目录 Contents

Uniprocessor Scheduling

1. Type of Processor Scheduling
2. Scheduling Algorithms
3. Real-Time Scheduling

7.1 Type of Processor Scheduling

Aim of Scheduling

- Aim is to assign processes to be executed by the processor in a way that meets system objectives
  - Response time(响应时间)
  - Throughput (吞吐量)
  - Processor efficiency

Types of Scheduling

- **Long-term scheduling(长程调度)**: performed when a new process is created. The decision to add a new process to the set of processes to be executed
- **Medium-term scheduling**: is a part of the swapping. The decision to add to the number of processes that are partially or fully in main memory
- **Short-term scheduling(dispatcher)**: The decision as to which available process will be executed by the processor
- **I/O scheduling**: The decision as to which process's pending I/O request shall be handled by an available I/O device

Scheduling and Process State Transitions

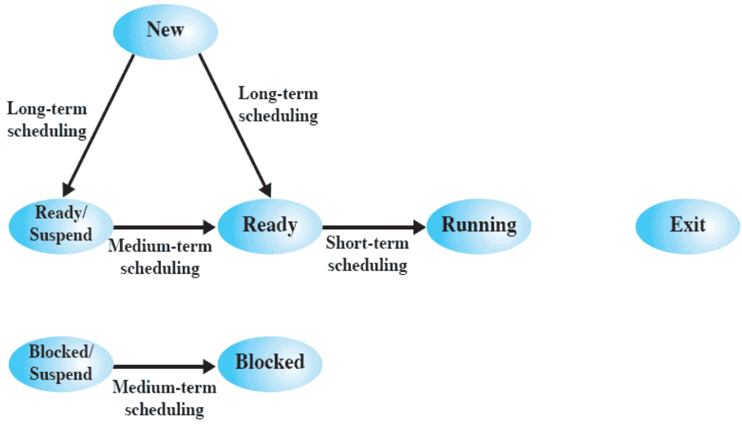


Figure 9.1 Scheduling and Process State Transitions

Queuing Diagram For Scheduling

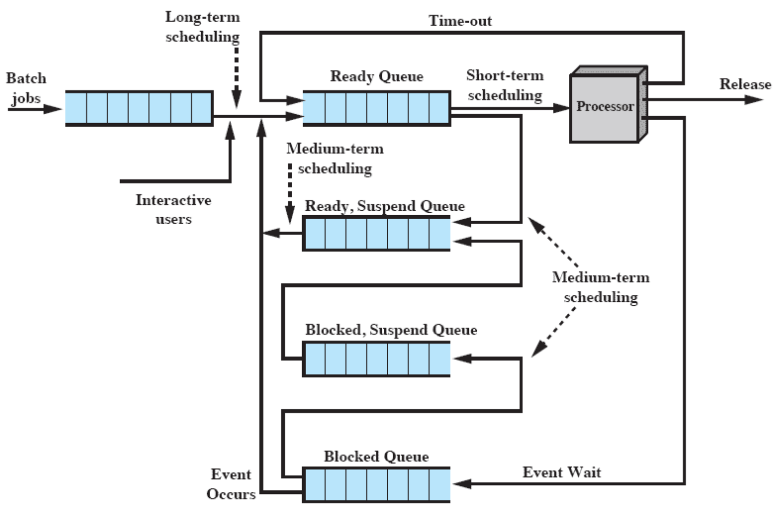
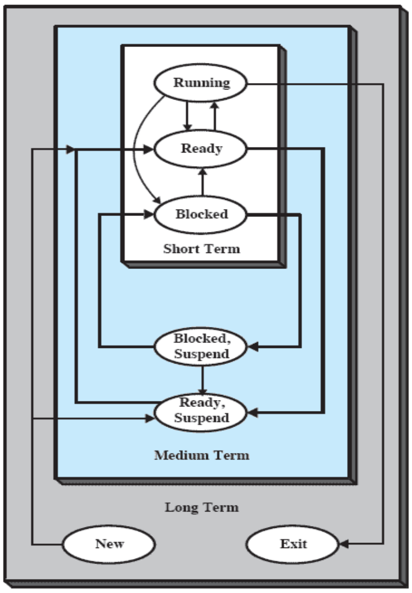


Figure 9.3 Queuing Diagram for Scheduling

Scheduling and Process State Transitions



Long-term Scheduling

- Determines which programs are admitted to the system for processing
- Controls the degree of multiprogramming
  - the more processes that are created, the smaller the percentage of time that each process can be executed
  - may limit the degree to provide satisfactory service to the current set of processes
- Creates processes from the queue of jobs when it can, but must decide:
  - when the operating system can take on one or more additional processes
  - which jobs to accept and turn into processes: first come, first served; priority, expected execution time, I/O requirements

Medium-Term Scheduling

- Part of the swapping function
- Swapping-in
  - decisions are based on the need to manage the degree of multiprogramming
  - considers the memory requirements of the swapped-out processes

Short-Term Scheduling

- Known as the **dispatcher**
- Executes most frequently
- Makes the fine-grained decision of which process to execute next
- Invoked when an event occurs that may lead to the blocking of the current process or that may provide an opportunity to preempt a currently running process in favor of another
  - Clock interrupts
  - I/O interrupts
  - Operating system calls
  - Signals

7.2 Scheduling Algorithms

1 Short-Term Scheduling Criteria

- Main objective is to allocate processor time to optimize certain aspects of system behavior:
  - User-oriented
    - Response Time: Elapsed time between the submission of a request until there is output.
  - System-oriented
    - Effective and efficient utilization of the processor
  - Performance-related
    - Quantitative: Measurable such as response time and throughput

User Oriented, Performance Related

- **Turnaround time(周转时间)**: The interval of time between the submission of a process and its completion. Includes actual execution time plus time spent waiting for resources, including the processor.
  - This is an appropriate measure for a batch job.
- **Response time(响应时间)**: For an interactive process, this is the time from the submission of a request until the response begins to be received.
  - Often a process can begin producing some output to the user while continuing to process the request. Thus, this is a better measure than turnaround time from the user's point of view.
  - The scheduling discipline should attempt to achieve low response time and to maximize the number of interactive users receiving acceptable response time.

User Oriented, Performance Related.

- **Deadlines(最后期限)**: When process completion deadlines can be specified, the scheduling discipline should subordinate other goals to that of maximizing the percentage of deadlines met.

User Oriented, Other

- **Predictability**: A given job should run in about the same amount of time and at about the same cost regardless of the load on the system. A wide variation in response time or turnaround time is distracting to users. It may signal a wide swing in system workloads or the need for system tuning to cure instabilities.

System Oriented, Performance Related

- **Throughput(吞吐量)**: The scheduling policy should attempt to maximize the number of processes completed per unit of time. This is a measure of how much work is being performed.
  - This clearly depends on the average length of a process, but is also influenced by the scheduling policy which may affect utilization
- **Processor utilization**: This is the percentage of time that the processor is busy.
  - For an expensive shared system, this is a significant criterion.
  - In single-user systems and in some other systems, such as real-time systems, this criterion is less important than some of the others.

System Oriented, Other

- **Fairness(公平性)**: In the absence of guidance from the user or other system-supplied guidance, processes should be treated the same, and no process should suffer starvation.
- **Enforcing priorities(强制优先级)**: When processes are assigned priorities, the scheduling policy should favor higher-priority processes.
- **Balancing resources(平衡资源)**: The scheduling policy should keep the resources of the system busy. Processes that will underutilize stressed resources should be favored. This criterion also involves medium-term and long-term scheduling.

7.2 Scheduling Algorithms

2 Priorities

- Scheduler will always choose a process of higher priority over one of lower priority
- Have multiple ready queues to represent each level of priority
- Lower-priority may suffer starvation(饥饿)
  - Allow a process to change its priority based on its age or execution history

Priority Queuing

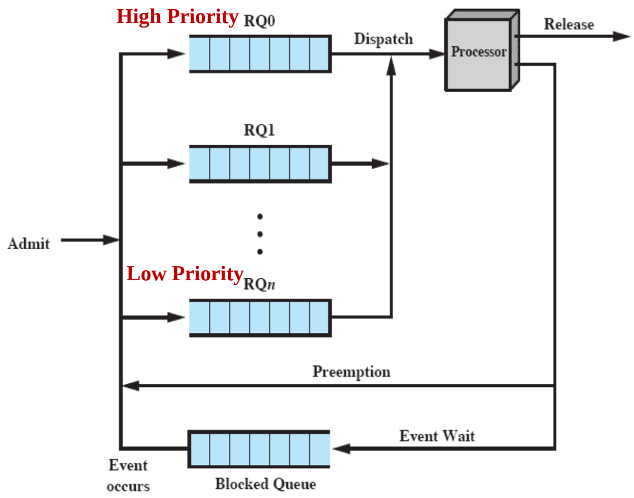


Figure 9.4 Priority Queuing

Decision Mode(决策模式)

- **Nonpreemptive(非抢占)**: Once a process is in the running state, it will continue until it terminates or blocks itself for I/O
- **Preemptive(抢占)**: Currently running process may be interrupted and moved to the Ready state by the operating system
  - Allows for better service since any one process cannot monopolize the processor for very long

7.2 Scheduling Algorithms

3 First-Come-First-Served (FCFS)

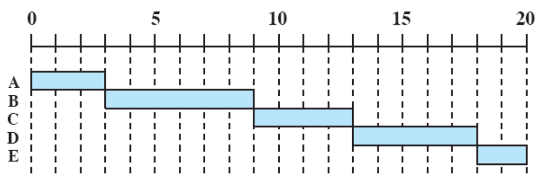
- Each process joins the Ready queue, when the current process ceases to execute, the oldest process in the Ready queue is selected
  - Simplest scheduling policy

> For batch jobs, the **service time** is the total execution time required. Alternatively, if the process require use of the processor and I/O in a repetitive fashion, the service times represent the processor time required in one cycle.

First-Come-First-Served (FCFS).

Process	Arrival Time	Service Time
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2

First-Come-First Served (FCFS)

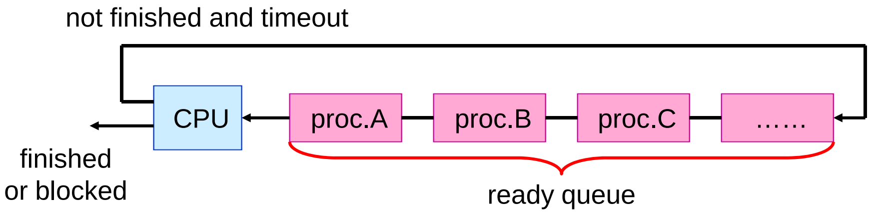


> Nonpreemptive

- A short process may have to wait a very long time before it can execute
- Favors **CPU-bound** processes
  - I/O processes have to wait until CPU-bound process completes

7.2 Scheduling Algorithms

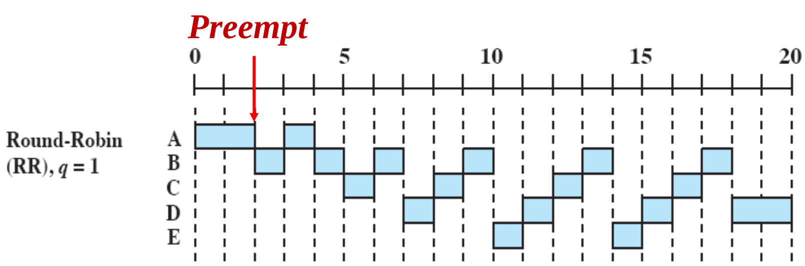
4 Round-Robin(RR)



- Uses **preemption** based on a clock
- An amount of time is determined that allows each process to use the processor for that length of time

Round-Robin(RR)..

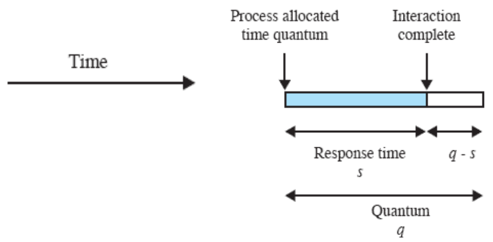
Process	Arrival Time	Service Time
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2



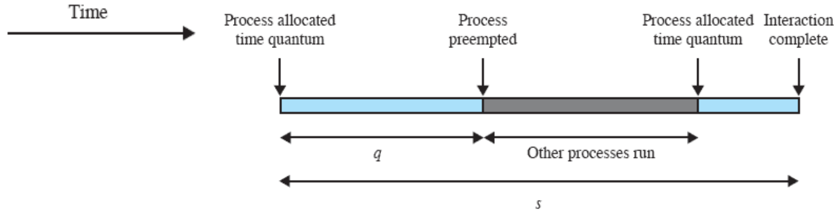
Round-Robin(RR).

- Clock interrupt is generated at periodic intervals
- When an interrupt occurs, the currently running process is placed in the ready queue
  - Next ready job is selected
- the time quantum known as **time slicing**
- If a time quantum is longer than the longest-running process, round robin **degenerates** to FCFS

Effect of Size of Preemption Time Quantum

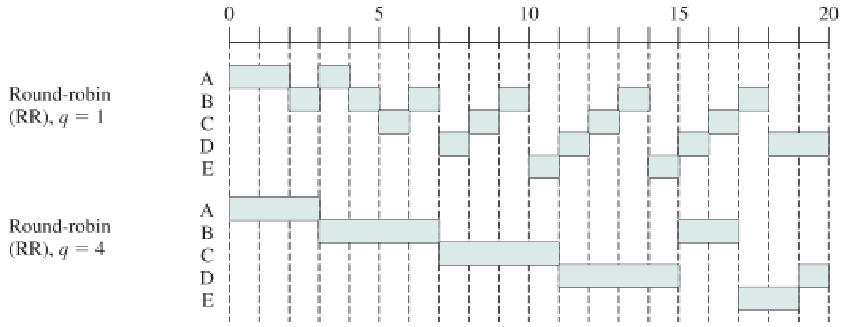


(a) Time quantum greater than typical interaction



(b) Time quantum less than typical interaction

Round-Robin(RR).



Virtual Round-Robin Scheduler(VRR)

- Drawback of RR
  - Unfairness
    - CPU-bound
    - I/O-bound

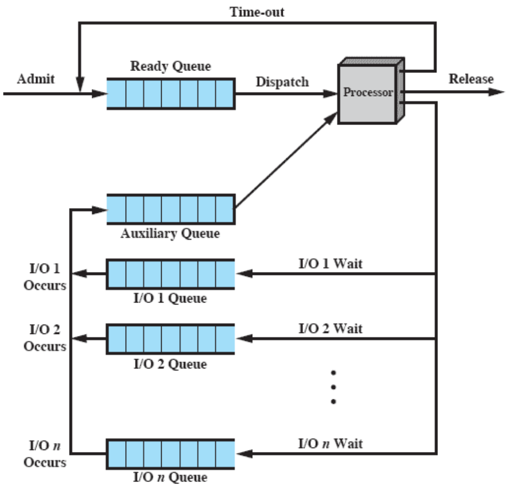
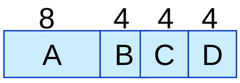


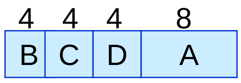
Figure 9.7 Queuing Diagram for Virtual Round-Robin Scheduler

7.2 Scheduling Algorithms

5 Shortest Process Next(SPN)

- Nonpreemptive policy in which the process with the shortest expected processing time is selected next
  - Short process jumps ahead of longer processes
- **Optimal** if and only if all jobs are available at the same time





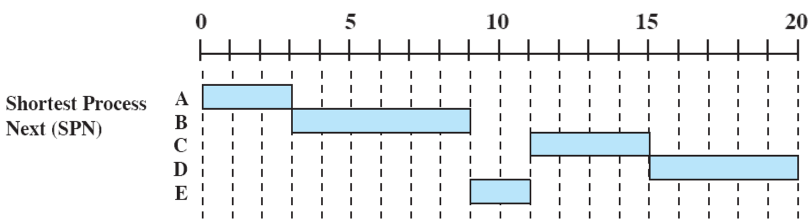
(A) (A+B) (A+B+C) (A+B+C+D) = 4A+3B+2C+D

(B) (B+C) (B+C+D) (B+C+D+A) = 4B+3C+2D+A

average turnaround time=14 average turnaround time=11

Shortest Process Next.

Process	Arrival Time	Service Time
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2



- Nonpreemptive

Shortest Process Next..

- One difficulty is the need to know, or at least estimate, the required processing time of each process
- If estimated time for process not correct, the operating system may abort it
- Possibility of starvation for longer processes

Estimate Running Time

- Running averaging:

$$S_{n+1} = \frac{1}{n}\sum_{i=1}^{n} T_i \qquad S_{n+1} = \frac{1}{n}T_n + \frac{n-1}{n}S_n$$

predicted time ***execution time***

- Exponential averaging:

$$S_{n+1} = \alpha T_n + (1-\alpha)S_n$$

$$S_{n+1} = \alpha T_n + (1-\alpha)\alpha T_{n-1} \ldots + (1-\alpha)^n S_1$$

- Eg: aging(老化). a=1/2

predicted value	execution time	next predicted value
S_1	T_1	$S_2=S_1/2+T_1/2$
S_2	T_2	$S_3=S_2/2+T_2/2$ $=S_1/4+T_1/4+T_2/2$
S_3	T_3	……

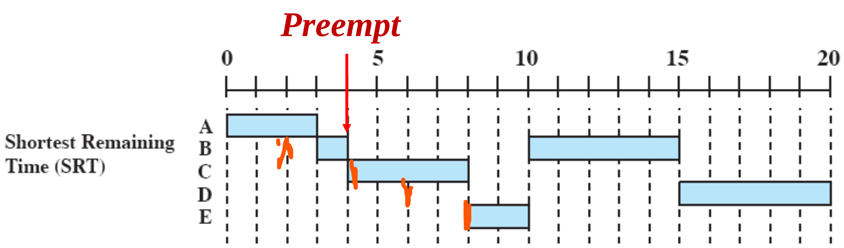
7.2 Scheduling Algorithms

6 Shortest Remaining Time (SRT)

- Scheduler always chooses the process that has the shortest expected remaining processing time
  - Preemptive version of SPN(Shortest Process Next)
- Must estimate processing time
- Should give superior turnaround time performance to SPN because a short job is given immediate preference to a running longer job

Shortest Remaining Time (SRT).

Process	Arrival Time	Service Time
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2



7.2 Scheduling Algorithms

7 Highest Response Ratio Next(HRRN)

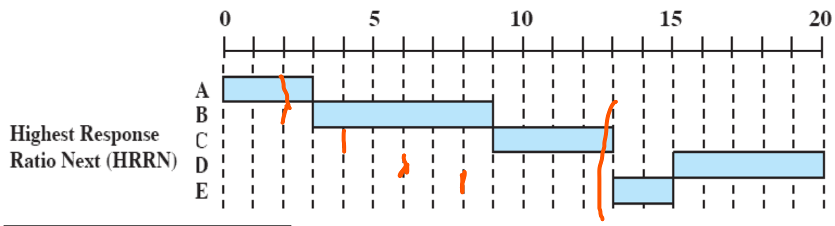
- Chooses next process with the greatest ratio

$$Ratio = \frac{\overset{w}{time\ spent\ waiting} + \overset{s}{expected\ service\ time}}{\underset{s}{expected\ service\ time}}$$

- While shorter jobs are favored, aging without service increases the ratio so that a longer process will eventually get past competing shorter jobs

Highest Response Ratio Next(HRRN)

Process	Arrival Time	Service Time
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2



- Nonpreemptive

7.2 Scheduling Algorithms

8 Feedback Scheduling

- If we don't know remaining time process needs to execute

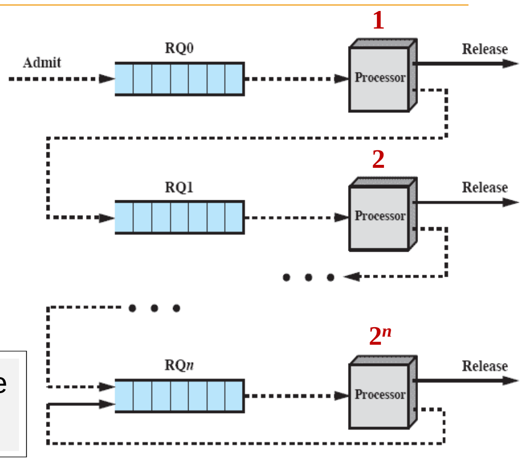
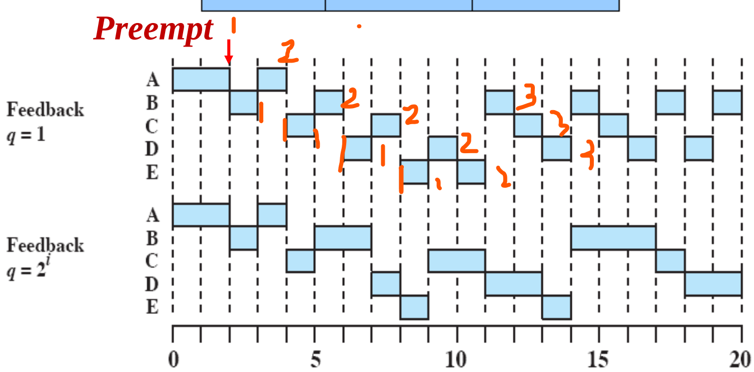


Figure 9.10 Multilevel Feedback Scheduling

> Penalize jobs that have been running longer

Multilevel Feedback

Process	Arrival Time	Service Time
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2



Comparison of Scheduling Policies

	Process	A	B	C	D	E	
	Arrival Time	0	2	4	6	8	
	Service Time (T_s)	3	6	4	5	2	Mean
FCFS	Finish Time	3	9	13	18	20	
	Turnaround Time (T_r)	3	7	9	12	12	8.60
	T_r/T_s	1.00	1.17	2.25	2.40	6.00	2.56
RR $q = 1$	Finish Time	4	18	17	20	15	
	Turnaround Time (T_r)	4	16	13	14	7	10.80
	T_r/T_s	1.33	2.67	3.25	2.80	3.50	2.71
RR $q = 4$	Finish Time	3	17	11	20	19	
	Turnaround Time (T_r)	3	15	7	14	11	10.00
	T_r/T_s	1.00	2.5	1.75	2.80	5.50	2.71
SPN	Finish Time	3	9	15	20	11	
	Turnaround Time (T_r)	3	7	11	14	3	7.60
	T_r/T_s	1.00	1.17	2.75	2.80	1.50	1.84
SRT	Finish Time	3	15	8	20	10	
	Turnaround Time (T_r)	3	13	4	14	2	7.20
	T_r/T_s	1.00	2.17	1.00	2.80	1.00	1.59
HRRN	Finish Time	3	9	13	20	15	
	Turnaround Time (T_r)	3	7	9	14	7	8.00
	T_r/T_s	1.00	1.17	2.25	2.80	3.5	2.14
FB $q = 1$	Finish Time	4	20	16	19	11	
	Turnaround Time (T_r)	4	18	12	13	3	10.00
	T_r/T_s	1.33	3.00	3.00	2.60	1.5	2.29
FB $q = 2^i$	Finish Time	4	17	18	20	14	
	Turnaround Time (T_r)	4	15	14	14	6	10.60
	T_r/T_s	1.33	2.50	3.50	2.80	3.00	2.63

Characteristics of Various Policies

Table 9.3 Characteristics of Various Scheduling Policies

	FCFS	Round robin	SPN	SRT	HRRN	Feedback
Selection function	max[w]	constant	min[s]	min[s − e]	$\max\left(\frac{w+s}{s}\right)$	(see text)
Decision mode	Non-preemptive	Preemptive (at time quantum)	Non-preemptive	Preemptive (at arrival)	Non-preemptive	Preemptive (at time quantum)
Throughput	Not emphasized	May be low if quantum is too small	High	High	High	Not emphasized
Response time	May be high, especially if there is a large variance in process execution times	Provides good response time for short processes	Provides good response time for short processes	Provides good response time	Provides good response time	Not emphasized
Overhead	Minimum	Minimum	Can be high	Can be high	Can be high	Can be high
Effect on processes	Penalizes short processes; penalizes I/O bound processes	Fair treatment	Penalizes long processes	Penalizes long processes	Good balance	May favor I/O bound processes
Starvation	No	No	Possible	Possible	No	Possible

Terminology

- long term scheduling
- medium term scheduling
- short term scheduling; dispatcher
- I/O scheduling
- turnaround time
- response time
- throughput
- predictability
- processor utilization
- nonpreemptive; preemptive
- service time

Terminology

- scheduling algorithm
  - Priority policies
  - first-come, first-served (FCFS)
  - round-robin (RR)
  - shortest process next (SPN)
  - shortest remaining time (SRT)
  - highest response ratio next (HRRN)
  - multi-level feedback queues (MLFQ)