Scheduling Algorithm 1: FCFS
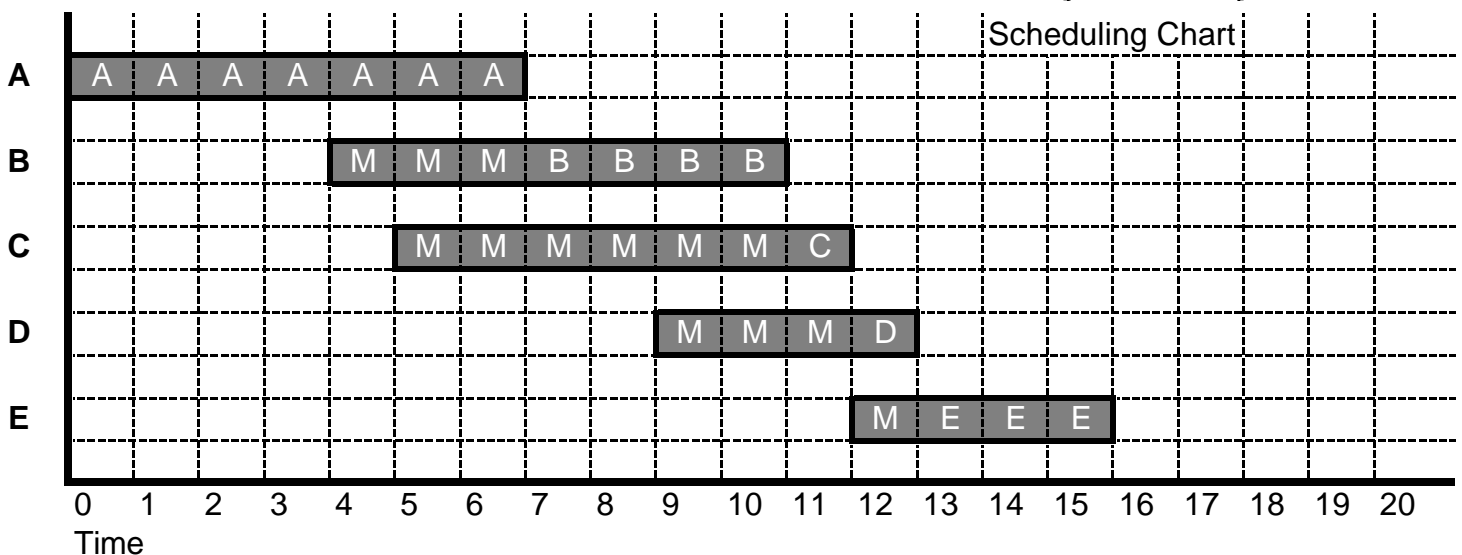
Problem Data

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)

Scheduling Chart

Gantt: A A A A A A A B B B B C D E E E

☐ = 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	7	7	0	1	1
B	4	7	3	4/7	7/4
C	1	7	6	1/7	7
D	1	4	3	1/4	4
E	3	4	1	3/4	4/3

13

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)

Scheduling Chart

| A | A | A | A | A | M | A | M | M | A | M | M | A | | | | | | | | |

B row: B M M B M B M M B

C row: M C

D row: M D

E row: M E E E

Time 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
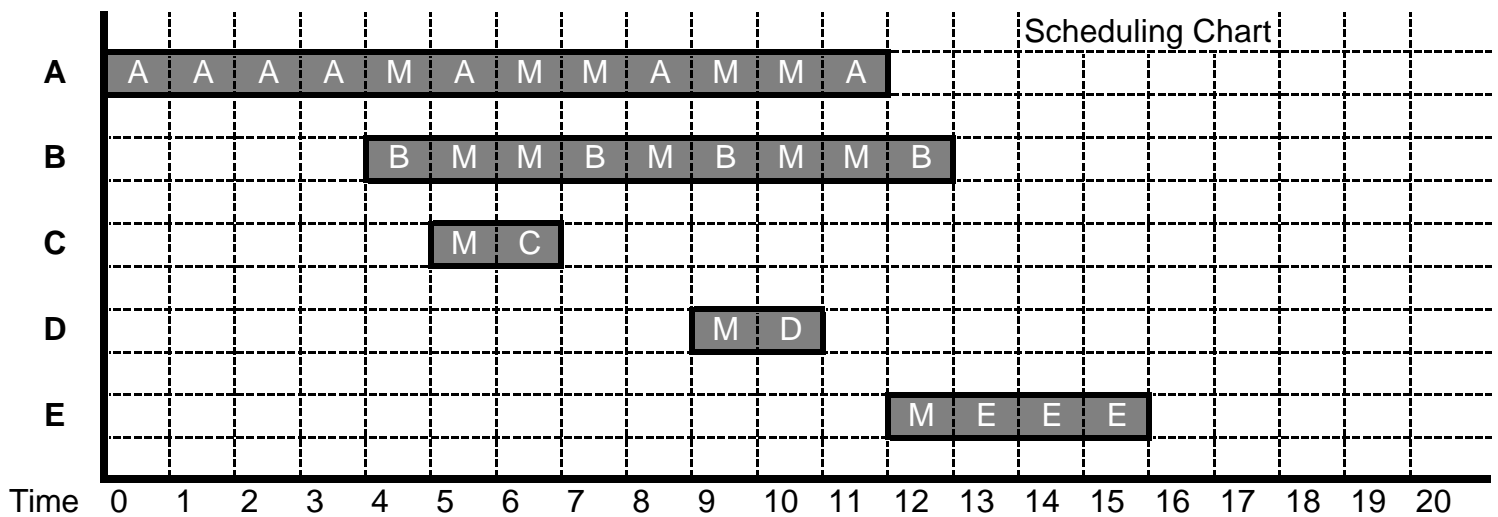
Run	A	A	A	A	B	A	C	B	A	B	D	A	B	E	E	E	Running on processor
QP1					A	C	B	A	B	D	A	B	E				Queue Position 1
QP2					B	A			A	B							Queue Position 2
QP3																	Queue Position 3

Gantt: A A A A B A C B A B D A B E E E

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	7	12	5	7/12	12/7
B	4	9	5	4/9	9/4
C	1	2	1	1/2	2
D	1	2	1	1/2	2
E	3	4	1	3/4	4/3

Scheduling Algorithm 3: Shortest Job First (SJF)

Problem Data

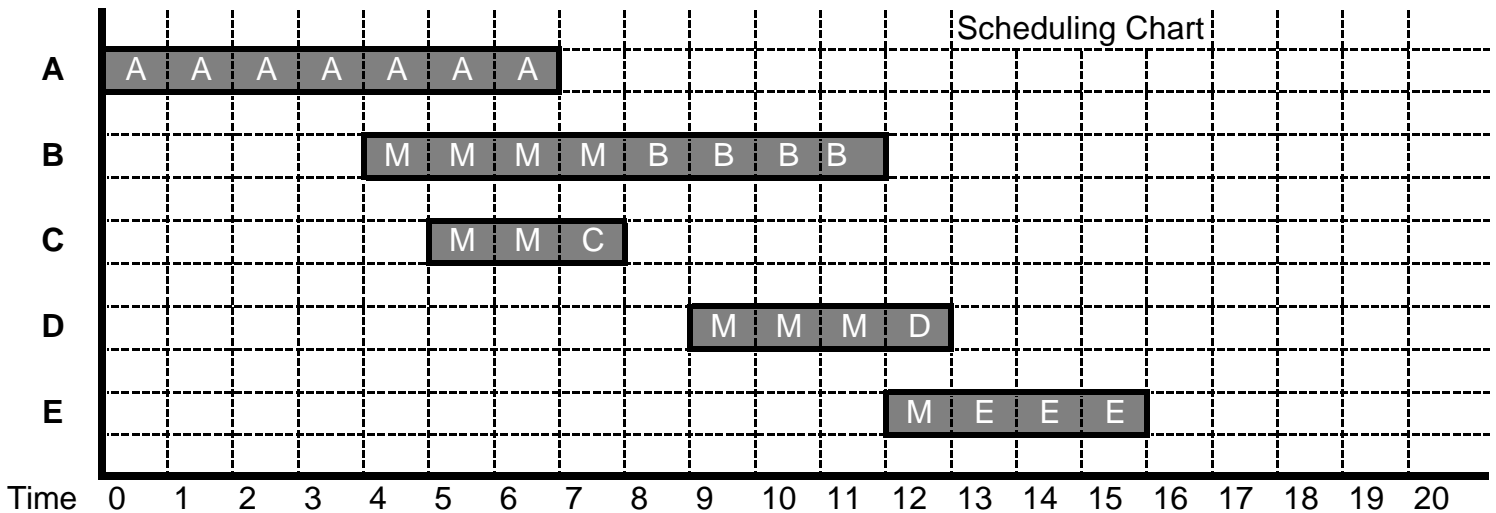| Process	Start	Burst Time t
A	0	7
B	4	4
C	5	1
D	9	1
E	12	3

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)

Scheduling Chart

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
Run	A	A	A	A	A	A	A	C	B	B	B	B	D	E	E	E	Running on processor
RL1					B	C	C	B		D	D	D	E				Ready List, position 1
RL2						B	B										Ready List, position 2
RL3																	Ready List, position 3

Gantt A A A A A A A C B B B B D E E E

Process	t	T	M = T - t	R = t / T	P = T / t
A	7	7	0	1	1
B	4	8	4	4/8	2
C	1	3	2	1/3	3
D	1	4	3	1/4	4
E	3	4	1	3/4	4/3

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

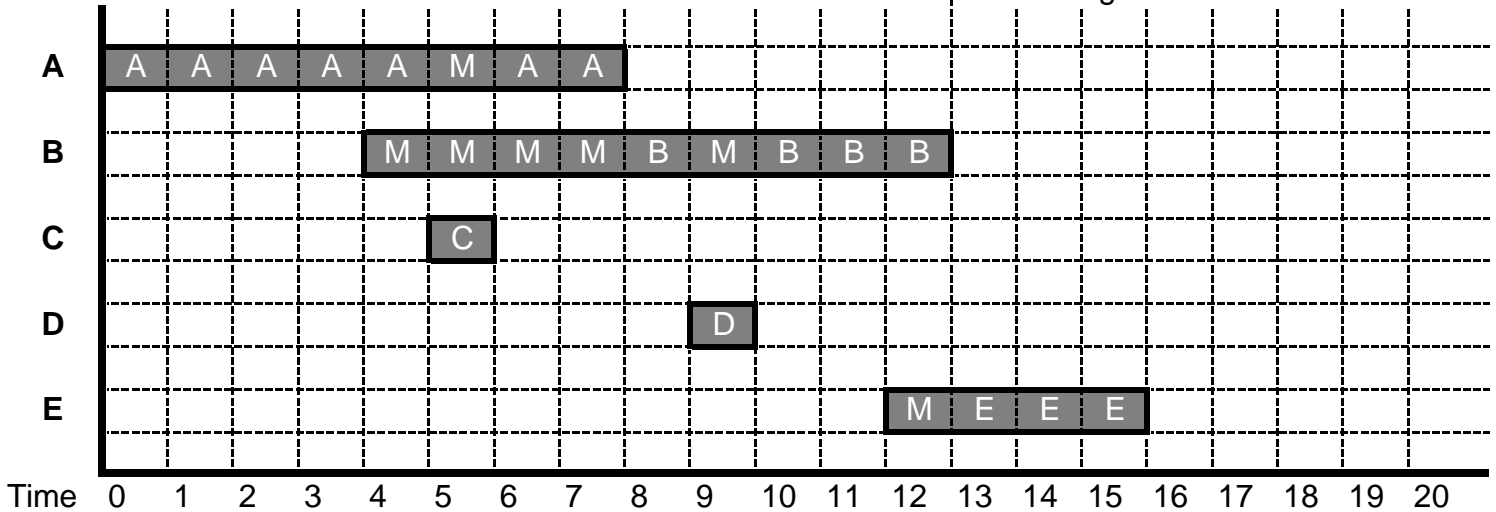# Scheduling Algorithm 4: Shortest Remaining Time (SRT)

Problem Data

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.

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

4. Shortest Remaining Time (SRT)

Scheduling Chart

Run	A	A	A	A	A	C	A	A	B	D	B	B	B	E	E	E	Running on processor
RL1						B	A	B	B		B			E			Ready List, position 1
RL2						B											Ready List, position 2
RL3																	Ready List, position 3

Gantt | A | A | A | A | A | C | A | A | B | D | B | B | B | E | E | E |

Process	t	T	M = T - t	R = t / T	P = T / t
A	7	8	1	7 / 8	8 / 7
B	4	9	5	4 / 9	9 / 4
C	1	1	0	1	1
D	1	1	0	1	1
E	3	4	1	3 / 4	4 / 3

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

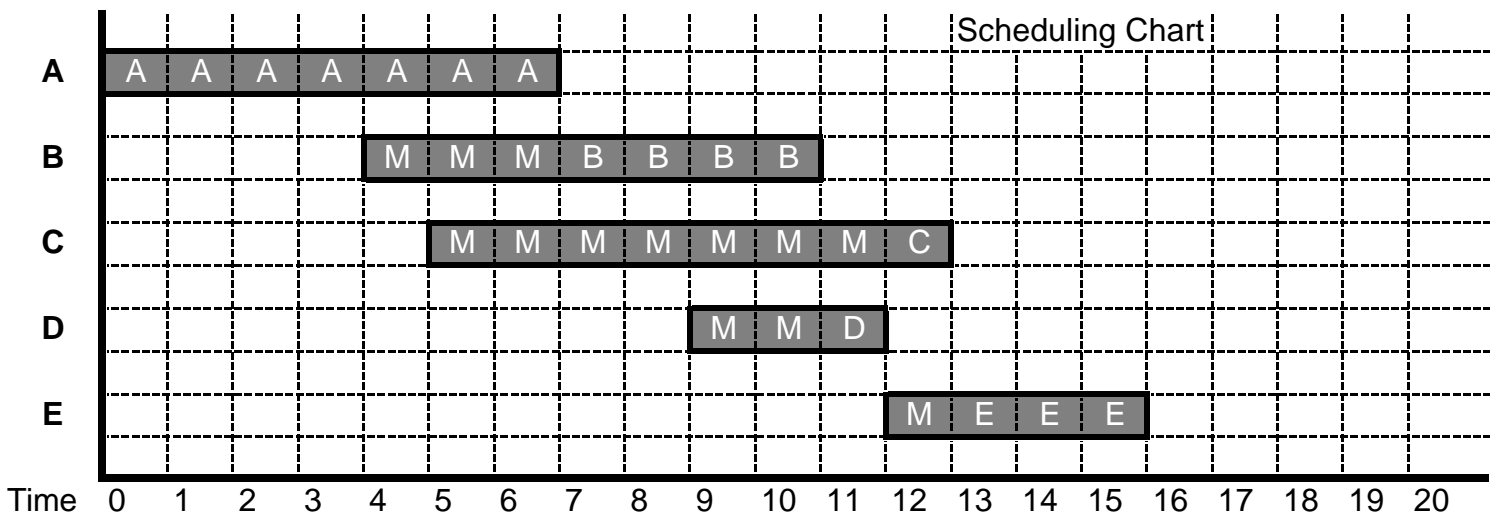# Scheduling Algorithm 5: Nonpreemptive Priority

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

Scheduling Chart

A	A	A	A	A	A	A	A													
B					M	M	M	B	B	B	B									
C					M	M	M	M	M	M	M	C								
D									M	M	D									
E													M	E	E	E				

Time 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

Run	A	A	A	A	A	A	A	B	B	B	B	D	C	E	E	E	Running on processor
RL1					B	B	B	C	C	D	D	C	E				Ready List, position 1
RL2						C	C			C	C						Ready List, position 2
RL3																	Ready List, position 3

Gantt A A A A A A A B B B B D C E E E

Process	t	T	M = T - t	R = t / T	P = T / t
A	7	7	0	1	1
B	4	7	3	4 / 7	7 / 4
C	1	8	7	1 / 8	8
D	1	3	2	1 / 3	3

| | E | 3 | 4 | 1 | 3 / 4 | 4 / 3 |

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

Scheduling Algorithm 6: Preemptive Priority

Problem Data

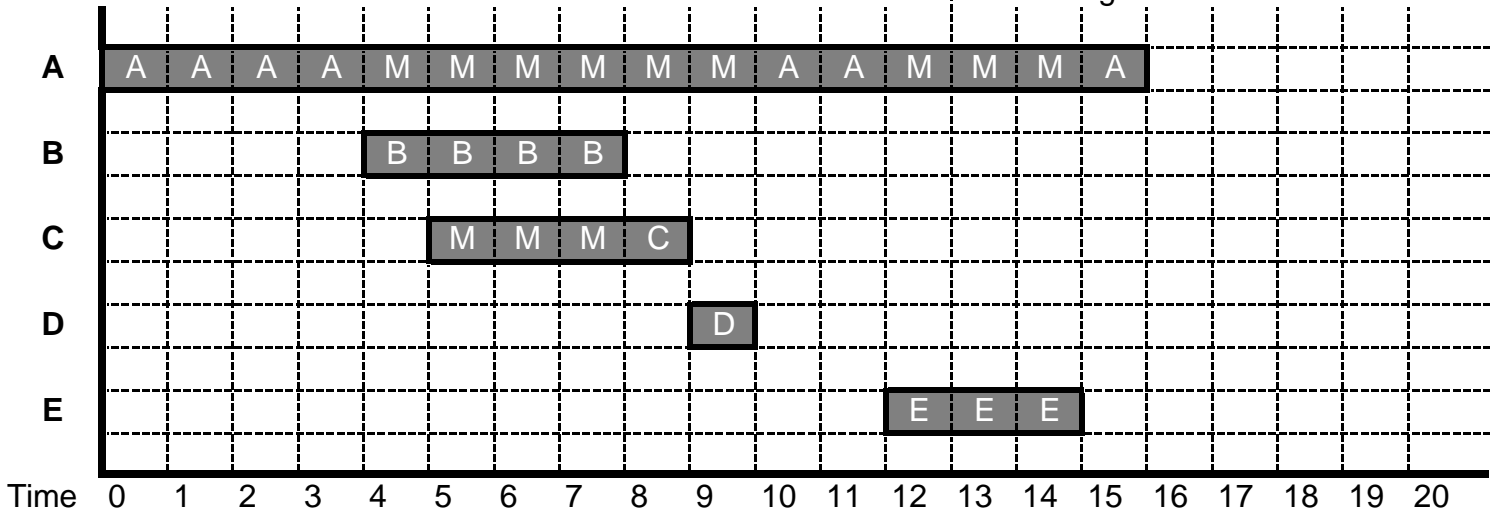| Process	Start	Burst Time t	Priority
A	0	7	4
B	4	4	0
C	5	1	2
D	9	1	1
E	12	3	3

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	A	A	A	A	B	B	B	B	C	D	A	A	E	E	E	A	Running on processor
RL1					A	C	C	C	A	A			A	A	A		Ready List, position 1
RL2						A	A	A									Ready List, position 2
RL3																	Ready List, position 3

Gantt A A A A B B B B C D A A E E E A

Process	t	T	M = T - t	R = t / T	P = T / t
A	7	16	9	7 / 16	16 / 7
B	4	4	0	1	1
C	1	4	3	1 / 4	4

D	1	1	0	1	1
E	3	3	0	1	1

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

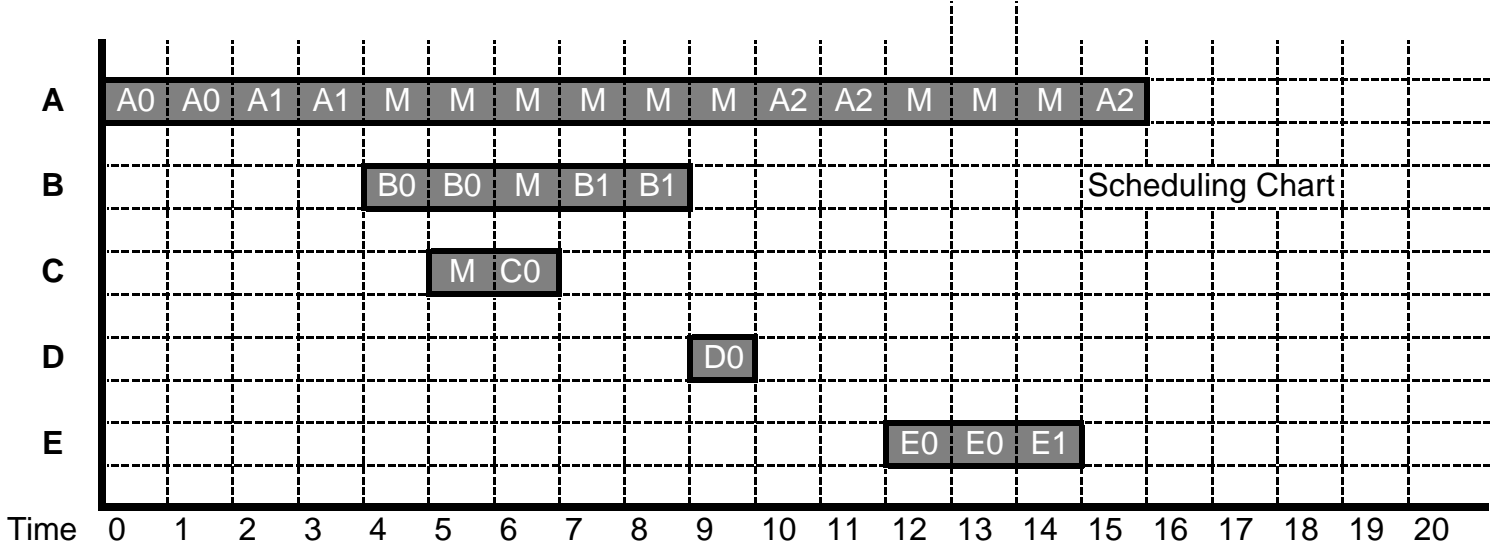# Scheduling Algorithm 7: Feedback (FB)

Problem Data

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

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

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

| A | A0 | A0 | A1 | A1 | M | M | M | M | M | M | A2 | A2 | M | M | M | A2 | |

| B | | | | | B0 | B0 | M | B1 | B1 | | | | | | | | |

| C | | | | | M | C0 | | | | | | | | | | | |

| D | | | | | | | | | | D0 | | | | | | | |

| E | | | | | | | | | | | | E0 | E0 | E1 | | | |

| Time | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |

Run	A	A	A	A	B	B	C	B	B	D	A	A	E	E	E	A	Running on processor
Q0						C											Queue 0: pos 0
																	Queue 0: pos 1, etc.
Q1						B											Queue 1: pos 0
																	Queue 1: pos 1, etc.
Q2					A	A	A	A	A	A			A	A	A		Queue 2: pos 0
																	Queue 2: pos 1, etc.

| Gantt | A | A | A | A | B | B | C | B | B | D | A | A | E | E | E | A | |

Process	t	T	M = T - t	R = t / T	P = T / t
A	7	16	9	7 / 16	16 / 7

B	4	5	1	4 / 5	5 / 4
C	1	2	1	1 / 2	2
D	1	1	0	1	1
E	3	3	0	1	1