Scheduling Algorithm 1: FCFS

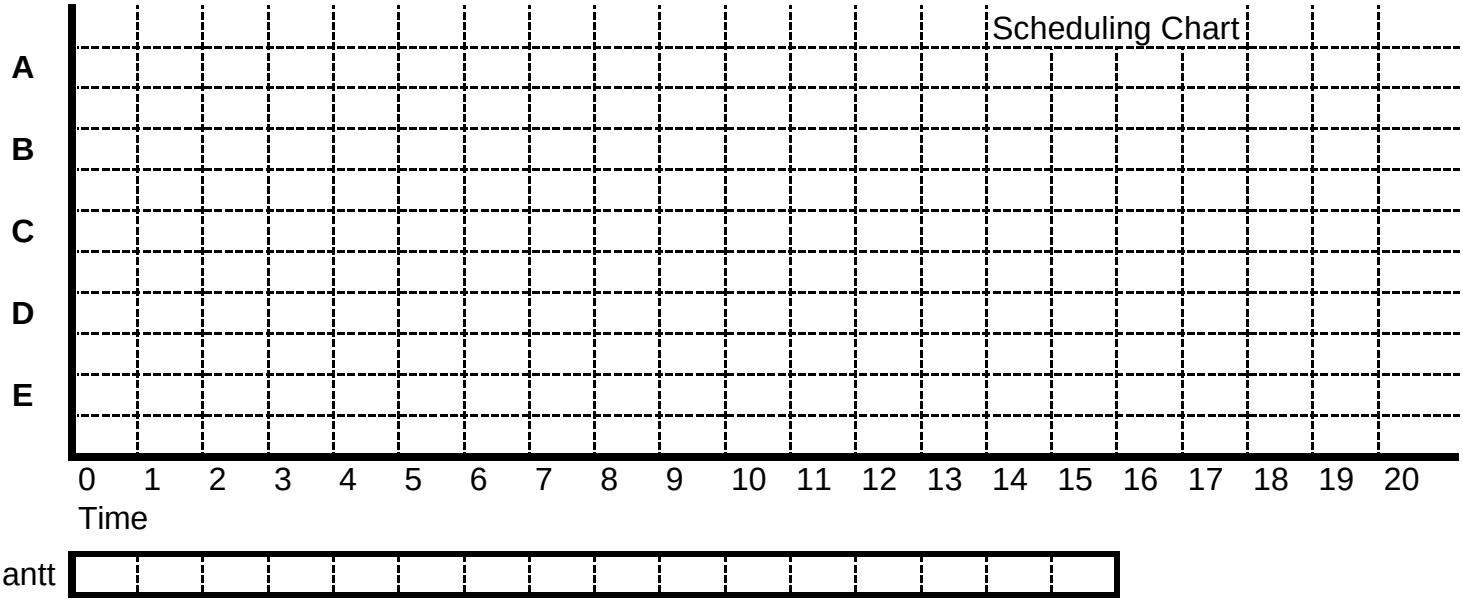
Problem Data

Process	Start	Burst Time t
A	0	7
B	4	4
C	5	1
D	9	1
E	12	3

Legend

M = process missing the processor
A = process A executing
B = process B executing
C = process C executing
D = process D executing
E = process E executing

1. First-Come-First-Serve (FCFS)



Gantt

= the time interval from 0 to 1

A scheduling decision is made at a time point, and then some process runs during the time interval between time points.

t = processing time required (burst time)
T = elapsed time (including missed)
M = T-t = missed (idle) time for process
R = t/T = ratio (response) time
P = T/t =penalty ratio = 1/R

Process	t	T	M	R	P
A					
B					
C					
D					
E					

Scheduling Algorithm 2: Round Robin (RR)

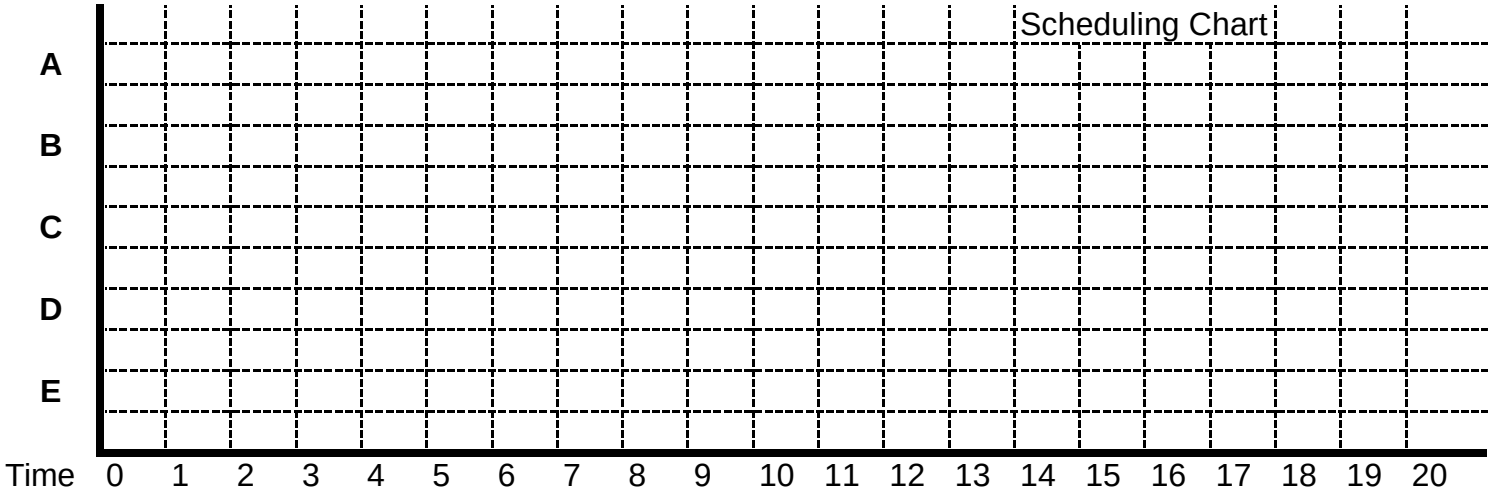
Problem Data

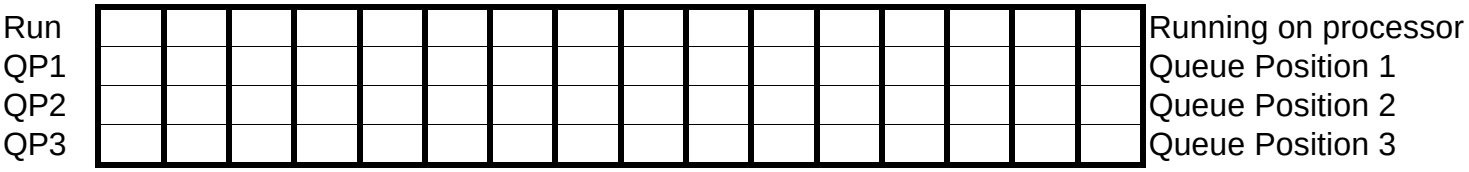
Process	Start	Burst Time t
A	0	7
B	4	4
C	5	1
D	9	1
E	12	3

Legend

M = process missing the processor
A = process A executing
B = process B executing
C = process C executing
D = process D executing
E = process E executing

2. Round Robin (RR)





When adding to the back of the queue, a new process is added first then the preempted process

Process	t	T	M = T - t	R = t / T	P = T / t
A					
B					
C					
D					

E					

Scheduling Algorithm 3: Shortest Job First (SJF)

Problem Data

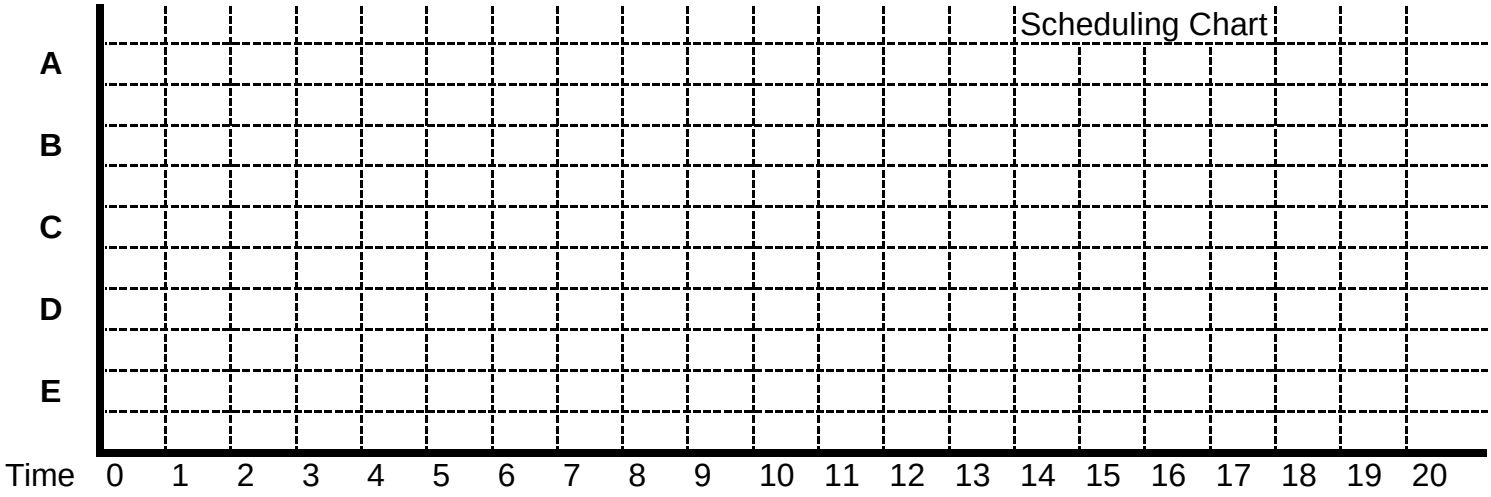
Process	Start	Burst Time t
A	0	7
B	4	4
C	5	1
D	9	1
E	12	3

Nonpreemptive
Text includes SRT as a special case of SJF, but I don't. So SJF is always nonpreemptive.

Easy if you know burst length.
In practice, must estimate.

A form of priority algorithm, where priority is determined by (estimated) burst length.

3. Shortest Job First (SJF)



Run		Running on processor
RL1		Ready List, position 1
RL2		Ready List, position 2
RL3		Ready List, position 3

Gantt	

Process	t	T	M = T - t	R = t / T	P = T / t
A					
B					
C					
D					
E					

If two processes have equal priority, the one waiting longer in the ready list is chosen.

Scheduling Algorithm 4: Shortest Remaining Time (SRT)

Problem Data

Process	Start	Burst Time t
A	0	7
B	4	4
C	5	1
D	9	1
E	12	3

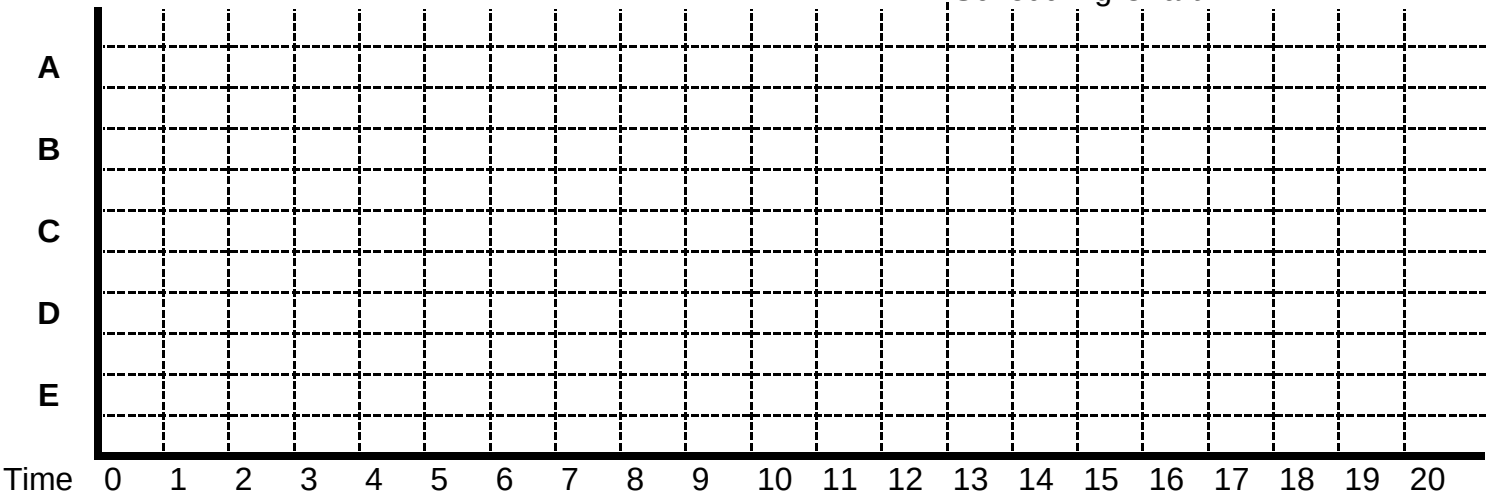
Preemptive, based on time left in burst.

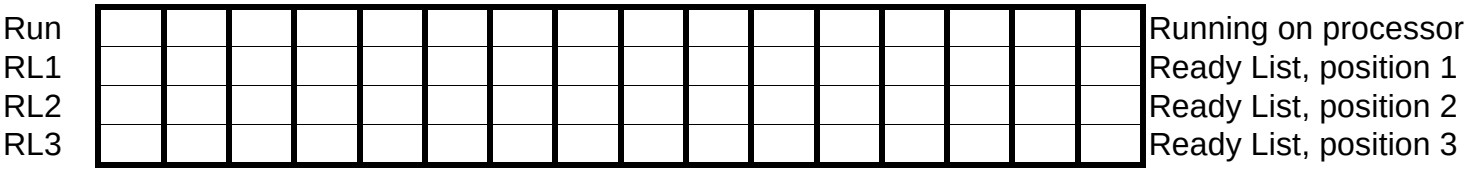
Easy if you know burst length.
In practice, you must estimate.

A form of priority algorithm, where priority is determined by (estimated) remaining burst length.

4. Shortest Remaining Time (SRT)

Scheduling Chart





Process	t	T	M = T - t	R = t / T	P = T / t
A					
B					
C					
D					

E					

If two processes have equal priority, the one waiting longer in the ready list is chosen.

Scheduling Algorithm 5: Nonpreemptive Priority

Problem Data

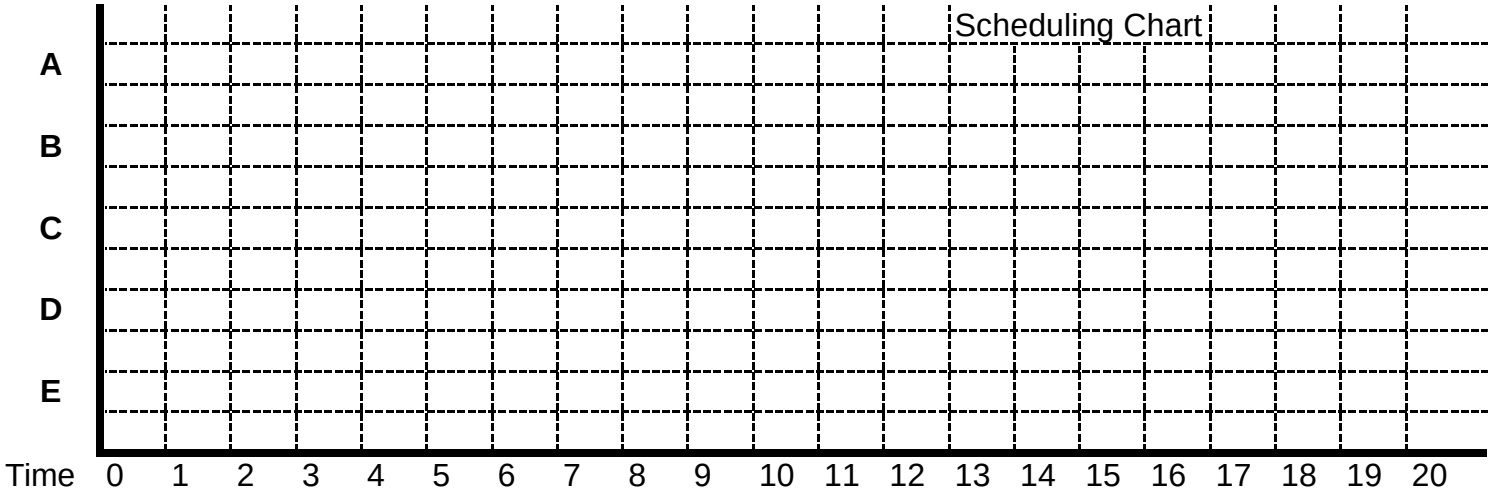
Process	Start	Burst Time t	Priority
A	0	7	4
B	4	4	0
C	5	1	2
D	9	1	1
E	12	3	3

0 is the highest priority

Many ways of choosing priorities:
- if you use estimated run time, you get shortest job first
- here, assign priorities

Ready list ordered by priority first, and arrival time second

5. Nonpreemptive Priority



Run		Running on processor
RL1		Ready List, position 1
RL2		Ready List, position 2
RL3		Ready List, position 3

Gantt	

Process	t	T	M = T - t	R = t / T	P = T / t
A					
B					
C					

D					
E					

If two processes have equal priority, the one waiting longer in the ready list is chosen.

Scheduling Algorithm 6: Preemptive Priority

Problem Data

Process	Start	Burst Time t	Priority
A	0	7	4
B	4	4	0
C	5	1	2
D	9	1	1
E	12	3	3

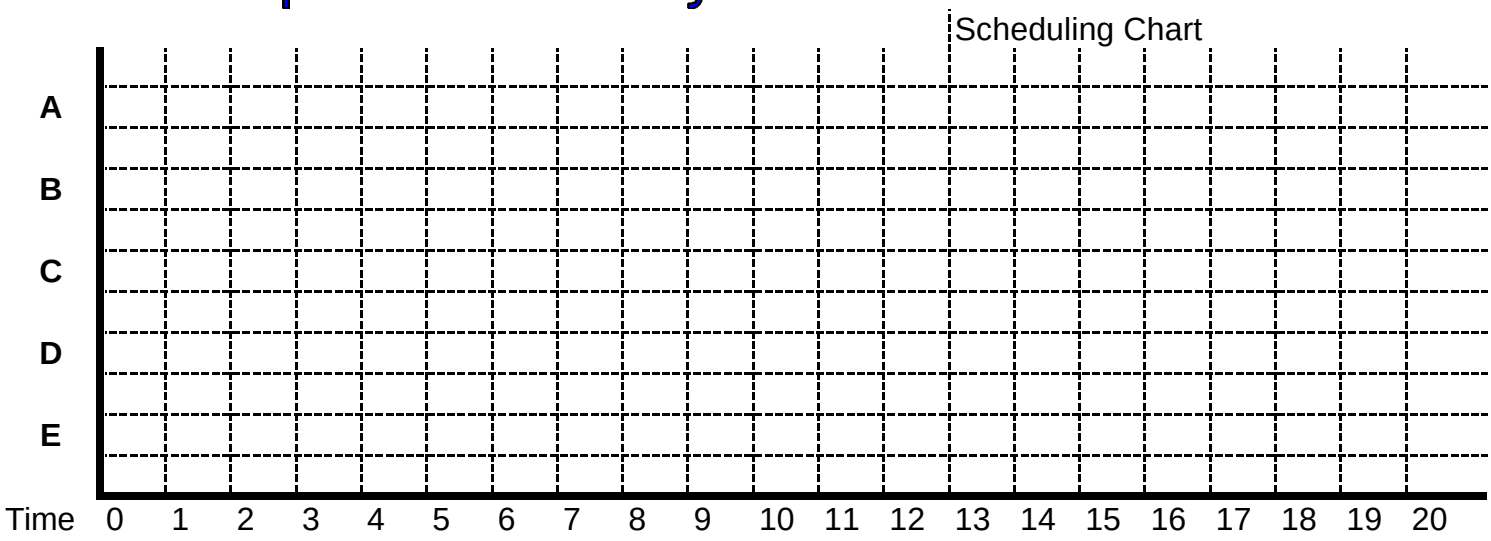
0 is the highest priority

Many ways of choosing priorities:
-- e.g., use estimated run time
-- here, priorities are assigned

Ready list ordered by priority first, and arrival time second

6. Preemptive Priority

Scheduling Chart



Run		Running on processor
RL1		Ready List, position 1
RL2		Ready List, position 2
RL3		Ready List, position 3

Gantt	

Process	t	T	M = T - t	R = t / T	P = T / t
A					
B					

C					
D					
E					

If two processes have equal priority, the one waiting longer in the ready list is chosen.

Scheduling Algorithm 7: Feedback (FB)

Problem Data

Process	Start	Burst Time t
A	0	7
B	4	4
C	5	1
D	9	1
E	12	3

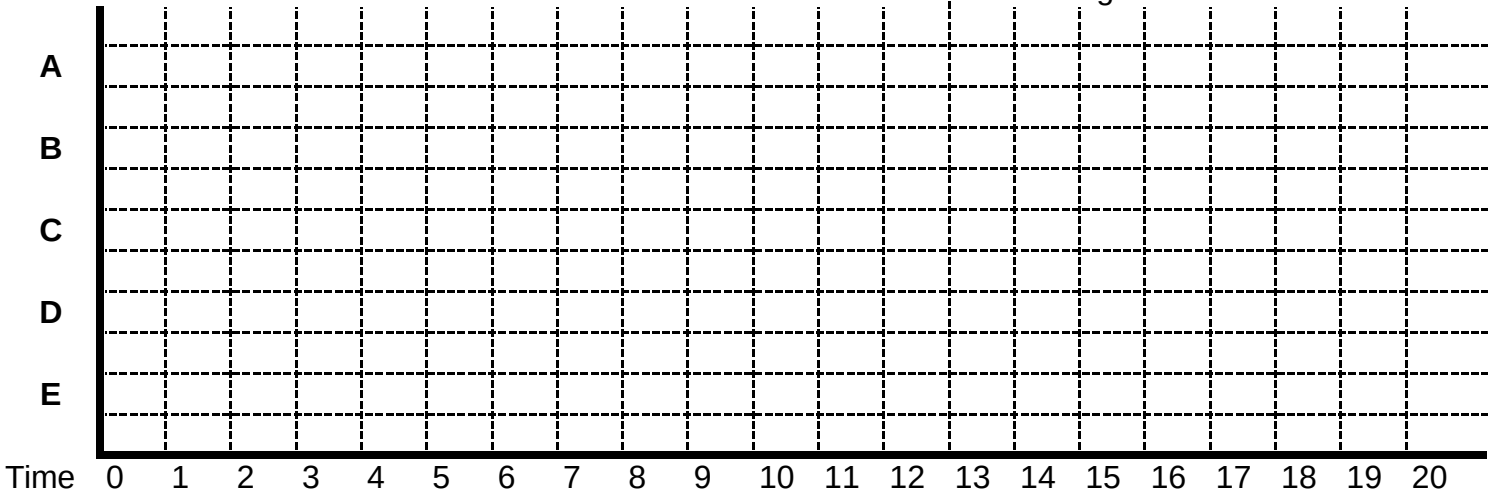
Use burst time to automatically determine priority. The longer the burst, the lower the priority.

Use multiple queues, each sorted by arrival order. Choose first process from highest level queue. Number of queues, Q = 3.

Always preemptive. Here, quantum q = 2.

7. Multilevel Feedback Queues (FB)

Scheduling Chart



Run																	Running on processor
Q0																	Queue 0: pos 0
																	Queue 0: pos 1, etc.
Q1																	Queue 1: pos 0
																	Queue 1: pos 1, etc.
Q2																	Queue 2: pos 0
																	Queue 2: pos 1, etc.

Gantt

Process	t	T	M = T - t	R = t / T	P = T / t

A					
B					
C					
D					
E					