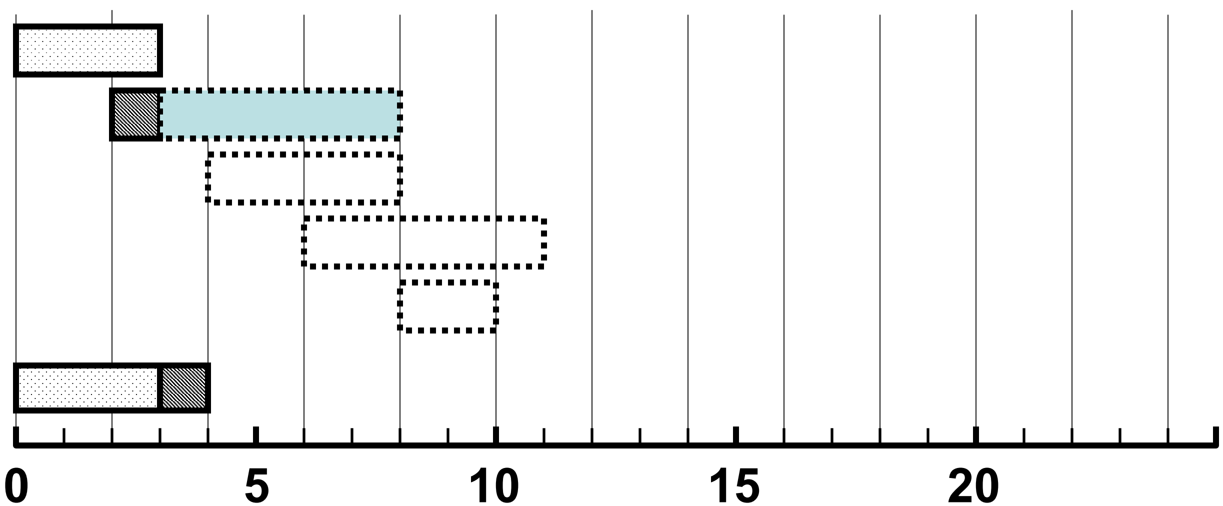
Shortest Remaining Time

Process	Arrival Time	Processing Time	Delay	Turnaround Time
1	0	3		
2	2	6		
3	4	4		
4	6	5		
5	8	2		



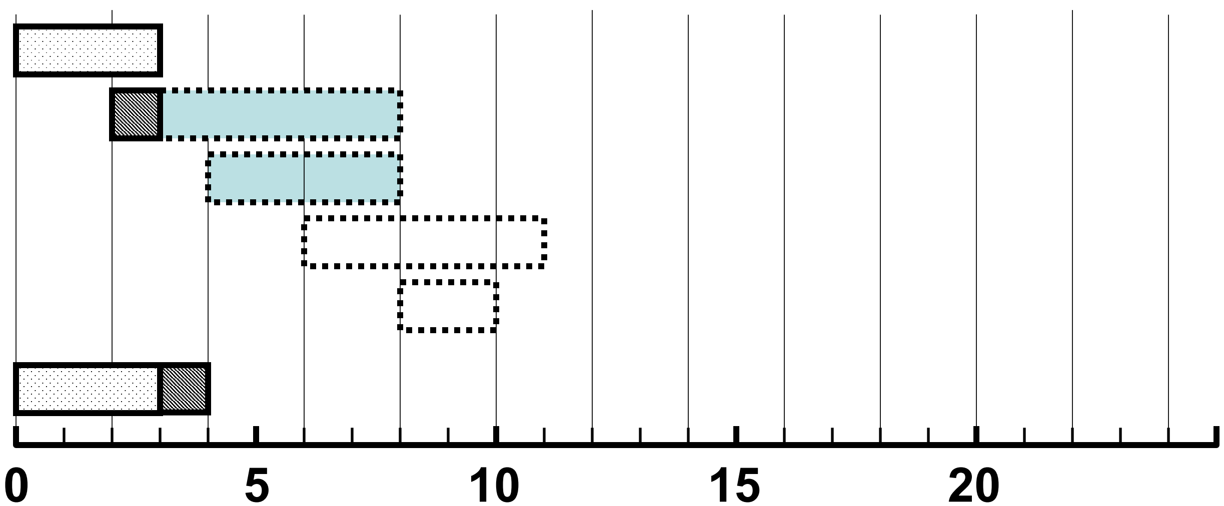
Shortest Remaining Time

	Process	Arrival Time	Processing Time	Delay	Turnaround Time
	1	0	3		
	2	2	6		
	3	4	4		
	4	6	5		
	5	8	2		



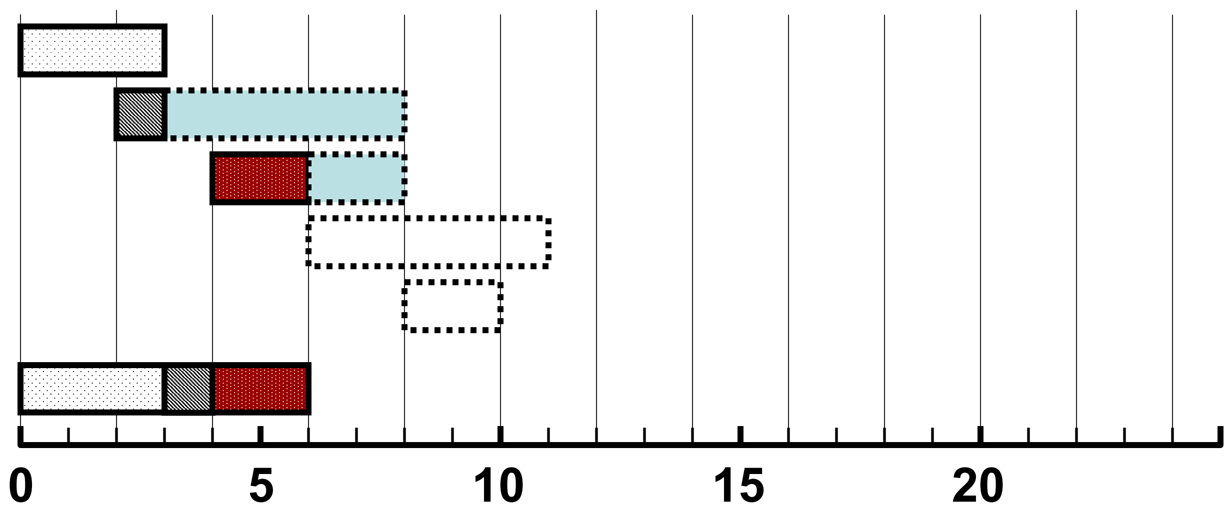
Shortest Remaining Time

Process		Arrival Time	Processing Time	Delay	Turnaround Time
	1	0	3		
	2	2	6		
	3	4	4		
	4	6	5		
	5	8	2		



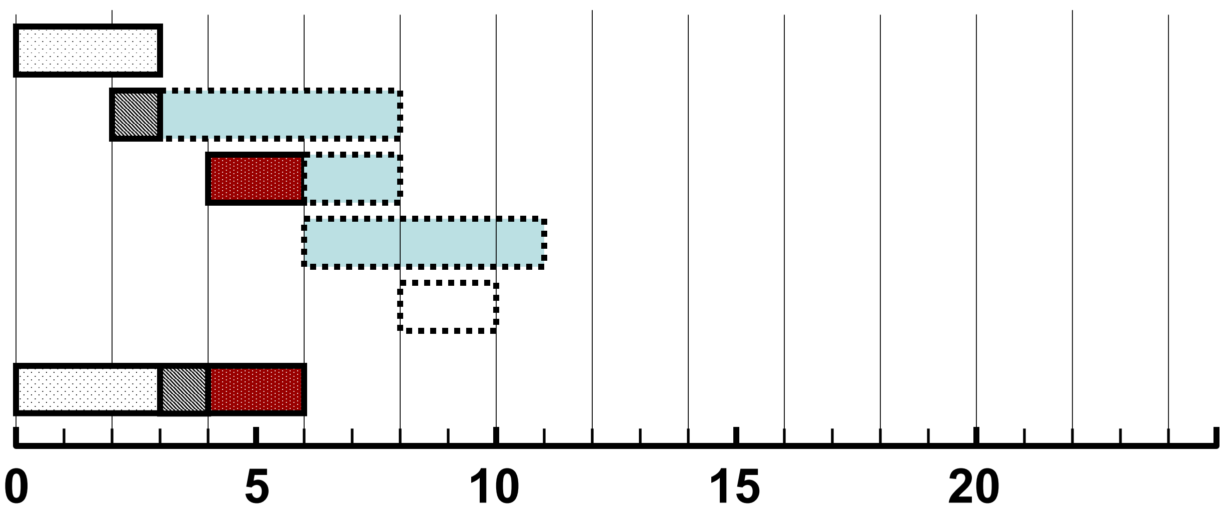
Shortest Remaining Time

Process		Arrival Time	Processing Time	Delay	Turnaround Time
	1	0	3		
	2	2	6		
	3	4	4		
	4	6	5		
	5	8	2		



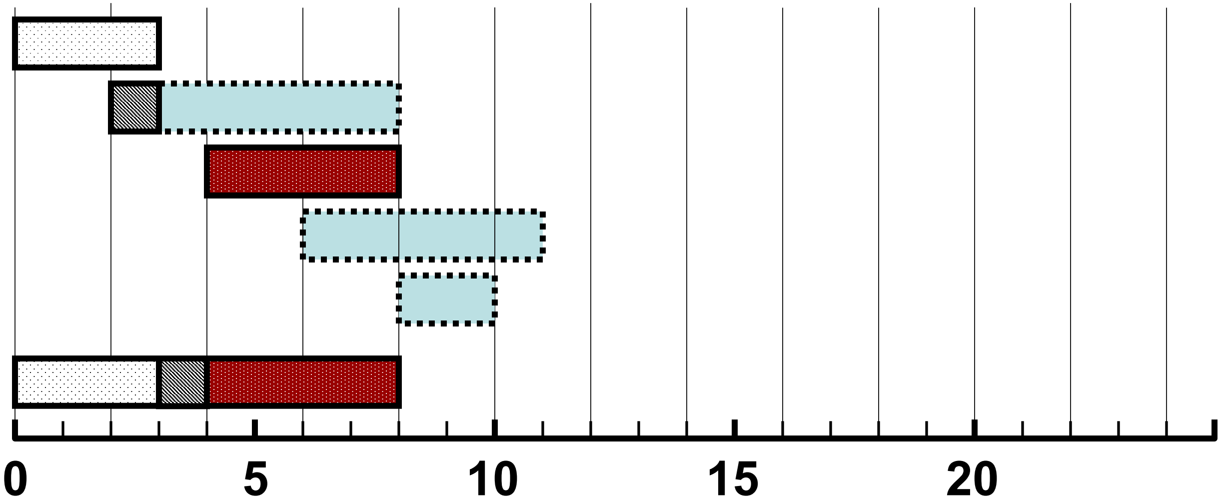
Shortest Remaining Time

Process	Arrival Time	Processing Time	Delay	Turnaround Time
1	0	3		
2	2	6		
3	4	4		
4	6	5		
5	8	2		



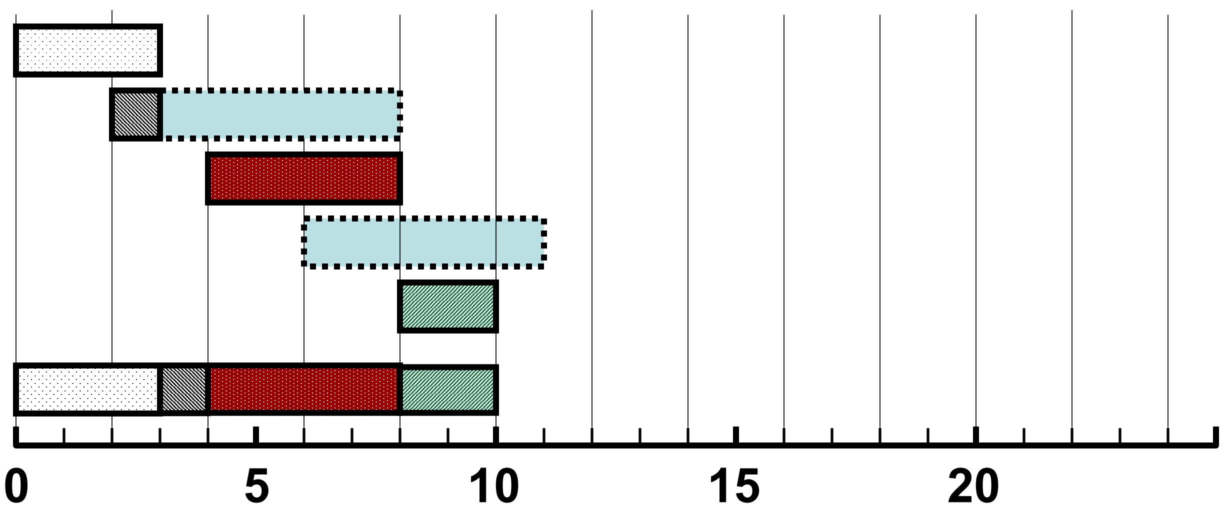
Shortest Remaining Time

Process		Arrival Time	Processing Time	Delay	Turnaround Time
	1	0	3		
	2	2	6		
	3	4	4		
	4	6	5		
	5	8	2		



Shortest Remaining Time

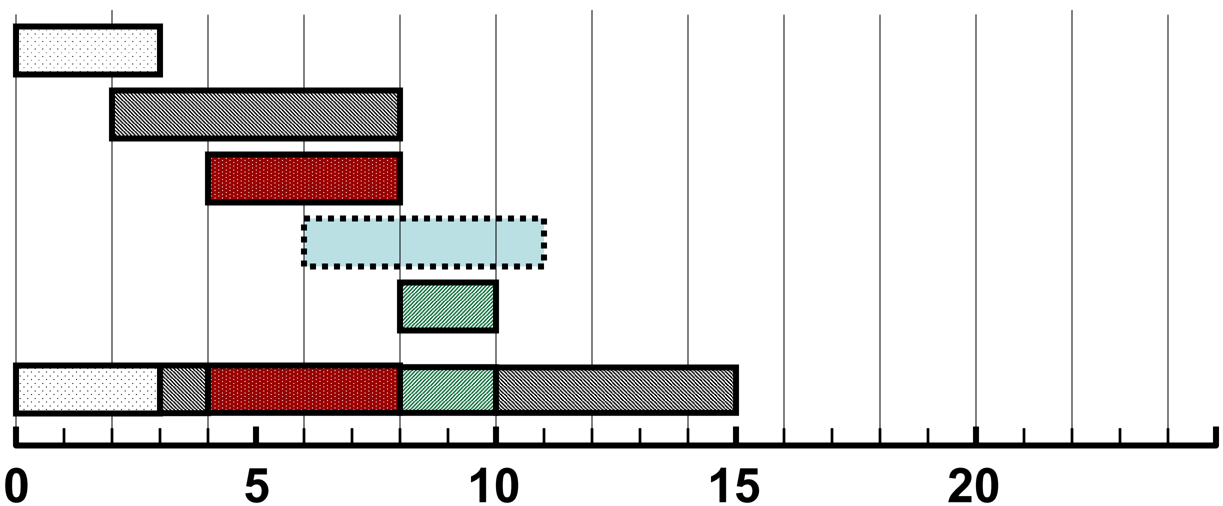
Process	Arrival Time	Processing Time	Delay	Turnaround Time
1	0	3		
2	2	6		
3	4	4		
4	6	5		
5	8	2		



Portland State UNIVERSITY

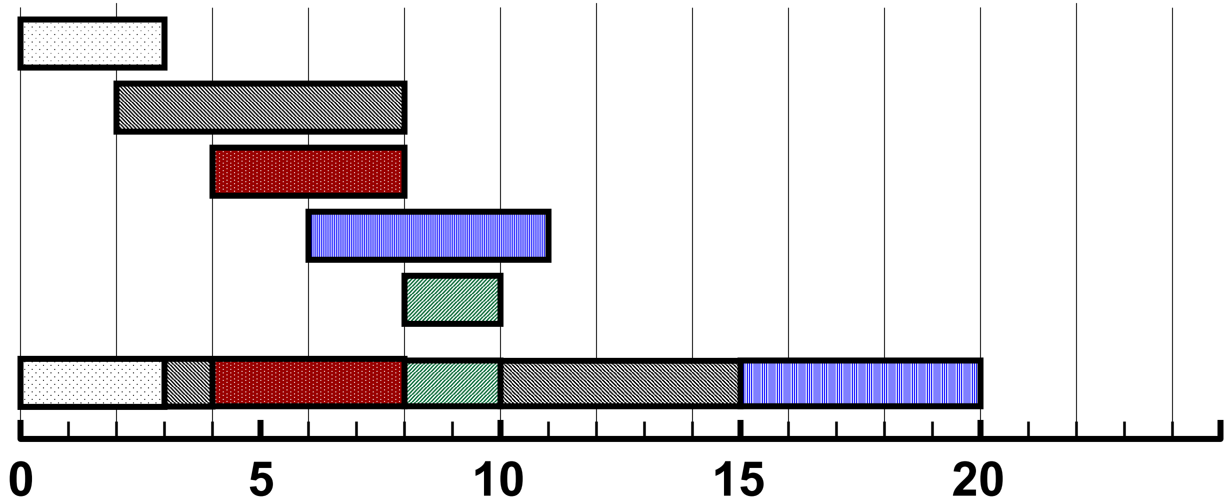
Shortest Remaining Time

Process	Arrival Time	Processing Time	Delay	Turnaround Time
1	0	3		
2	2	6		
3	4	4		
4	6	5		
5	8	2		



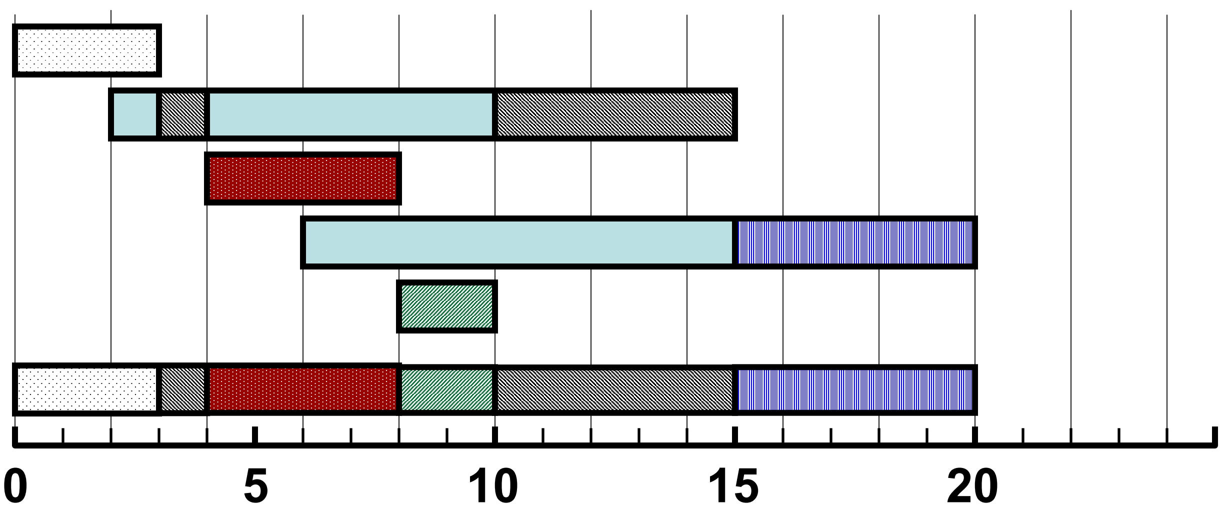
Shortest Remaining Time

Process		Arrival Time	Processing Time	Delay	Turnaround Time
	1	0	3		
	2	2	6		
	3	4	4		
	4	6	5		
	5	8	2		



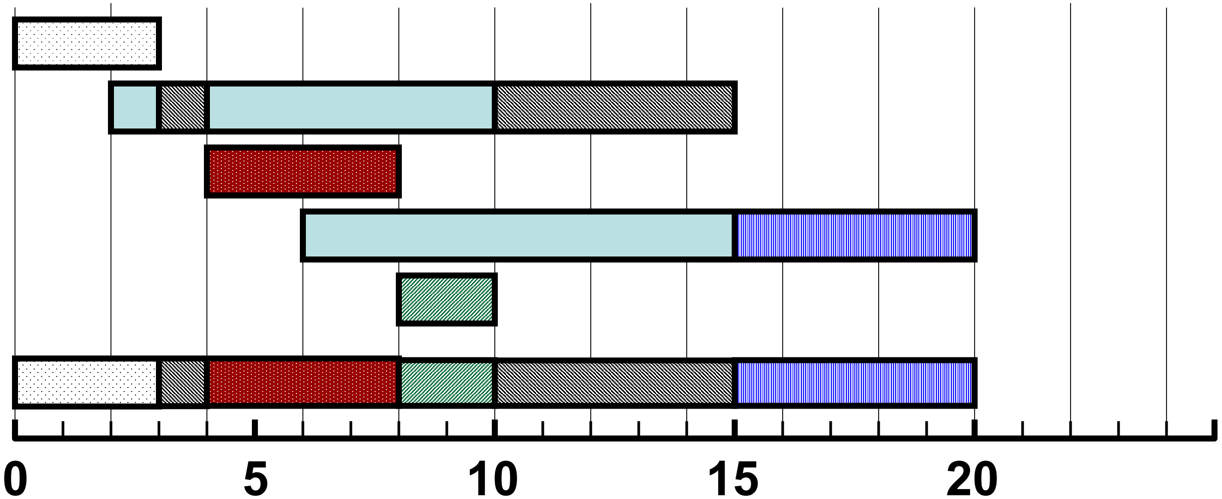
Shortest Remaining Time

Process	Arrival Time	Processing Time	Delay	Turnaround Time
1	0	3		
2	2	6		
3	4	4		
4	6	5		
5	8	2		



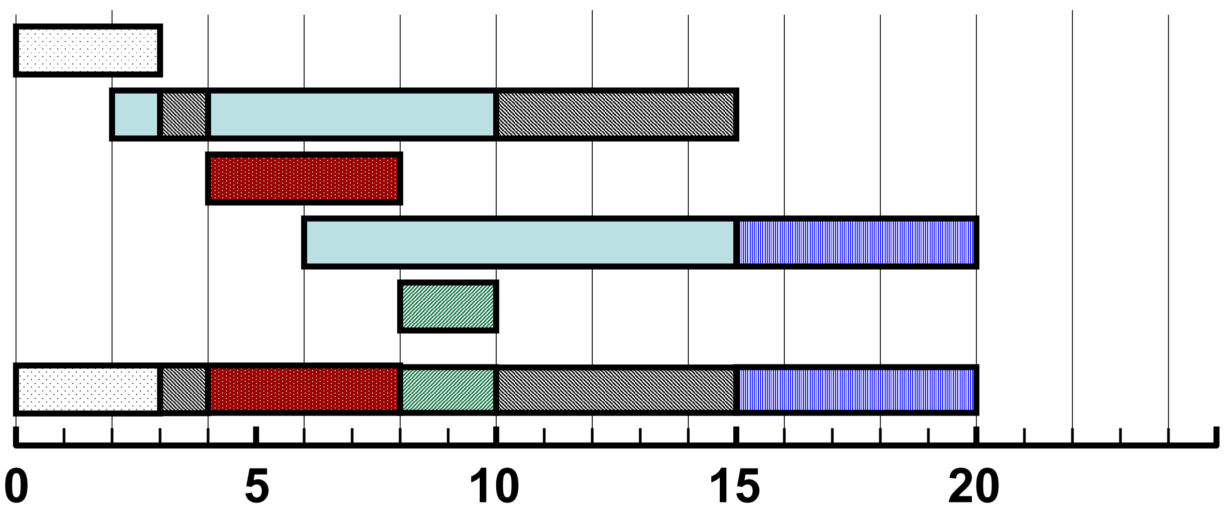
Shortest Remaining Time

Process	Arrival Time	Processing Time	Delay	Turnaround Time
1	0	3	0	
2	2	6	7	
3	4	4	0	
4	6	5	9	
5	8	2	0	



Shortest Remaining Time

Process	Arrival Time	Processing Time	Delay	Turnaround Time
1	0	3	0	3
2	2	6	7	13
3	4	4	0	4
4	6	5	9	14
5	8	2	0	2



Round-Robin Scheduling

Goal: Enable interactivity

Limit the amount of CPU that a thread can have at one time.

Time quantum

Amount of time the OS gives a thread before intervention

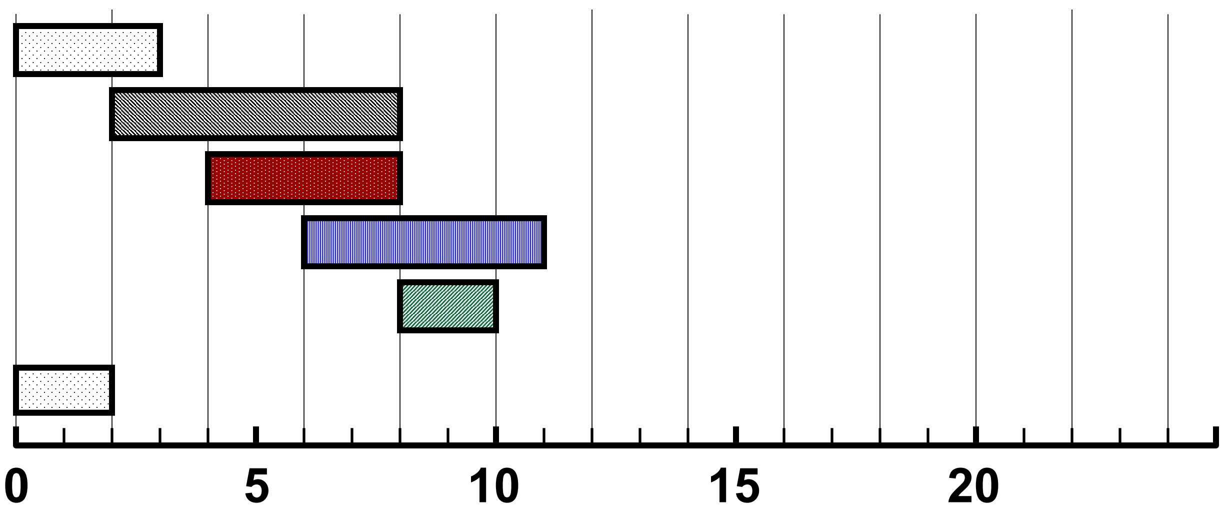
Sometimes called the “time slice”

Typically: 1 to 100ms

Not necessarily the same as the timer interrupt frequency!

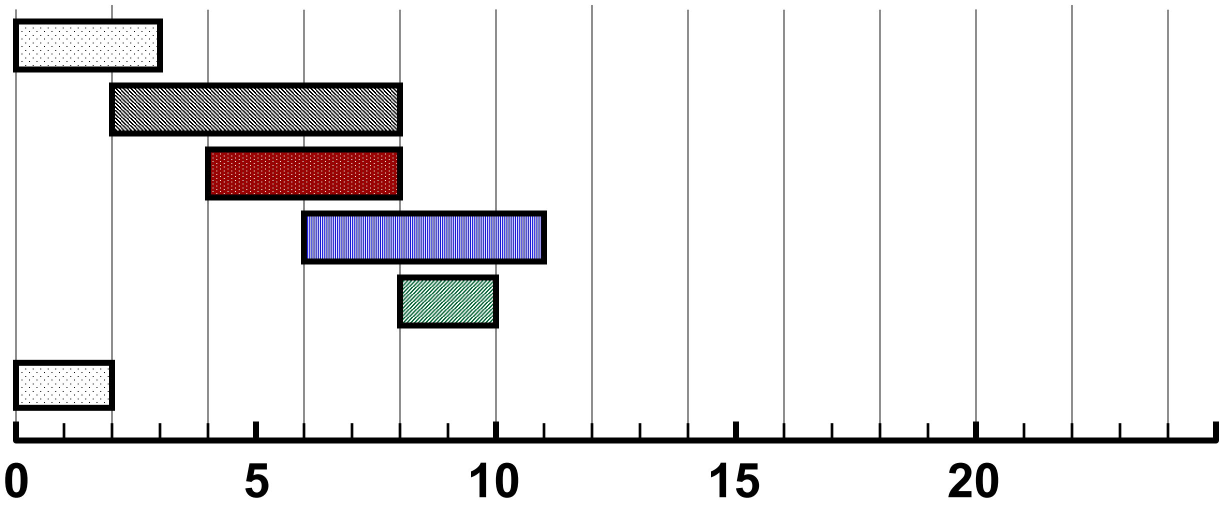
Round-Robin Scheduling

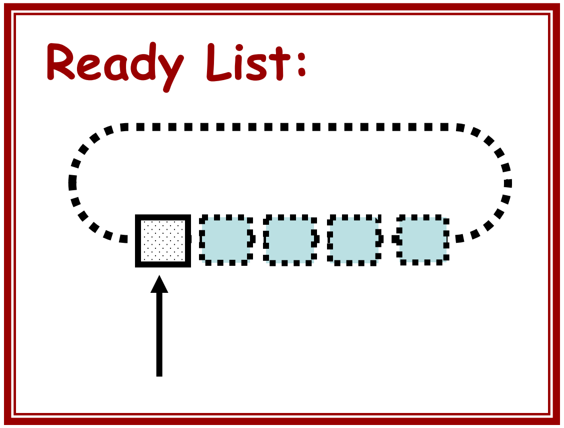
Process	Arrival Time	Processing Time
1	0	3
2	2	6
3	4	4
4	6	5
5	8	2



Round-Robin Scheduling

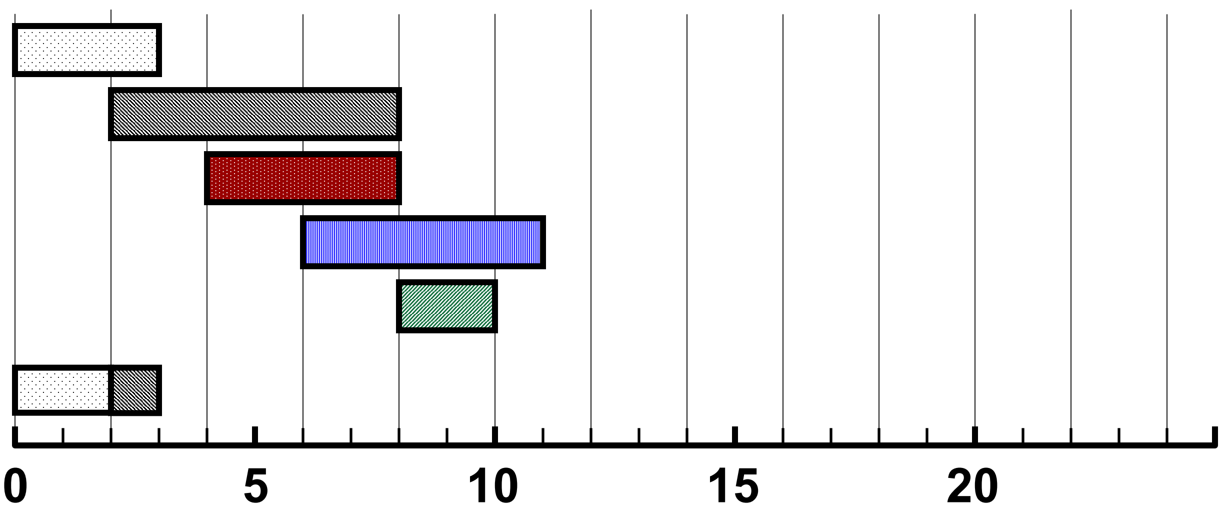
Process	Arrival Time	Processing Time
1	0	3
2	2	6
3	4	4
4	6	5
5	8	2

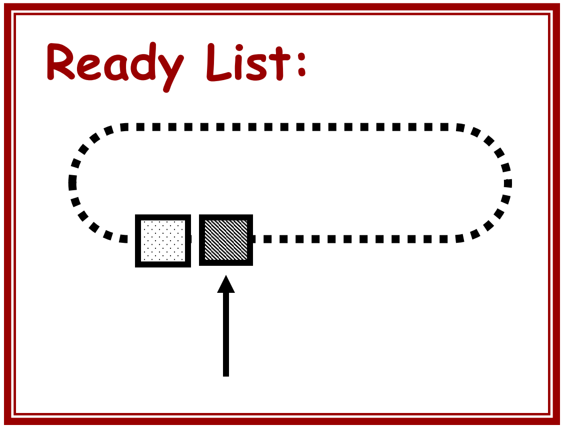




Round-Robin Scheduling

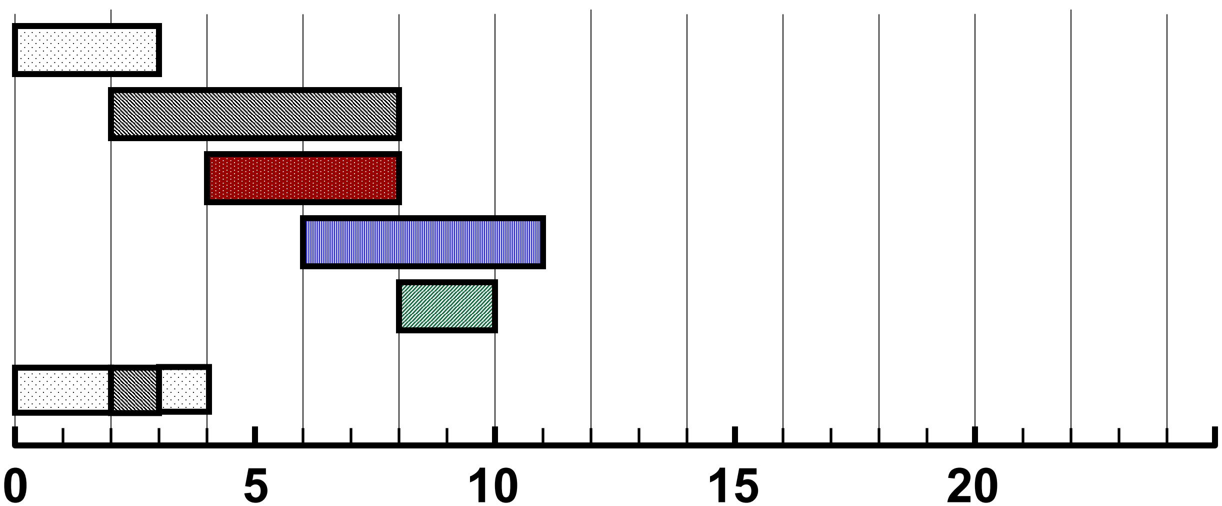
Process	Arrival Time	Processing Time
1	0	3
2	2	6
3	4	4
4	6	5
5	8	2

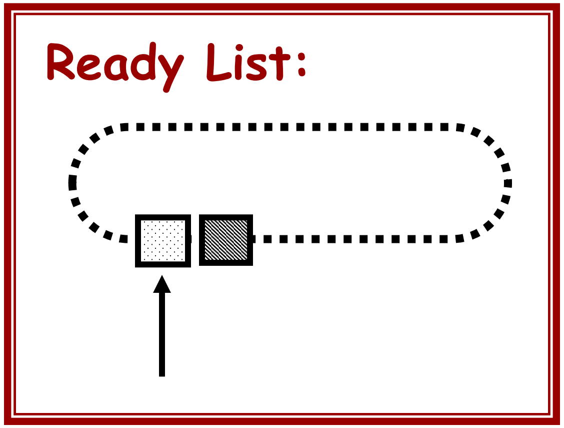




Round-Robin Scheduling

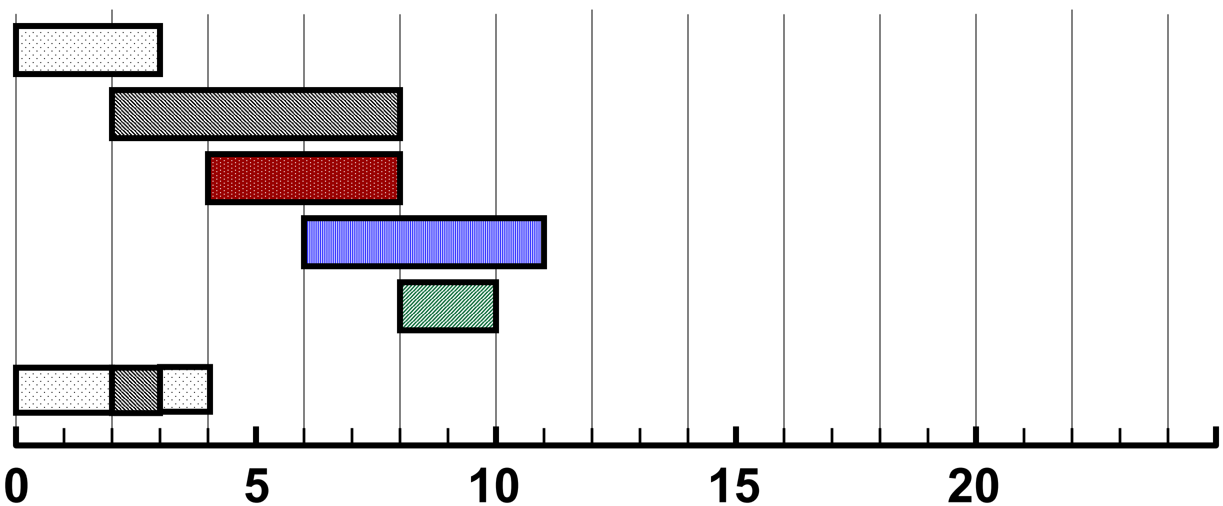
Process		Arrival Time	Processing Time
	1	0	3
	2	2	6
	3	4	4
	4	6	5
	5	8	2

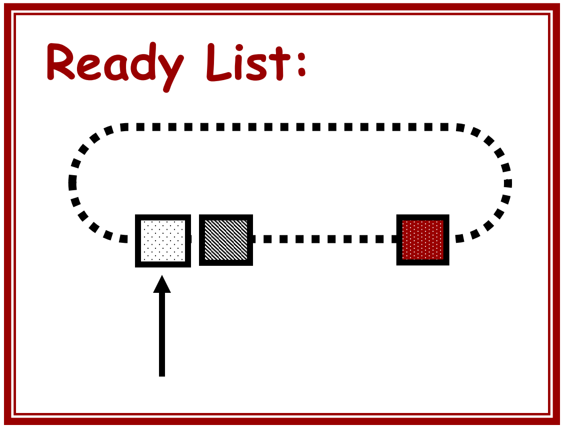




Round-Robin Scheduling

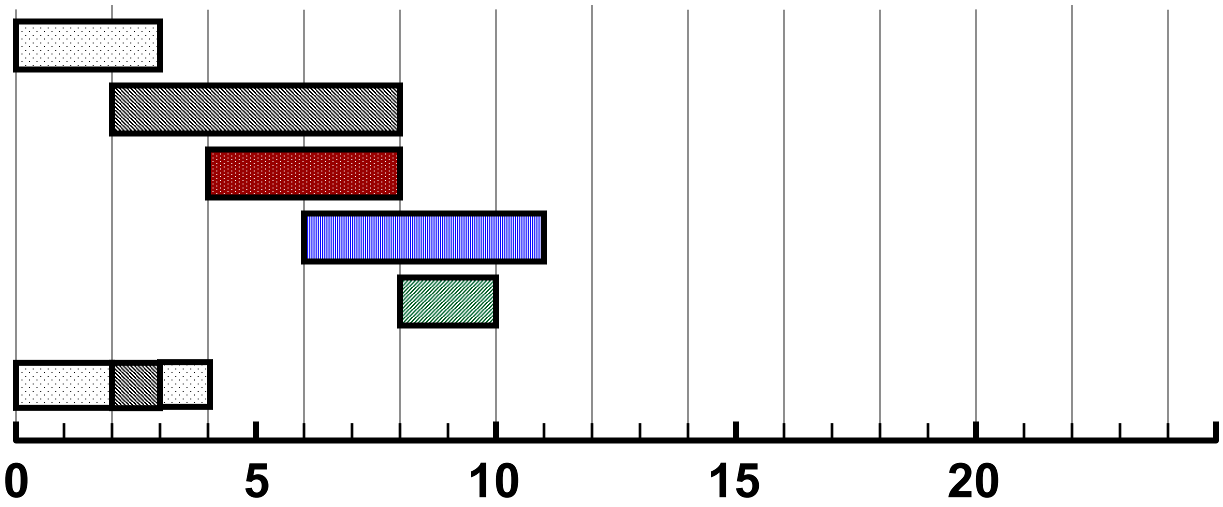
Process	Arrival Time	Processing Time
1	0	3
2	2	6
3	4	4
4	6	5
5	8	2

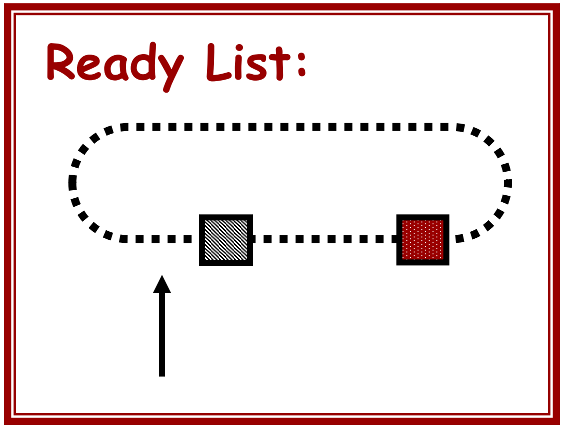




Round-Robin Scheduling

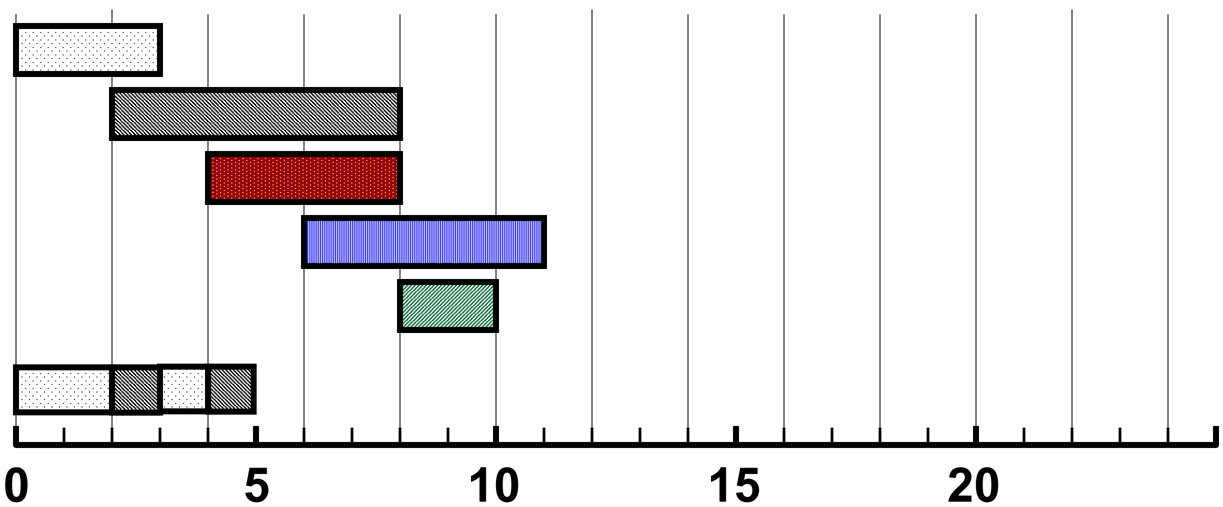
Process	Arrival Time	Processing Time
1	0	3
2	2	6
3	4	4
4	6	5
5	8	2

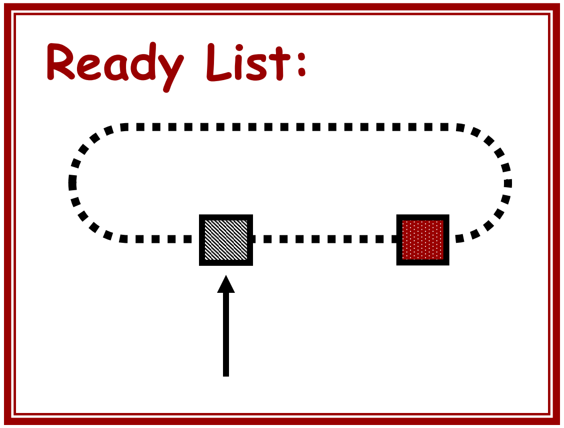




Round-Robin Scheduling

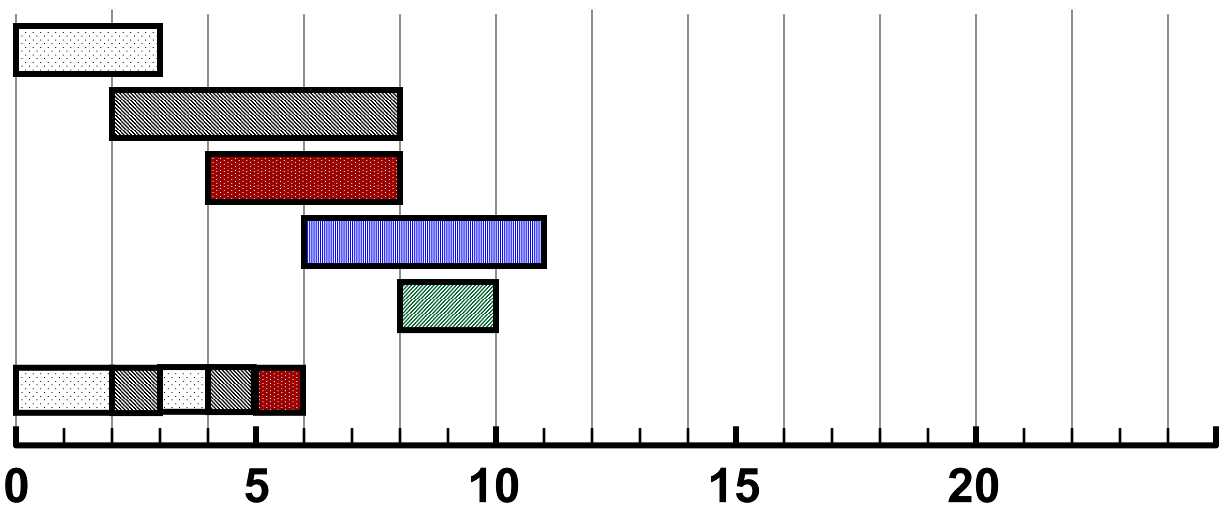
Process	Arrival Time	Processing Time
1	0	3
2	2	6
3	4	4
4	6	5
5	8	2

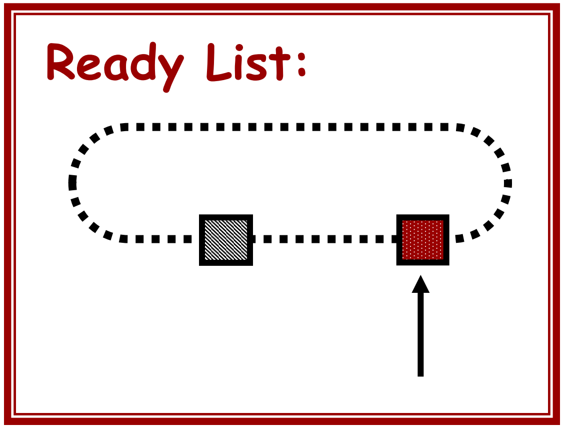




Round-Robin Scheduling

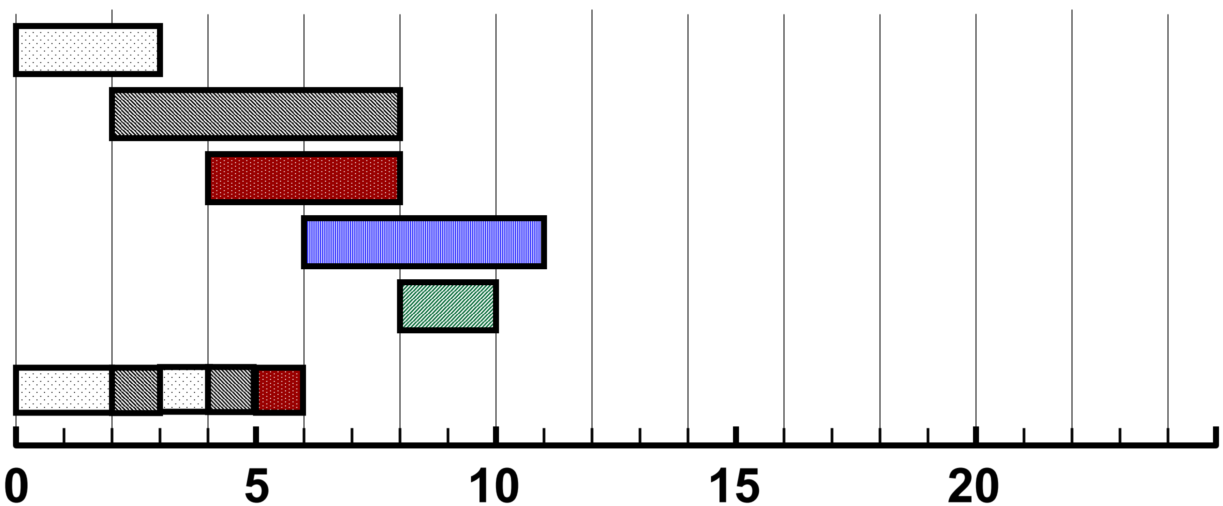
Process	Arrival Time	Processing Time
1	0	3
2	2	6
3	4	4
4	6	5
5	8	2

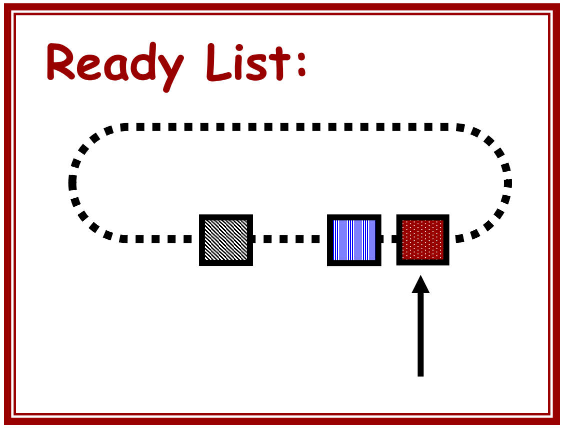




Round-Robin Scheduling

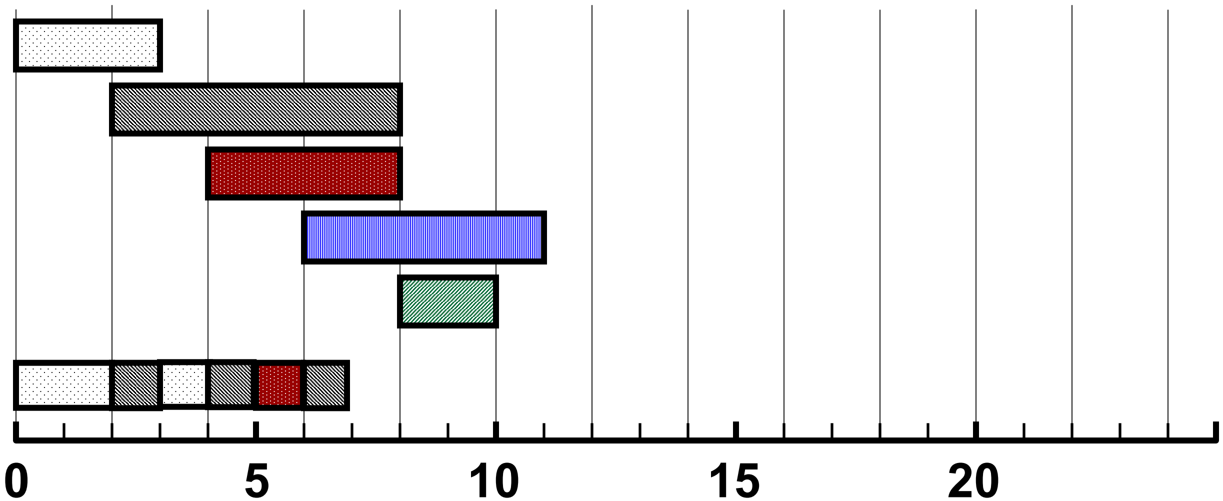
	Process	Arrival Time	Processing Time
	1	0	3
	2	2	6
	3	4	4
	4	6	5
	5	8	2

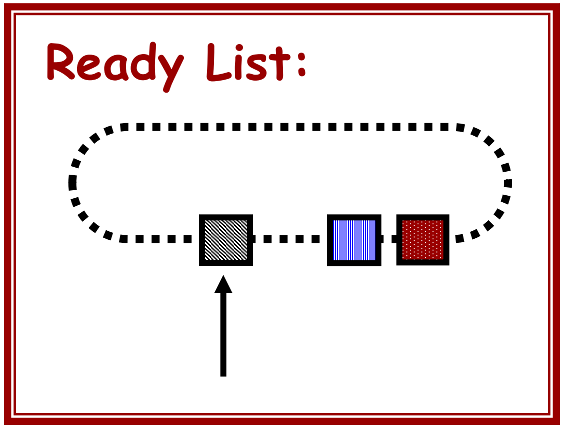




Round-Robin Scheduling

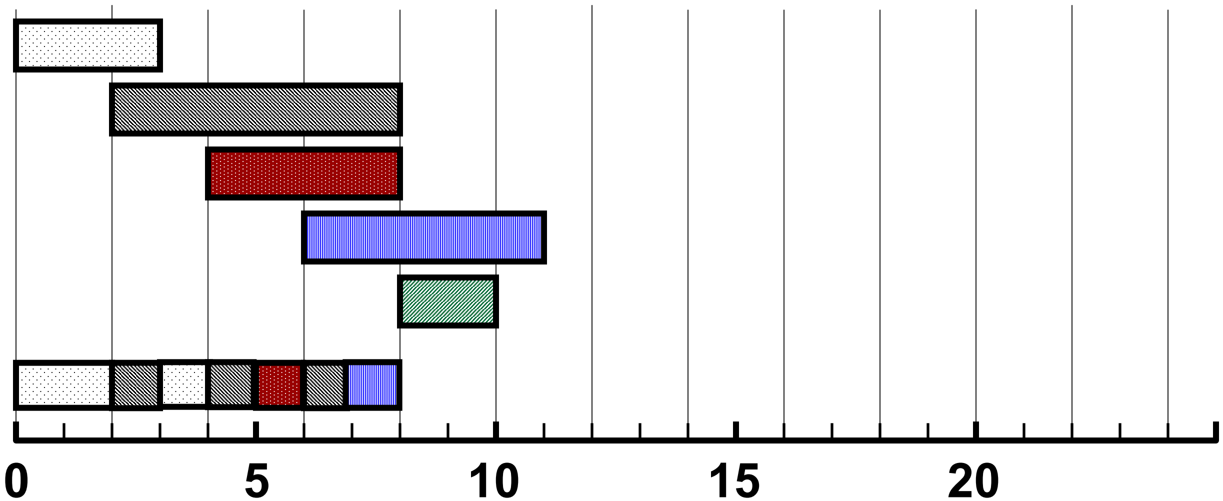
Process	Arrival Time	Processing Time
1	0	3
2	2	6
3	4	4
4	6	5
5	8	2

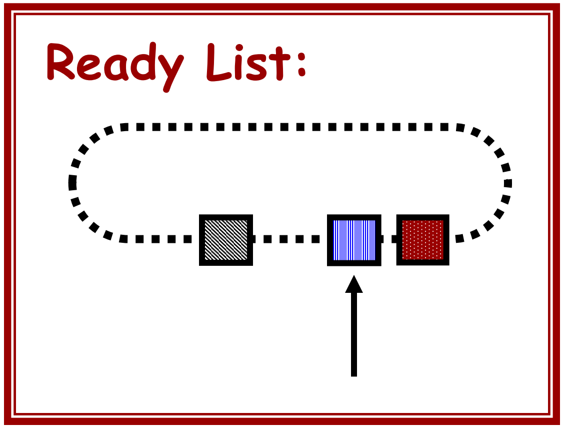




Round-Robin Scheduling

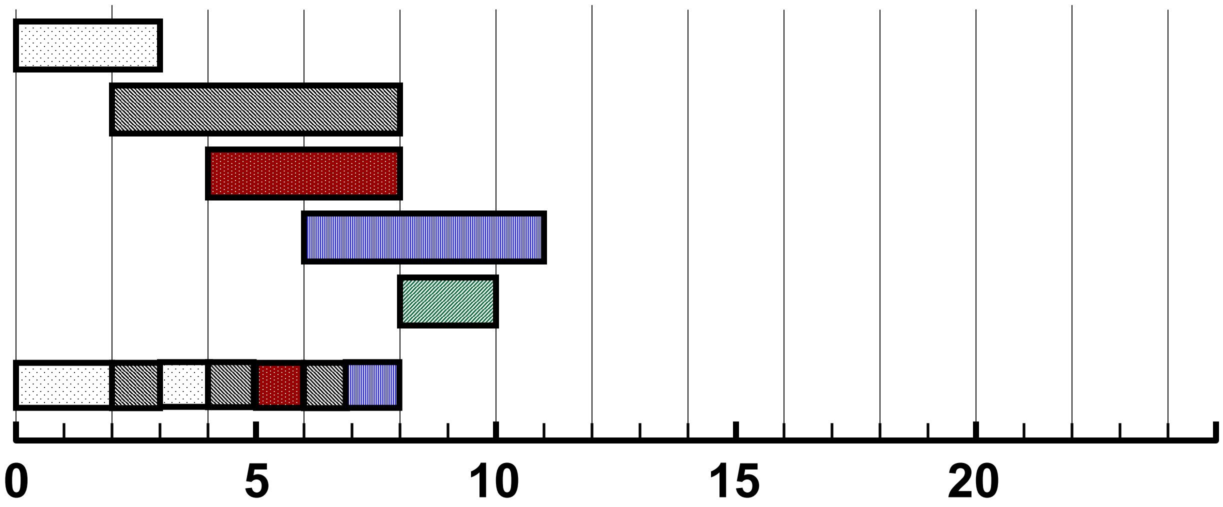
Process	Arrival Time	Processing Time
1	0	3
2	2	6
3	4	4
4	6	5
5	8	2

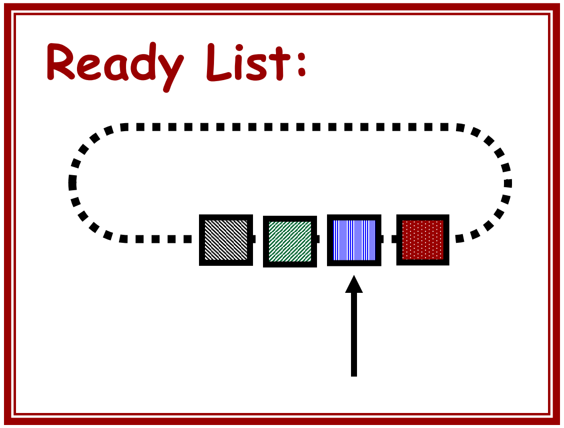




Round-Robin Scheduling

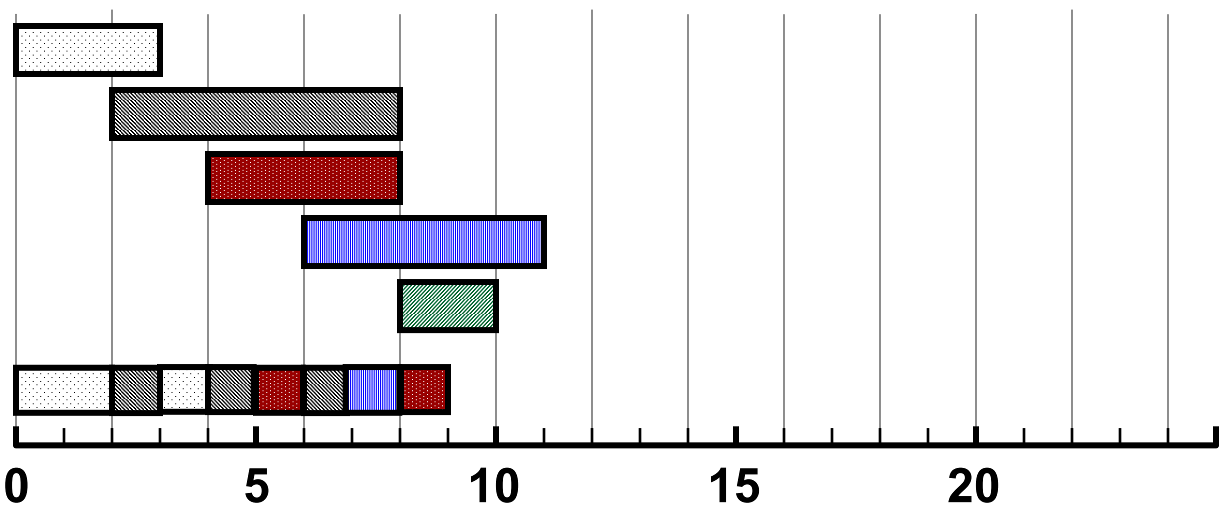
Process	Arrival Time	Processing Time
1	0	3
2	2	6
3	4	4
4	6	5
5	8	2

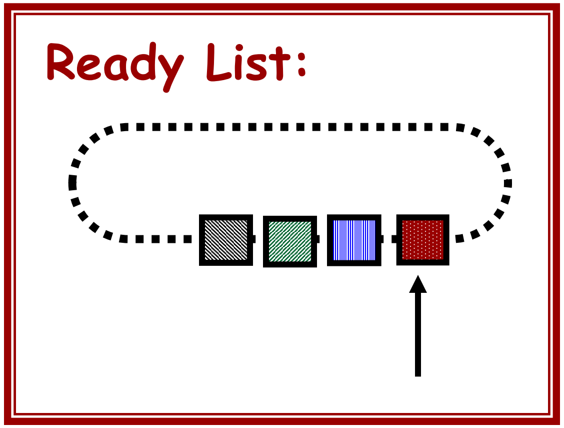




Round-Robin Scheduling

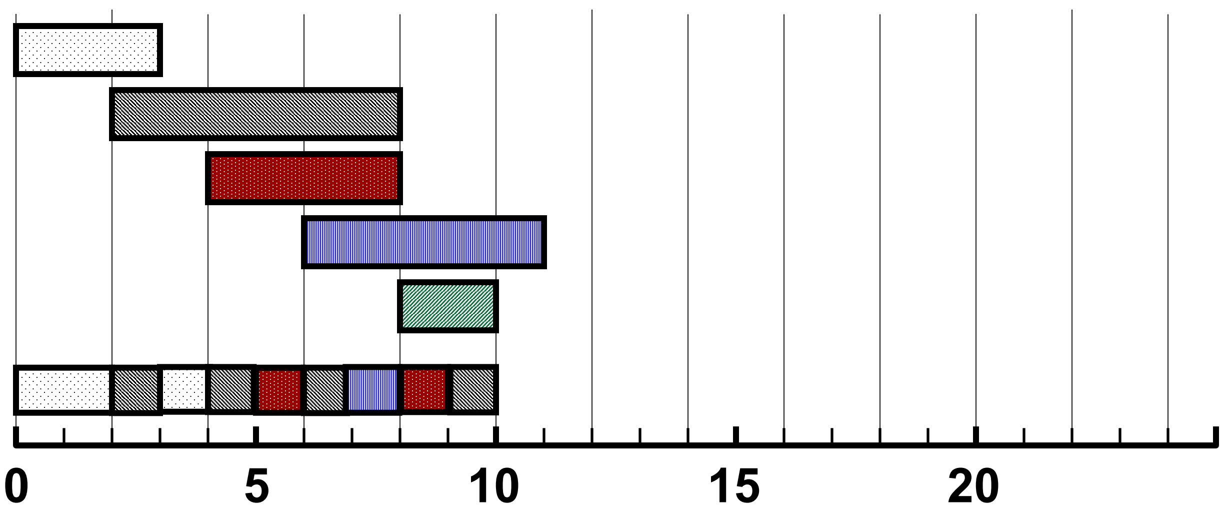
	Process	Arrival Time	Processing Time
	1	0	3
	2	2	6
	3	4	4
	4	6	5
	5	8	2

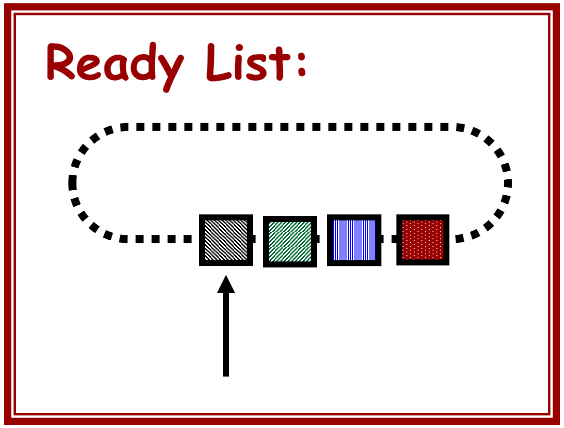




Round-Robin Scheduling

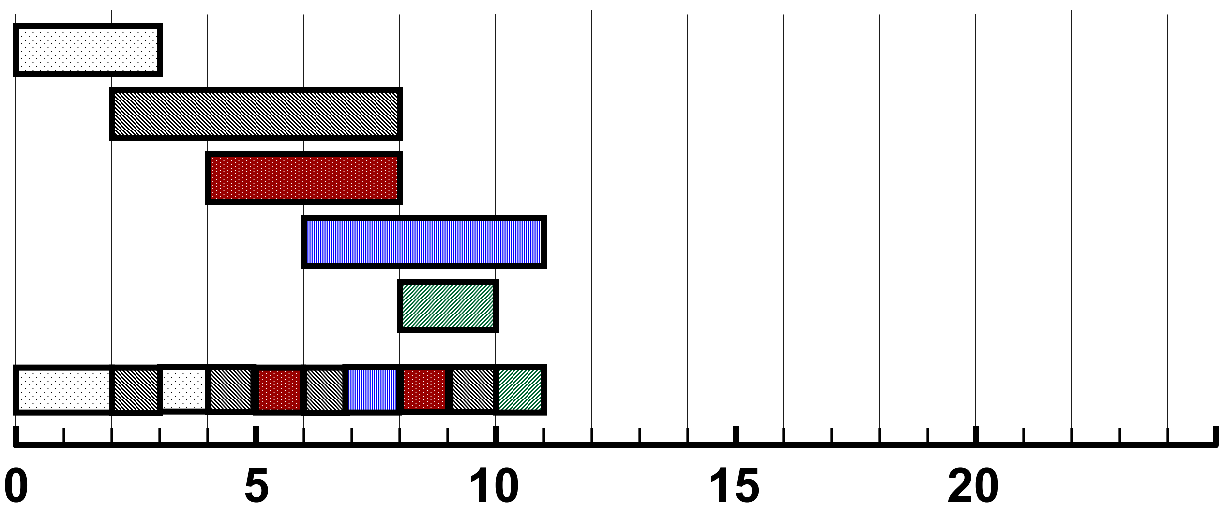
Process		Arrival Time	Processing Time
	1	0	3
	2	2	6
	3	4	4
	4	6	5
	5	8	2

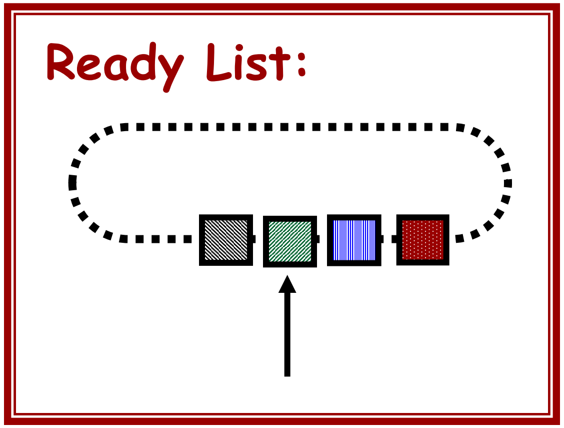




Round-Robin Scheduling

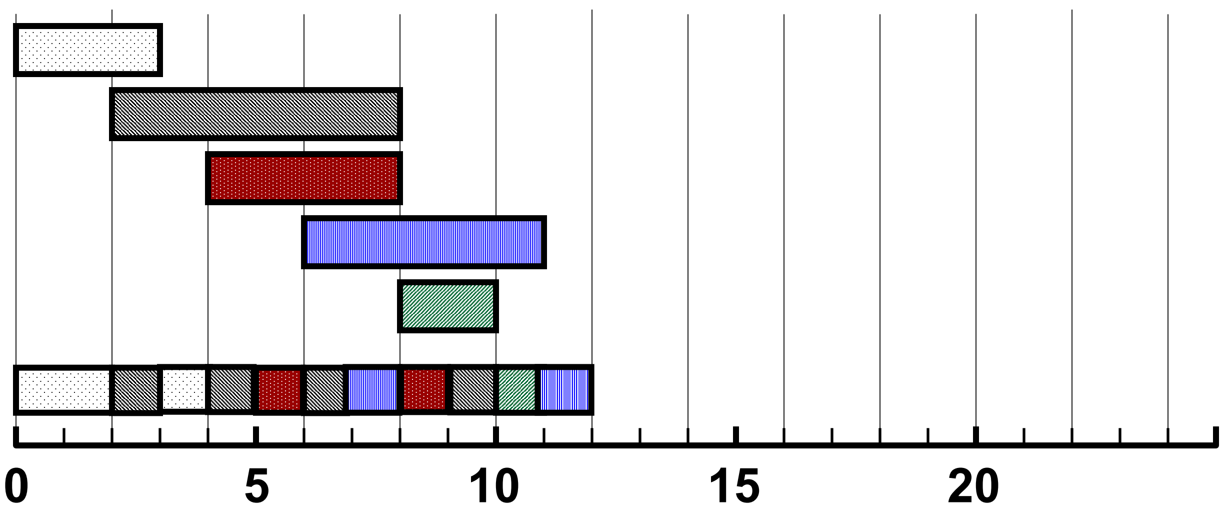
Process	Arrival Time	Processing Time
1	0	3
2	2	6
3	4	4
4	6	5
5	8	2

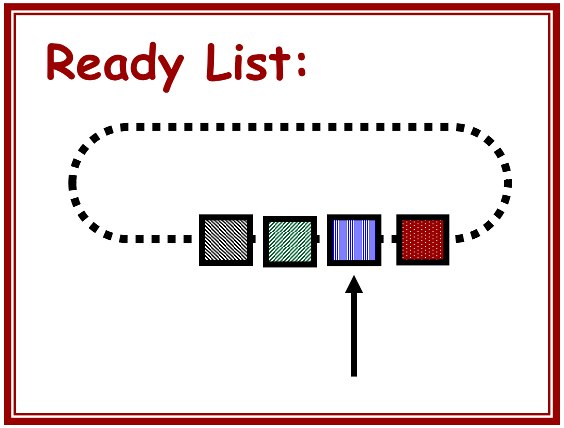




Round-Robin Scheduling

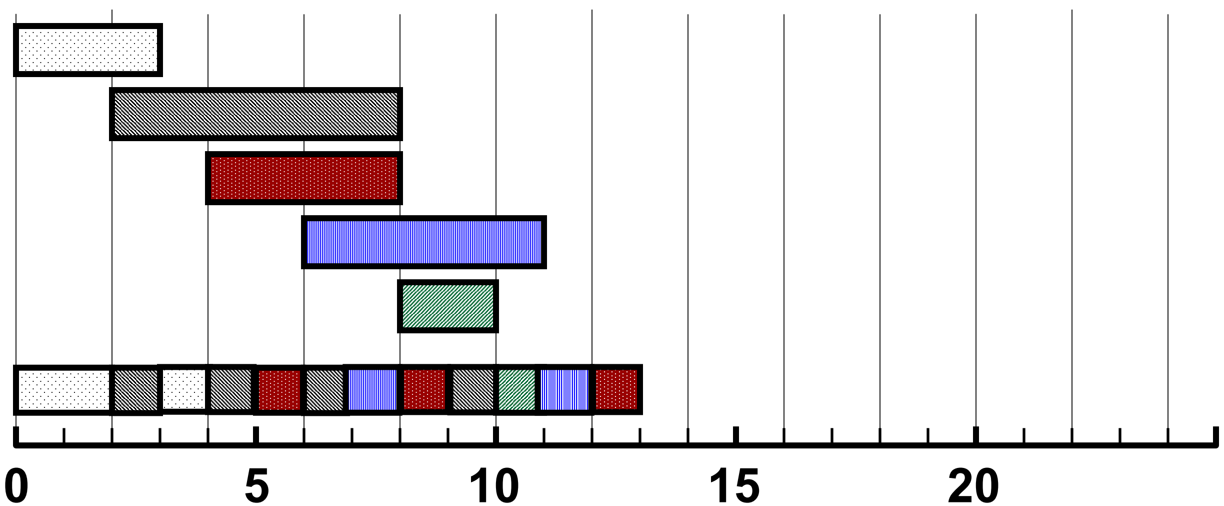
Process		Arrival Time	Processing Time
	1	0	3
	2	2	6
	3	4	4
	4	6	5
	5	8	2

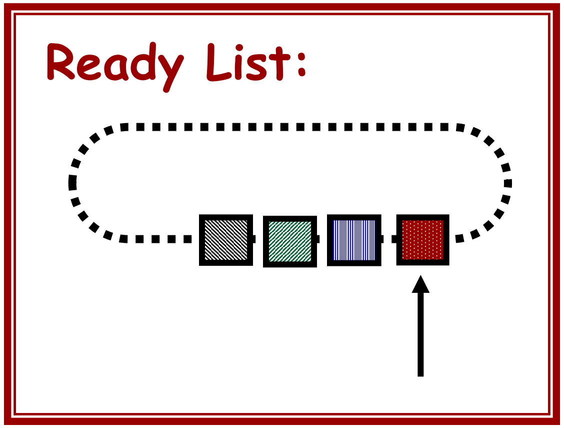




Round-Robin Scheduling

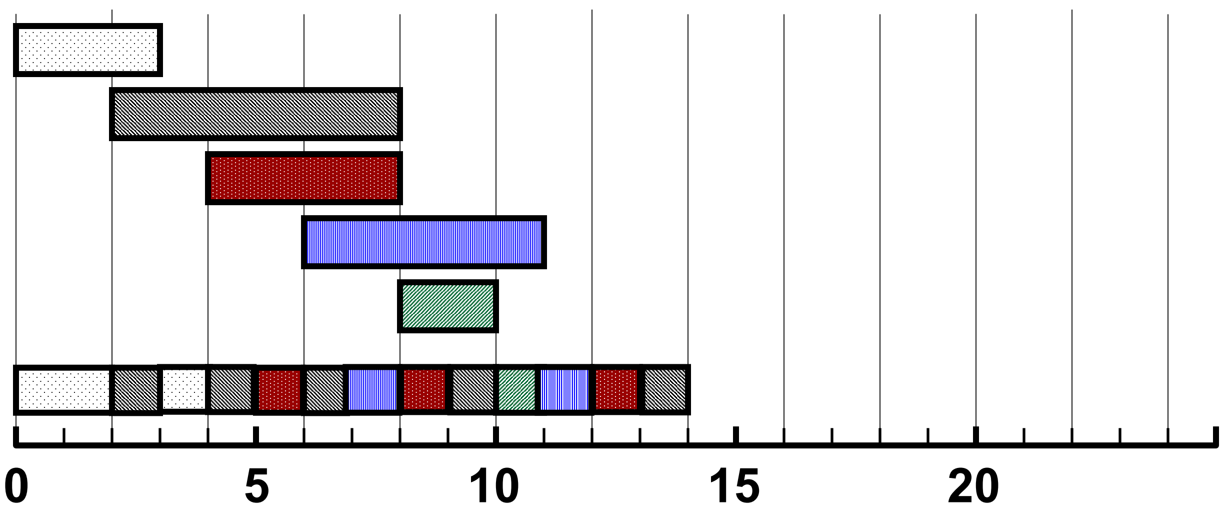
Process	Arrival Time	Processing Time
1	0	3
2	2	6
3	4	4
4	6	5
5	8	2

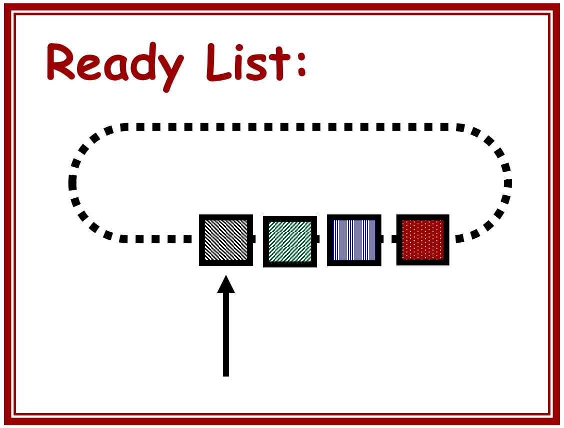




Round-Robin Scheduling

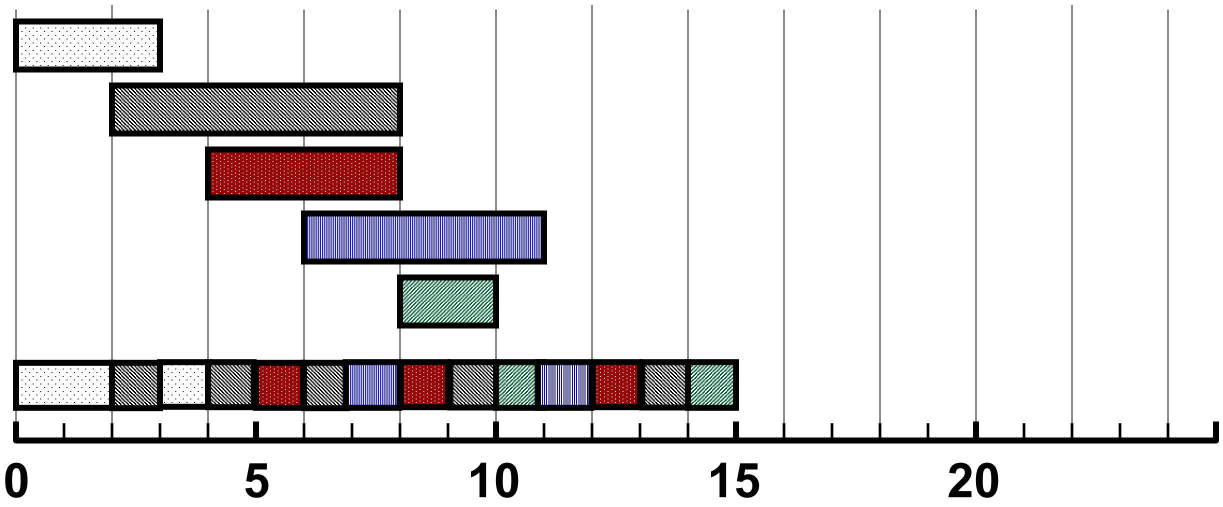
Process	Arrival Time	Processing Time
1	0	3
2	2	6
3	4	4
4	6	5
5	8	2

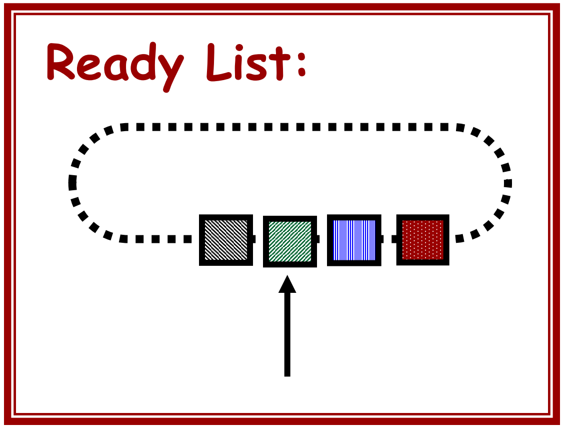




Round-Robin Scheduling

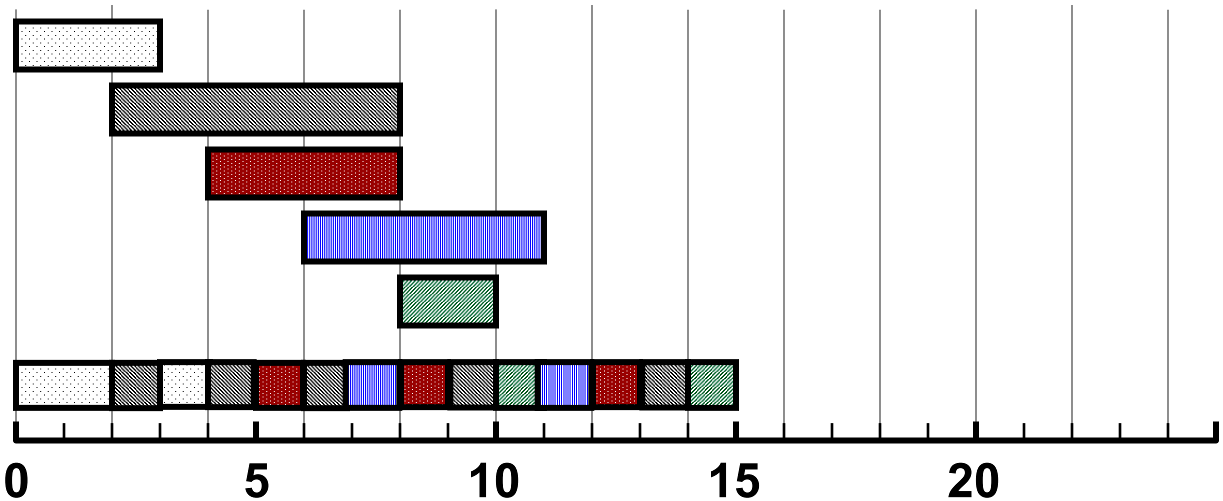
	Process	Arrival Time	Processing Time
	1	0	3
	2	2	6
	3	4	4
	4	6	5
	5	8	2

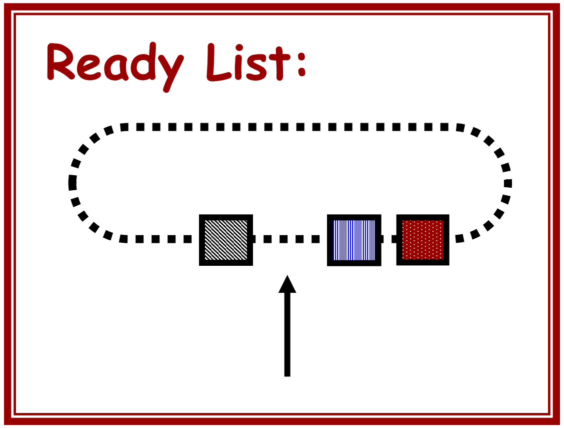




Round-Robin Scheduling

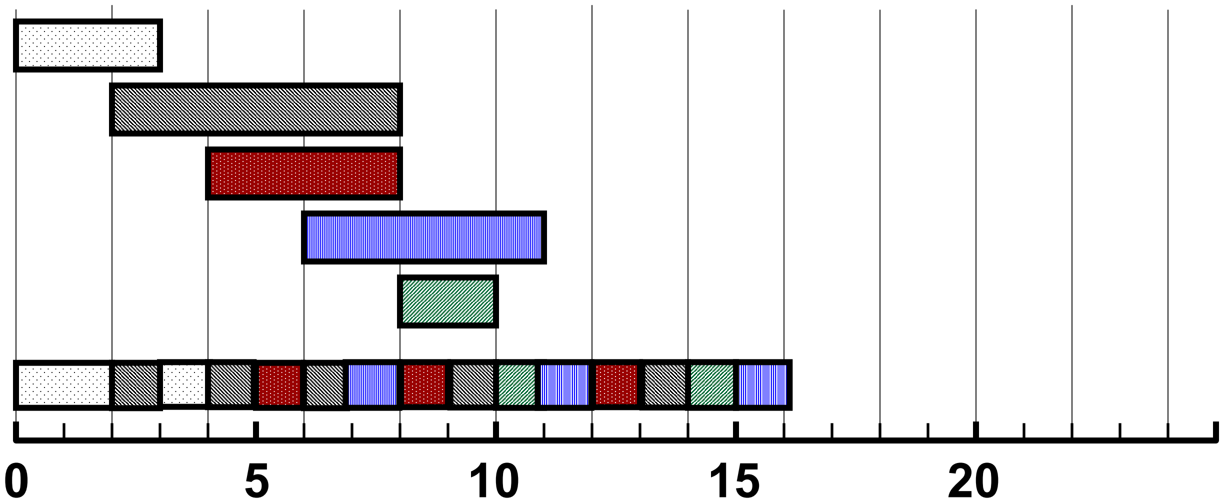
Process		Arrival Time	Processing Time
	1	0	3
	2	2	6
	3	4	4
	4	6	5
	5	8	2

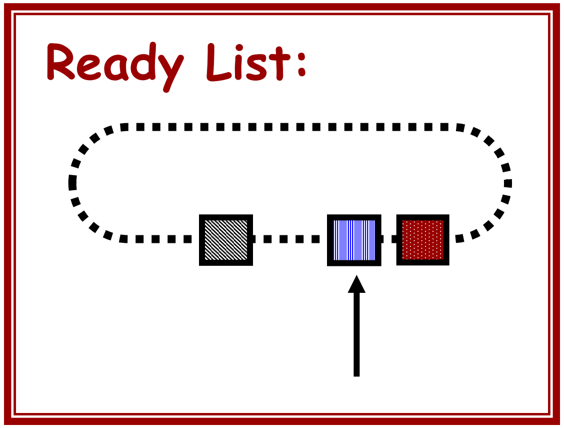




Round-Robin Scheduling

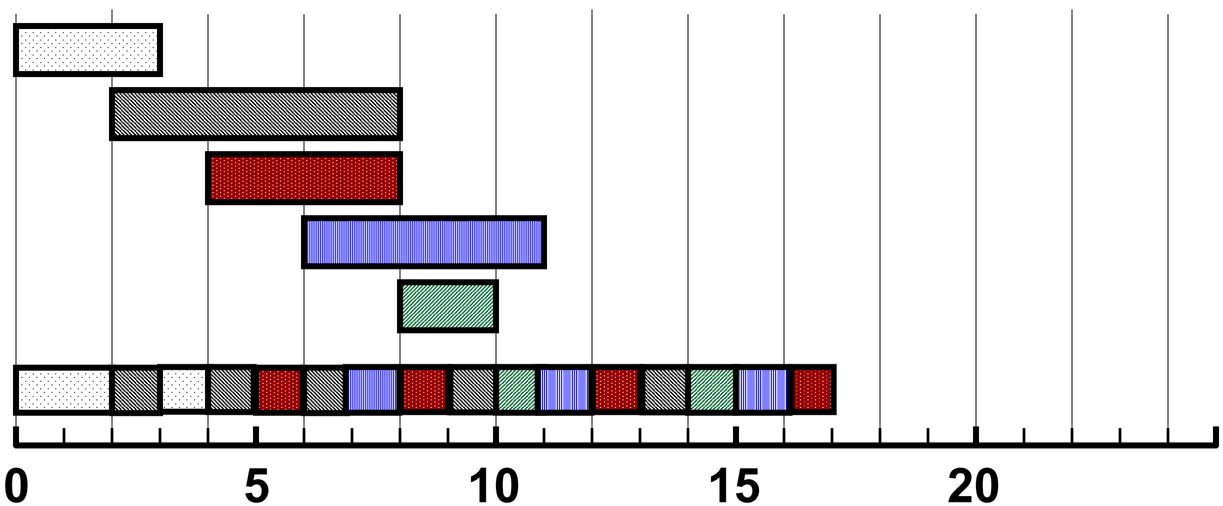
Process	Arrival Time	Processing Time
1	0	3
2	2	6
3	4	4
4	6	5
5	8	2

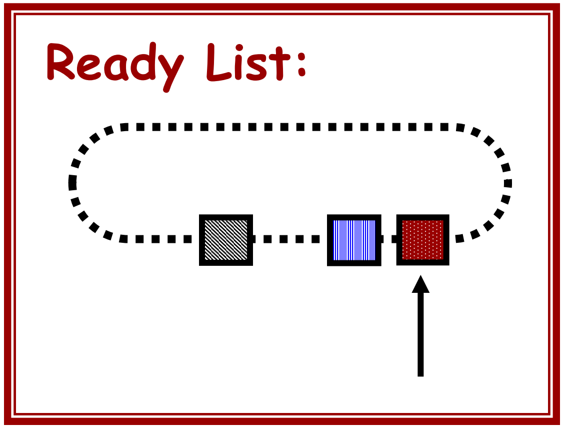




Round-Robin Scheduling

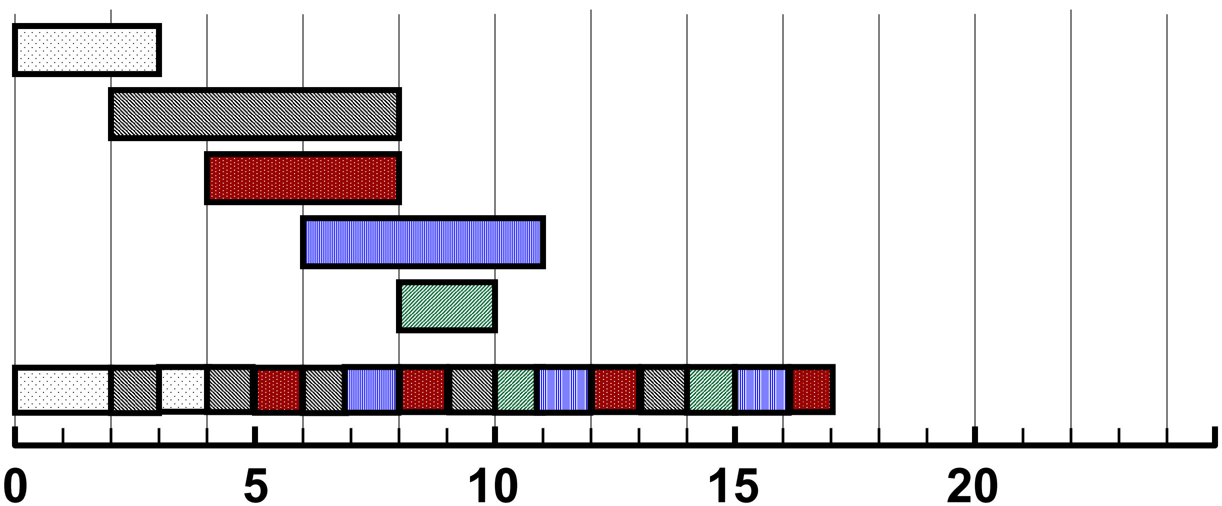
	Process	Arrival Time	Processing Time
	1	0	3
	2	2	6
	3	4	4
	4	6	5
	5	8	2

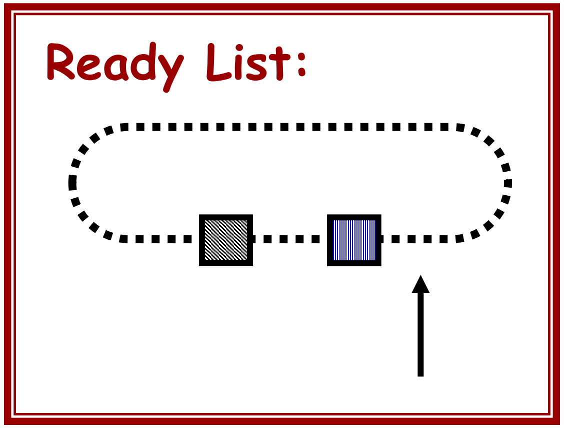




Round-Robin Scheduling

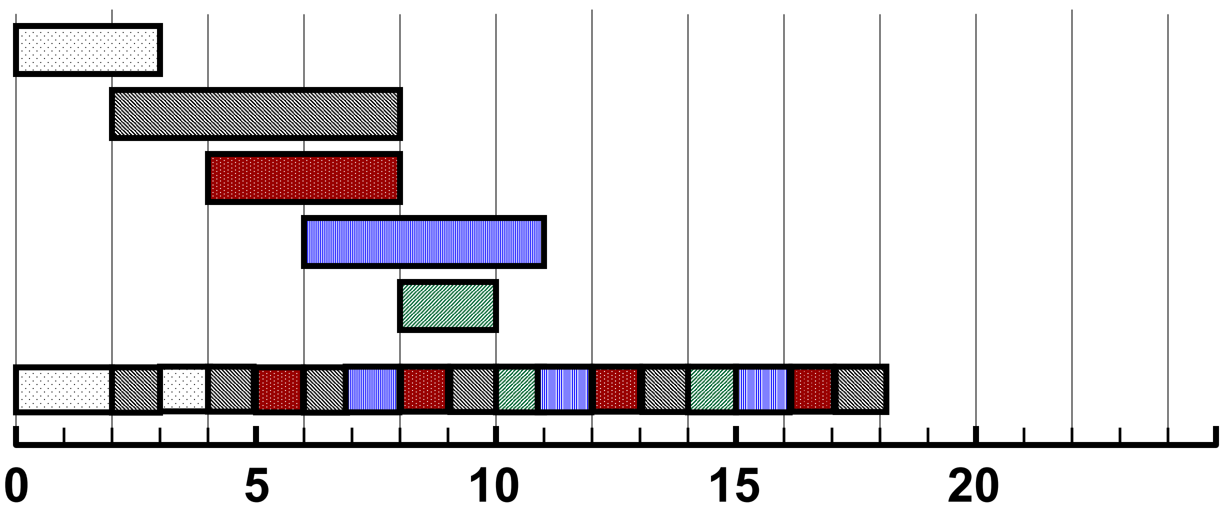
Process		Arrival Time	Processing Time
▢	1	0	3
▢	2	2	6
▢	3	4	4
▢	4	6	5
▢	5	8	2

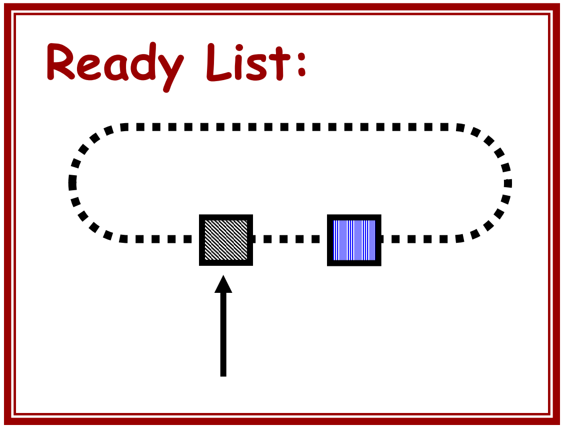




Round-Robin Scheduling

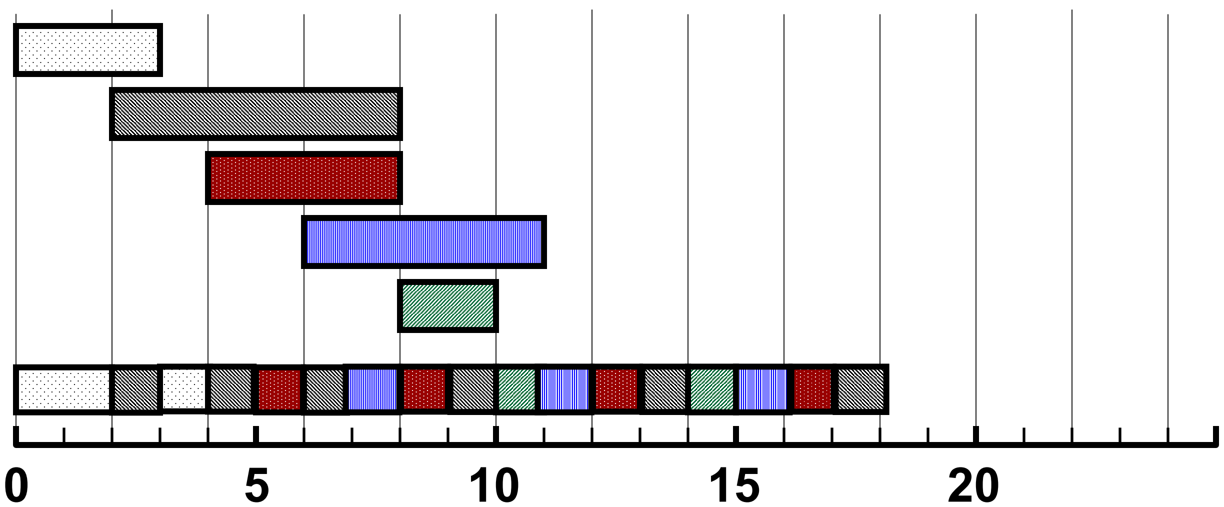
Process	Arrival Time	Processing Time
1	0	3
2	2	6
3	4	4
4	6	5
5	8	2

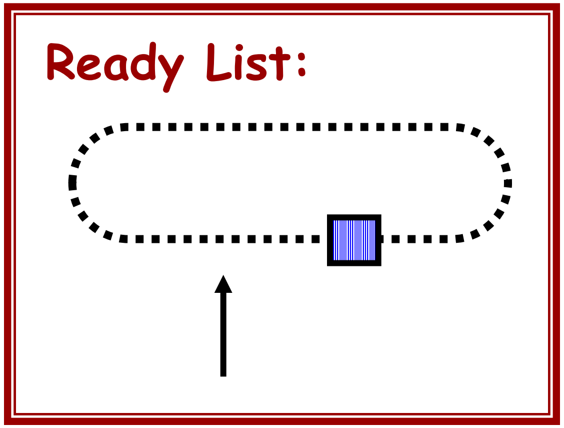




Round-Robin Scheduling

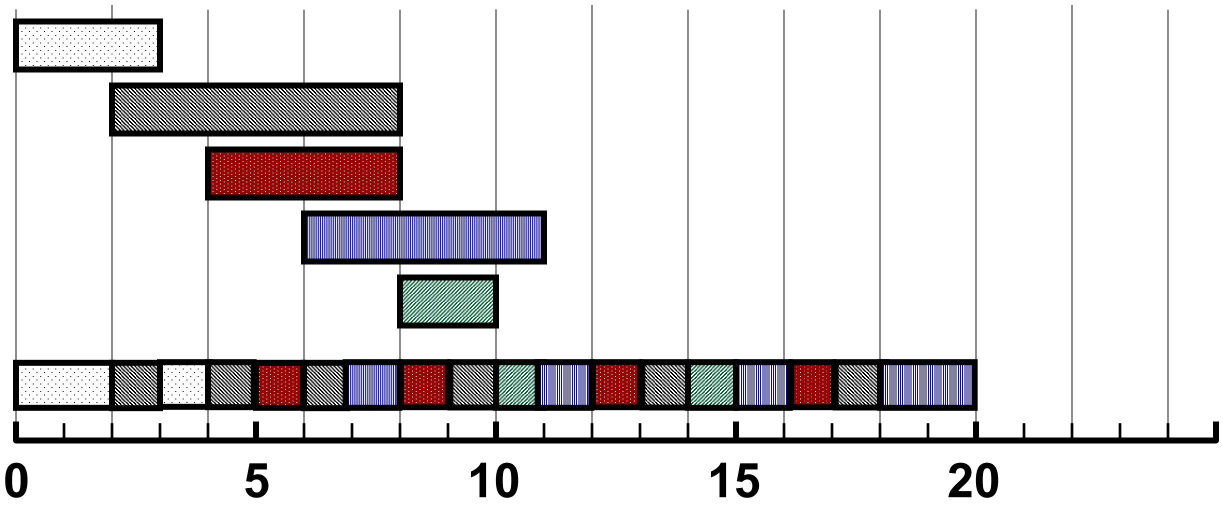
Process		Arrival Time	Processing Time
	1	0	3
	2	2	6
	3	4	4
	4	6	5
	5	8	2

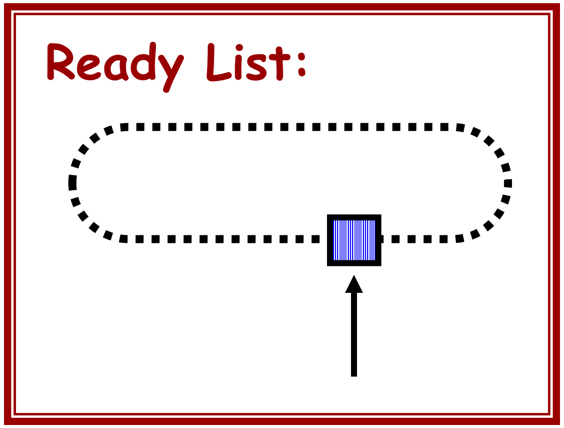




Round-Robin Scheduling

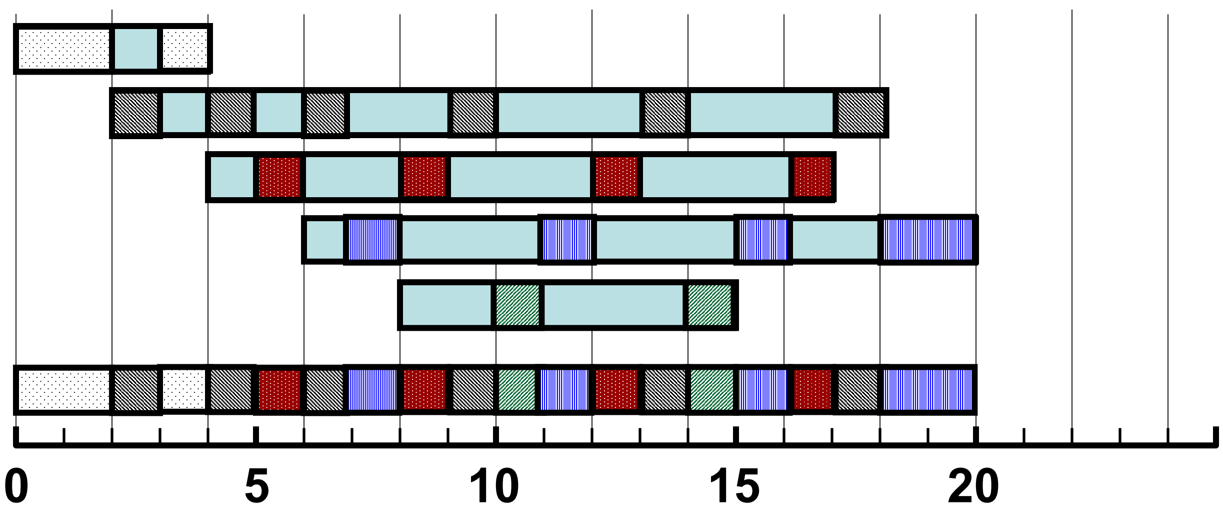
Process		Arrival Time	Processing Time
	1	0	3
	2	2	6
	3	4	4
	4	6	5
	5	8	2





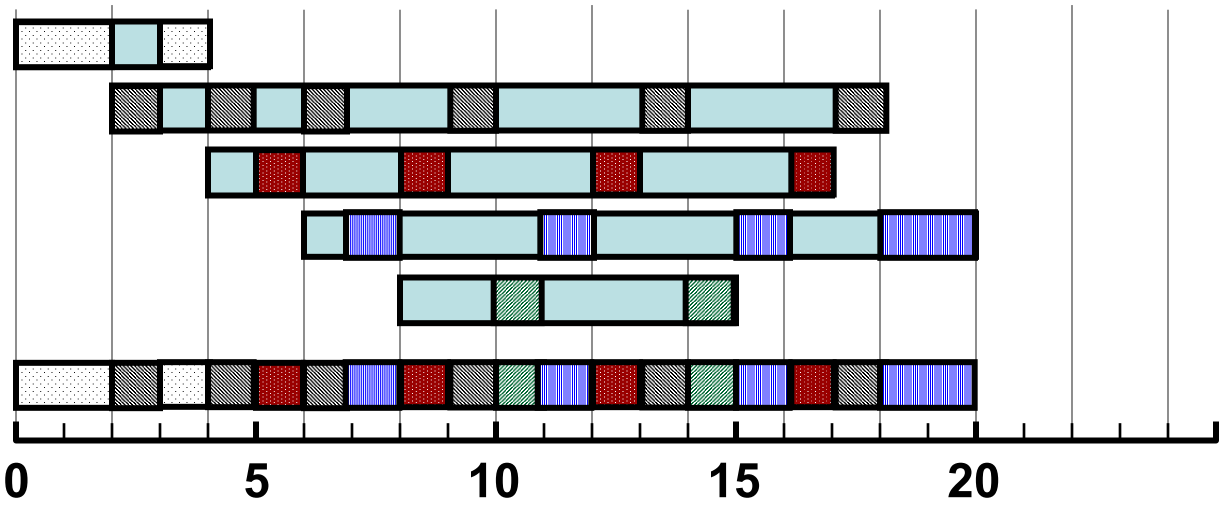
Round-Robin Scheduling

Process	Arrival Time	Processing Time
1	0	3
2	2	6
3	4	4
4	6	5
5	8	2



Round-Robin Scheduling

Process	Arrival Time	Processing Time	Delay	Turnaround Time
1	0	3	1	4
2	2	6	10	16
3	4	4	9	13
4	6	5	9	14
5	8	2	5	7



Round-Robin Scheduling

Effectiveness of round-robin depends on

- The number of threads, and
- The size of the time quantum.

Large # of threads means that the time between scheduling of a single thread increases

- Slow responses

Larger time quantum means that the time between the scheduling of a single thread also increases

- Slow responses

Smaller time quantum means higher processing rates but also more overhead!

General Purpose Schedulers

Priority Scheduling

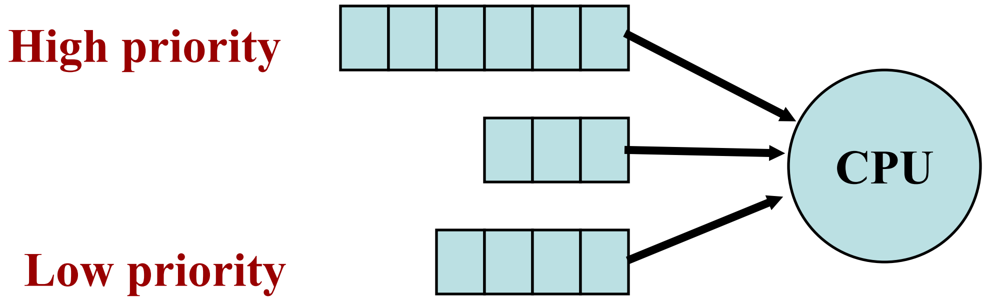
Assign a priority (number) to each thread
Schedule threads based on their priority
Higher priority threads get more CPU time
Starvation is possible!

Managing priorities

- Can use “nice” to voluntarily reduce your priority
- Scheduler can periodically adjust a process’ priority
  - Prevents starvation of a lower priority process
  - Can improve performance of I/O-bound processes by basing priority on fraction of last quantum used

Multi-Level Queue Scheduling

Multiple queues, each with its own priority
Equivalently: each priority level has its own ready queue
Round-robin scheduling is used within each queue
Simplest approach uses statically assigned priorities



Multi-Level Feedback Scheduling

Problem: with fixed priorities I/O-bound processes are disadvantaged

Can we fix this with dynamic priorities?

Multi-Level Feedback Scheduling

Solution: Let the amount of CPU that a thread used in the last quantum determine its scheduling priority for the next

- Expired time quantum → decrease priority
- Unexpired time quantum → increase priority

Rationale: if a thread didn't use all of its time (because it blocked for I/O) the system owes it some more CPU time

Also, if a thread is compute-bound, raising its priority risks it starving the other threads

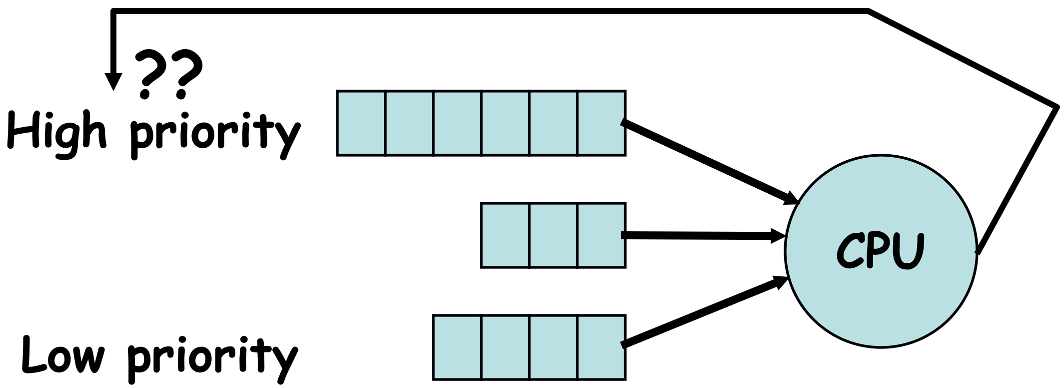
Multi-Level Feedback Scheduling

N priority levels, round-robin scheduling within a level

Should *Quanta* increase as priority decreases?

Are high priority threads more likely to be interactive?

What stops starvation?



Proportional Share Scheduling

The amount of CPU a thread gets is a separate concern from how urgently it must run

Latency and throughput are distinct scheduling concerns

Lottery Scheduling

A kind of proportional share scheduling by chance

Scheduler gives each thread some lottery tickets

To select the next process to run...

- The scheduler randomly selects a lottery number
- The winning process gets to run

Example

Thread A gets 50 tickets

Thread B gets 15 tickets

Thread C gets 35 tickets

There are 100 tickets outstanding

Lottery Scheduling

A kind of proportional share scheduling

Scheduler gives each thread some lottery tickets.

To select the next process to run...

- The scheduler randomly selects a lottery number
- The winning process gets to run

Example

Thread A gets 50 tickets → 50% of CPU

Thread B gets 15 tickets → 15% of CPU

Thread C gets 35 tickets → 35% of CPU

There are 100 tickets outstanding

Proportion-Period Scheduling

No need for randomization

Threads should be able to get throughput and latency guarantees

Necessary in time-sensitive applications