Scheduling Exercises: Rate Monotonic Scheduling (RMS) Algorithm and Earliest Deadline First (EDF) Algorithm

Multimedia Technology

Tutorial 1, section 1

RMS - EDF

- Assume we have a real-time system with 4 processes A, B, C, and D, with execution periods of 15, 20, 30, and 60 ms, and execution times of 5, 5, 10, and 5 ms, respectively. Show how these processes would be scheduled (or would fail to be scheduled) using the RMS and EDF scheduling algorithms. Assume that all processes are ready for the first time at time 0.
- In RMS, priority in each slot is given to the process that is ready and has the highest frequency of occurrence (or the shortest period).

Rate Monotonic Scheduling (RMS) Algorithm

Processes	Execution time	Period
A	5 ms	15 ms
B	5 ms	20 ms
C	10 ms	30 ms
D	5 ms	60 ms

Execution time refers to the duration a process takes to complete its execution.

Period refers to the interval after which the process will reappear, ready to be processed by the system again. Think of it as a recurring deadline.

Assume we have four processes, as shown in the table, all of which are ready at time 0 ms for the first time.

Note that after their deadlines pass, they become ready again for execution.

In the RMS algorithm, the process with the shortest period, the one that appears most frequently, is assigned the highest priority.

Priorities: Calculated based on the period of each process and sorted in ascending order of their periods: A > B > C > D

A	▨			▨			▨			▨			▨	
B	▨				▨				▨				▨	
C	▨	▨					▨	▨					▨	▨
D	▨												▨	
	0	5	10	15	20	25	30	35	40	45	50	55	60	

RMS

A													

At 0 ms, process A has the highest priority, so it will be executed first. Meanwhile, processes B, C, and D will remain pending and will be considered for execution in the next interval, starting at 5 ms.

A > B > C > D

A	▨			▨			▨			▨			▨	
B	▨				▨				▨				▨	
C	▨	▨					▨	▨					▨	▨
D	▨												▨	
	0	5	10	15	20	25	30	35	40	45	50	55	60	

RMS

A	B												

At 5 ms, process B has the highest priority, so it will be executed first during the current (5-10 ms) interval.

A > B > C > D

A	▨			▨			▨			▨			▨	
B	▨				▨				▨				▨	
C	▨	▨					▨	▨					▨	▨
D	▨												▨	
	0	5	10	15	20	25	30	35	40	45	50	55	60	

RMS

A	B	C											

A > B > C > D

At 10 ms, process C has the highest priority, so it will begin execution during this interval.
Remember that process C has an execution time of 10 ms, so 5 ms of its execution will still be pending beyond the current interval.

A	▨			▨			▨			▨			▨	
B	▨				▨				▨				▨	
C	▨	▨					▨	▨					▨	▨
D	▨												▨	
	0	5	10	15	20	25	30	35	40	45	50	55	60	

RMS

A	B	C	A										

At 15 ms, process A reappears in the system.
Process A has the highest priority, so it will begin execution during this interval.

A > B > C > D

A	▨			▨			▨			▨			▨	
B	▨				▨				▨				▨	
C	▨	▨					▨	▨					▨	▨
D	▨												▨	
	0	5	10	15	20	25	30	35	40	45	50	55	60	

RMS

A	B	C	A	B									

At 20 ms, process B reappears in the system.
Process B has the highest priority, so it will begin execution during this interval.

A > B > C > D

A	▨			▨			▨			▨			▨	
B	▨				▨				▨				▨	
C	▨	▨					▨	▨					▨	▨
D	▨												▨	
	0	5	10	15	20	25	30	35	40	45	50	55	60	

RMS

A	B	C	A	B	C								

At 25 ms, process C has the highest priority, so it will begin execution during this interval.

A > B > C > D

A	▨			▨			▨			▨			▨	
B	▨				▨				▨				▨	
C	▨	▨					▨	▨					▨	▨
D	▨												▨	
	0	5	10	15	20	25	30	35	40	45	50	55	60	

RMS

A	B	C	A	B	C	A							

At 30 ms, process A and process C reappear in the system.
Process A has the highest priority, so it will begin execution during this interval.

A > B > C > D

A	▨			▨			▨			▨			▨	
B	▨				▨				▨				▨	
C	▨	▨					▨	▨					▨	▨
D	▨												▨	
	0	5	10	15	20	25	30	35	40	45	50	55	60	

RMS

A	B	C	A	B	C	A	C						

At 35 ms, process C has the highest priority, so it will begin execution during this interval.
Note that process C has an execution time of 10 ms, so 5 ms of its execution will still be pending beyond the current interval.

A > B > C > D

A	▨			▨			▨			▨			▨	
B	▨				▨				▨				▨	
C	▨	▨					▨	▨					▨	▨
D	▨												▨	
	0	5	10	15	20	25	30	35	40	45	50	55	60	

RMS

A	B	C	A	B	C	A	C	B					

At 40 ms, process B reappears in the system.
Process B has the highest priority, so it will begin execution during this interval.

A > B > C > D

A	▨			▨			▨			▨			▨	
B	▨				▨				▨				▨	
C	▨	▨					▨	▨					▨	▨
D	▨												▨	
	0	5	10	15	20	25	30	35	40	45	50	55	60	

RMS	A	B	C	A	B	C	A	C	B	A				

At 45 ms, process A reappears in the system.
Process A has the highest priority, so it will begin execution during this interval.

A > B > C > D

A	▨			▨			▨			▨			▨	
B	▨				▨				▨				▨	
C	▨	▨					▨	▨					▨	▨
D	▨												▨	
	0	5	10	15	20	25	30	35	40	45	50	55	60	

RMS

A	B	C	A	B	C	A	C	B	A	C			

At 50 ms, process C has the highest priority, so it will begin execution during this interval.

A > B > C > D

A	▨			▨			▨			▨			▨	
B	▨				▨				▨				▨	
C	▨	▨					▨	▨					▨	▨
D	▨												▨	
	0	5	10	15	20	25	30	35	40	45	50	55	60	

RMS

A	B	C	A	B	C	A	C	B	A	C	D		

At 55 ms, process D has the highest priority and is the only process waiting for execution, so it will begin execution during this interval.

Note that at 60 ms, all processes reappear, ready for execution once again.

A > B > C > D

Earliest Deadline First (EDF) Algorithm

Processes	Execution time	Period
A	5 ms	15 ms
B	5 ms	20 ms
C	10 ms	30 ms
D	5 ms	60 ms

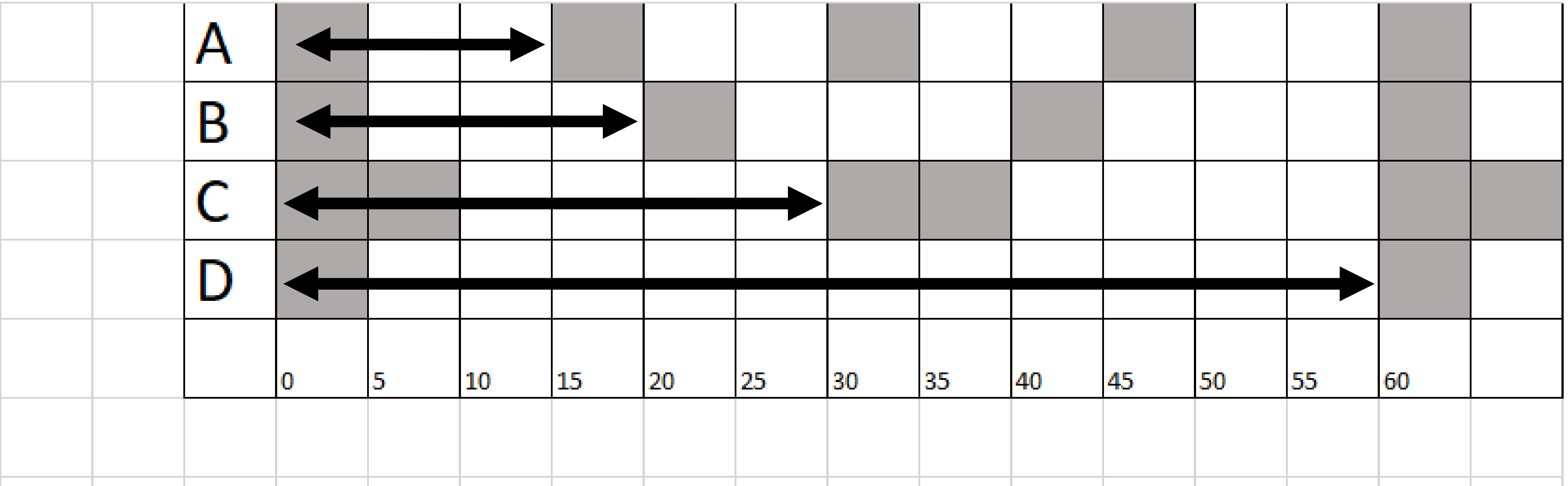
Execution time refers to the duration a process takes to complete its execution.

Period refers to the interval after which the process will reappear, ready to be processed by the system again. Think of it as a recurring deadline.

Assume we have four processes, as shown in the table, all of which are ready at time 0 ms for the first time.

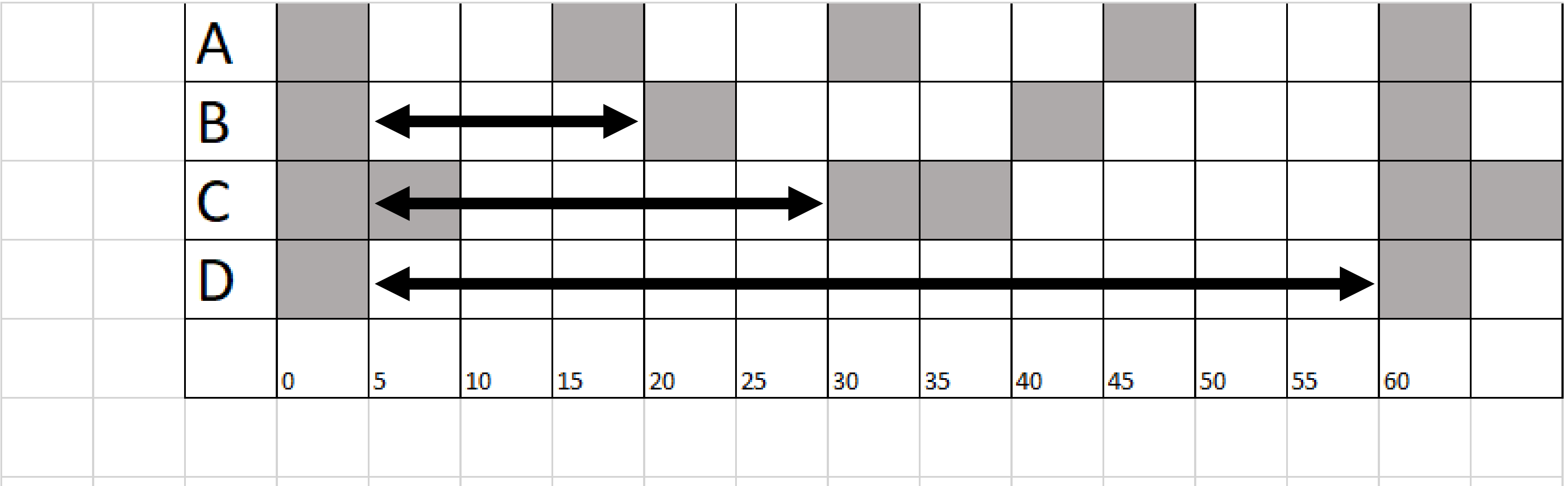
Note that after their deadlines pass, they become ready again for execution.

In the EDF algorithm, the process with the earliest deadline is assigned the highest priority.



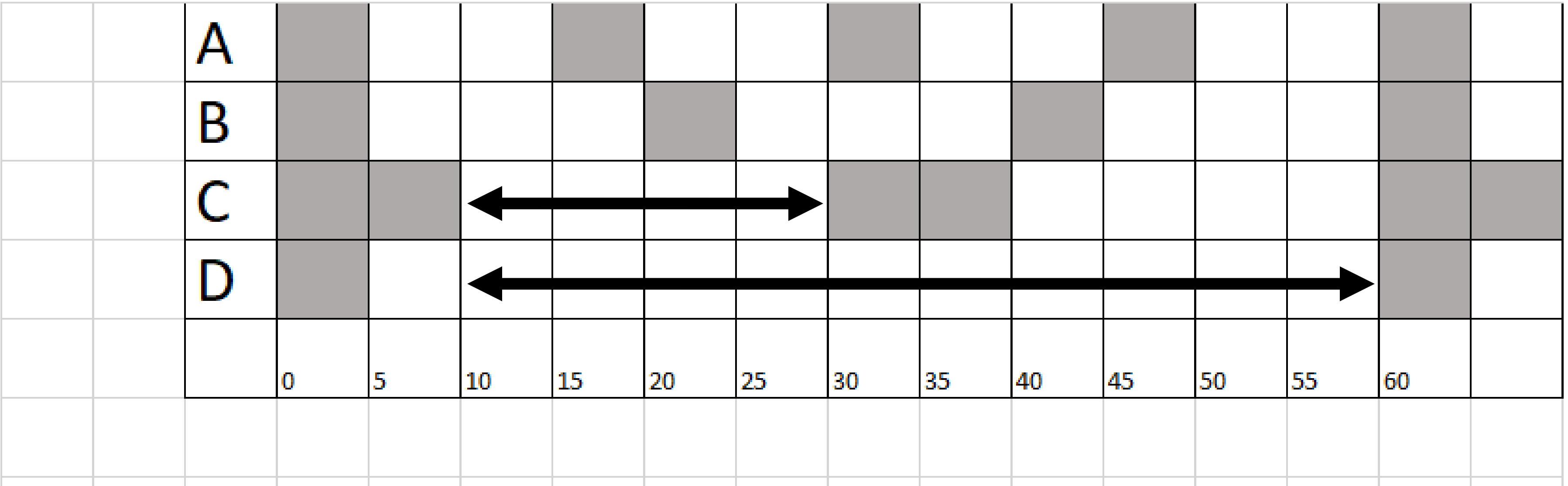
A													

At 0 ms, process A has the highest priority, as it has the earliest deadline, so it will be executed first.



A	B												

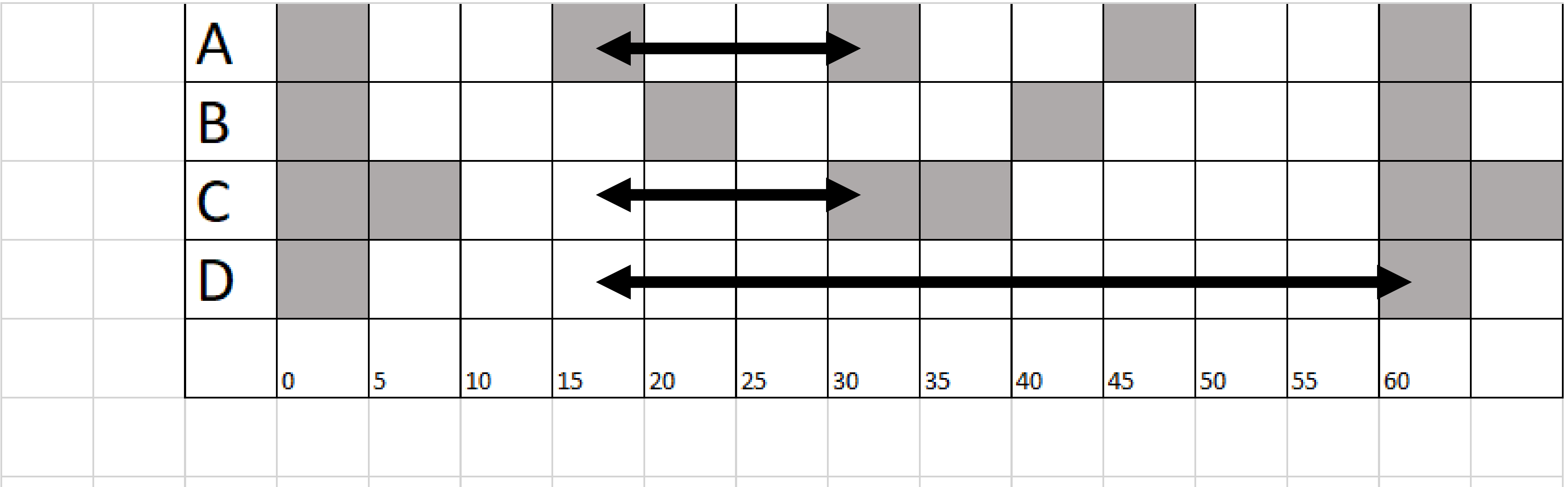
At 5 ms, we observe that process B has the shortest deadline, and thus is assigned the highest priority, so it will be executed first during the current (5-10 ms) interval.



A	B	C											

At 10 ms, process C has the shortest deadline and is assigned the highest priority, so it will begin execution during this interval.

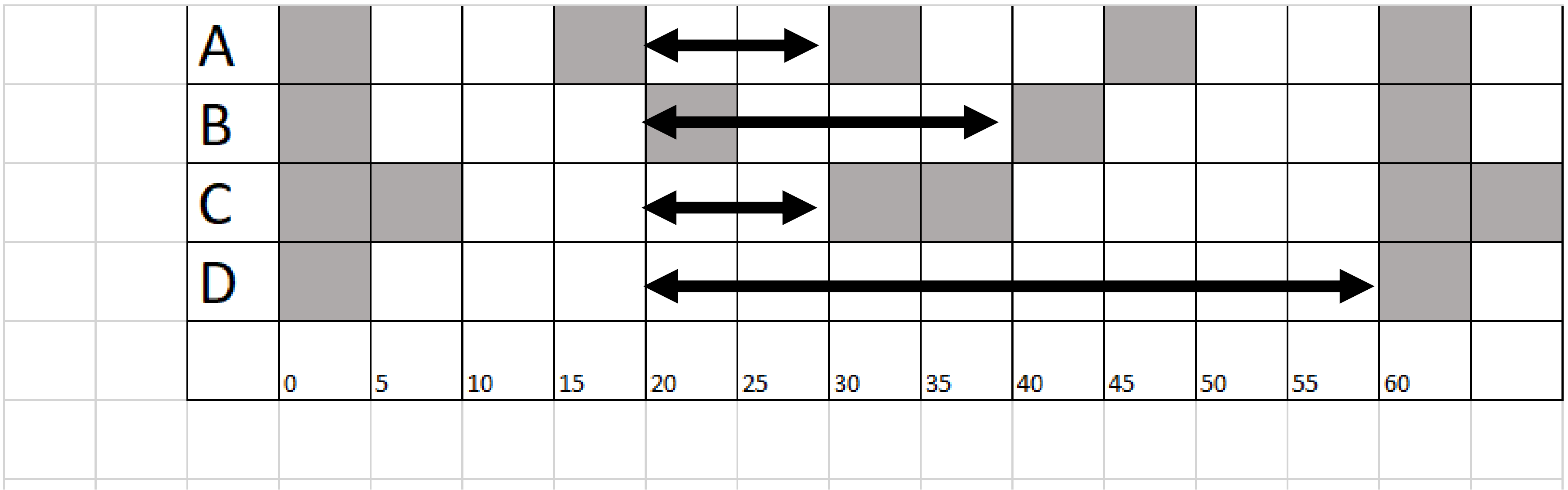
Remember that process C has an execution time of 10 ms, so 5 ms of its execution will still be pending beyond the current interval.



			A										
A	B	C											
			C										

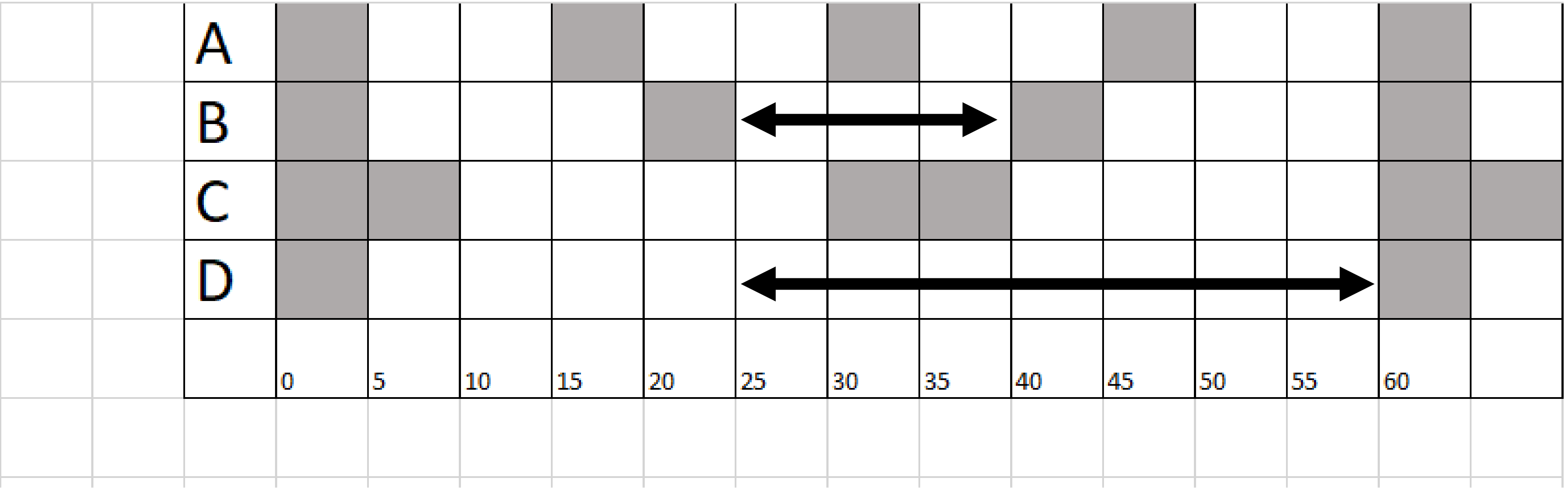
At 15 ms, we have three processes to consider for execution. We can see that processes A and the remaining portion of process C have the same deadline, which is earlier than the deadline of process D. Therefore, both processes A and C are assigned the highest (equal) priority.

At this point, selecting either process A or the remaining portion of process C for execution is interchangeable, as the other will execute in the next interval. Thus, either choice is acceptable.



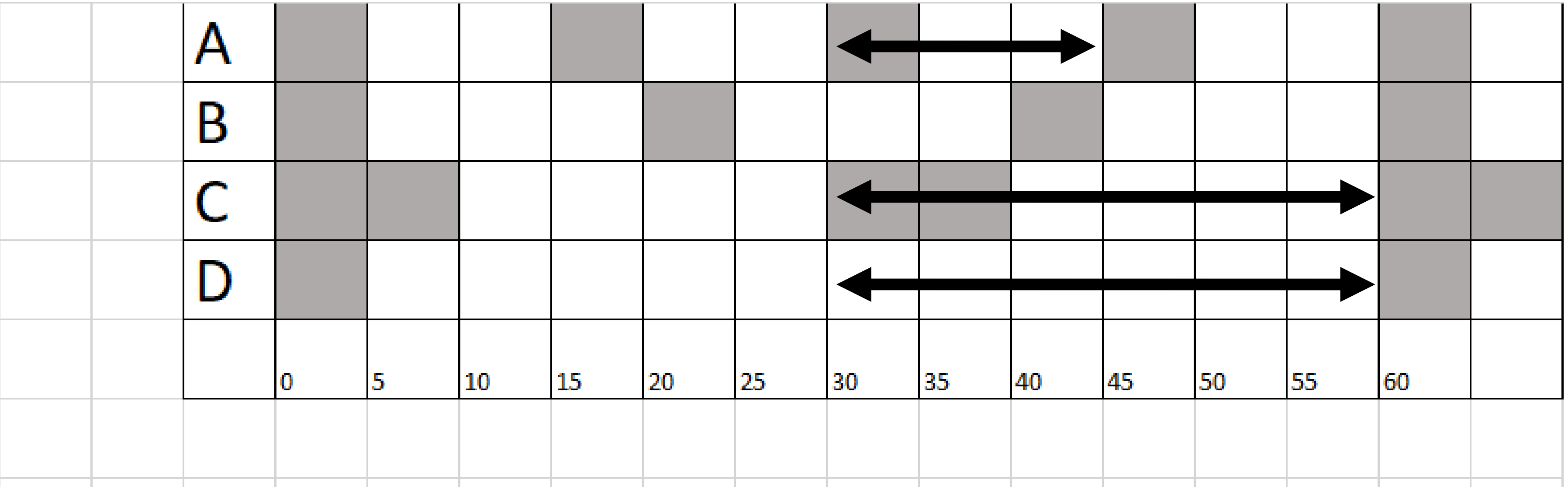
			A	C									
A	B	C											
			C	A									

At 20 ms, process B reappears. Since process A or the remainder of process C (whichever has not been selected previously to execute), has the earliest deadline, it is assigned the highest priority and will execute during this interval.



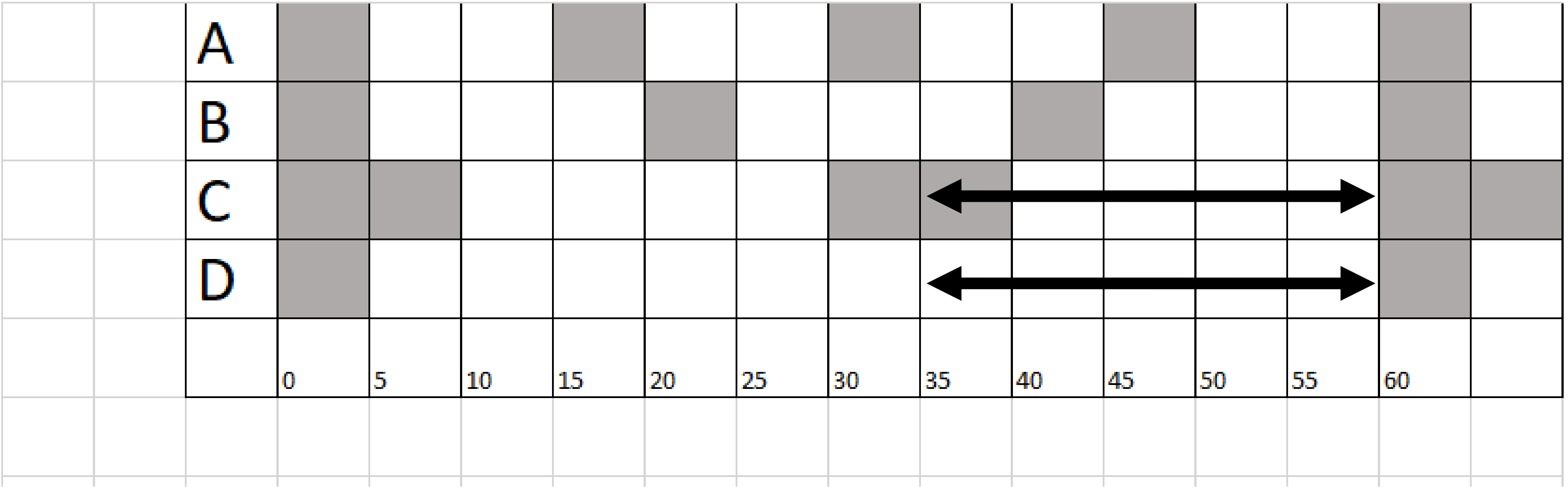
			A	C									
A	B	C			B								
			C	A									

At 25 ms, process B has the earliest deadline, is assigned the highest priority and executes.



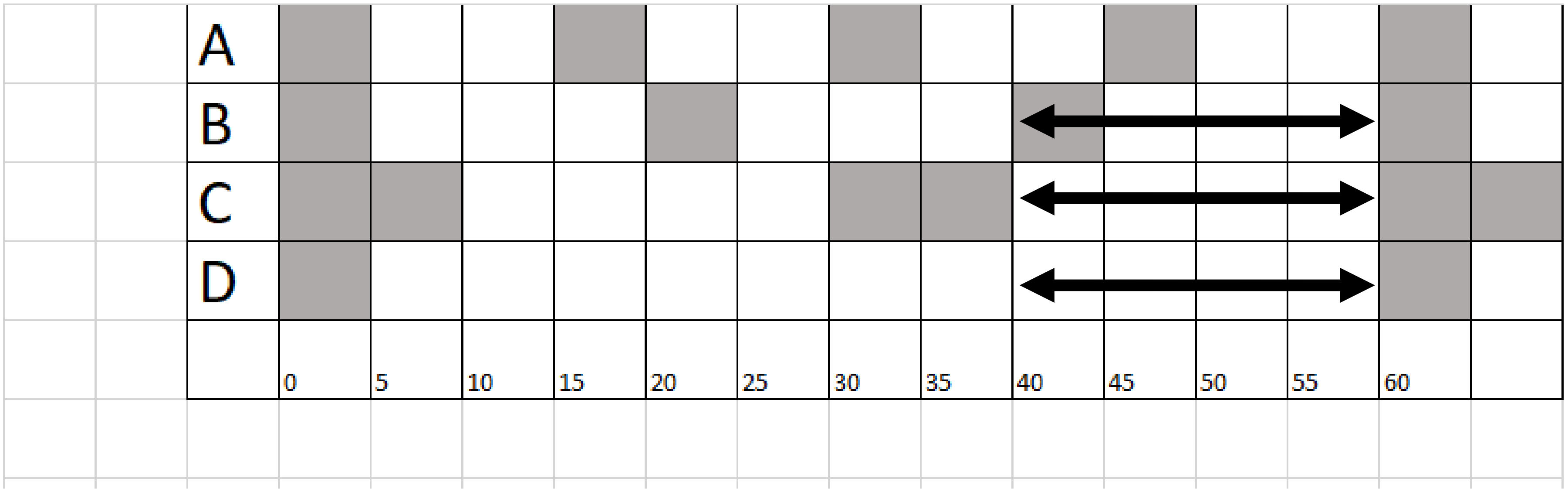
			A	C									
A	B	C			B	A							
			C	A									

At 30 ms, process A has the earliest deadline, is assigned the highest priority, and executes during this interval.



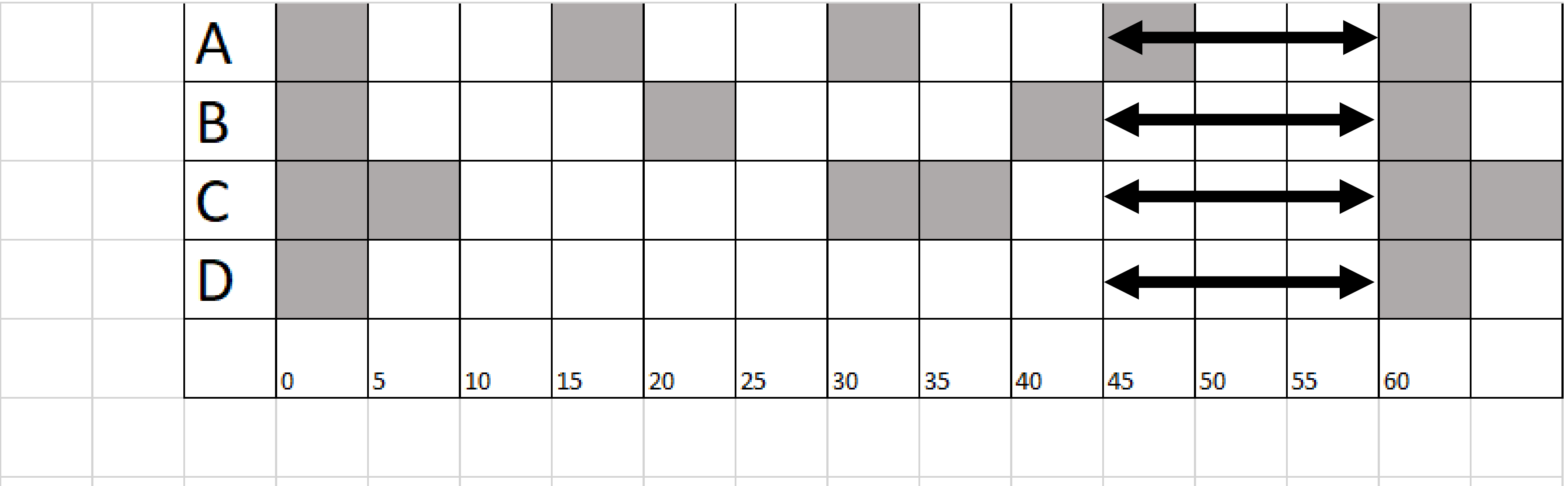
			A	C			C						
A	B	C			B	A							
			C	A			D						

At 35 ms, processes C and D have the same deadline, and thus have been assigned equal priority. Again, we can select either one to execute.



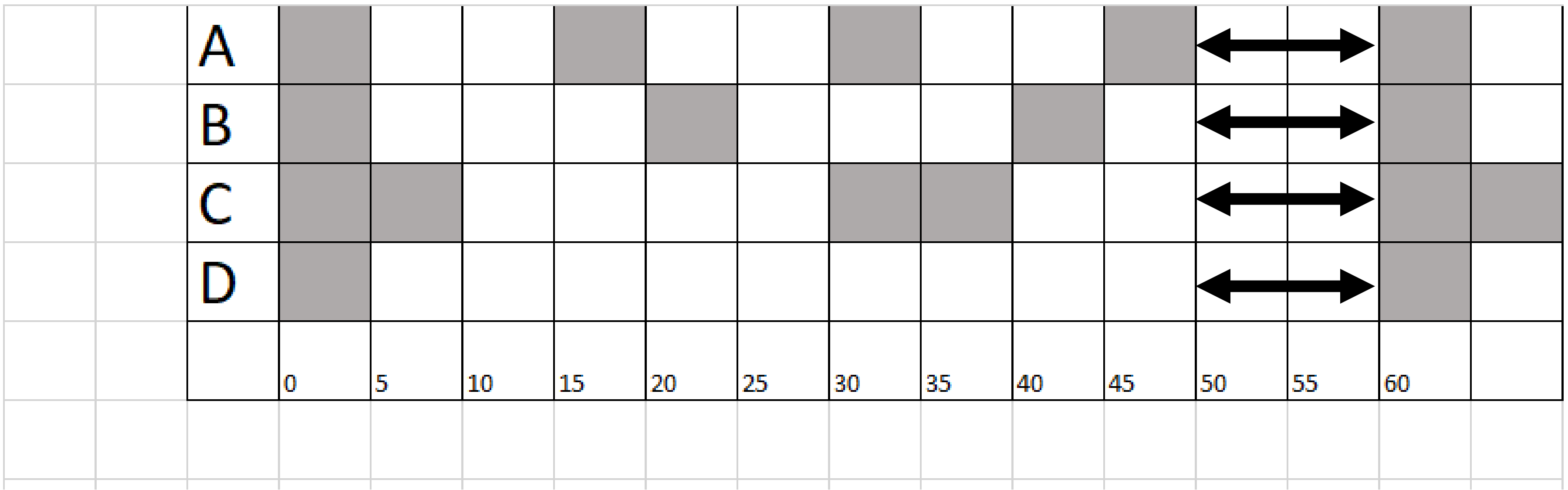
			A	C			C	D					
A	B	C			B	A							
			C	A			D	C					

At 40 ms, processes C or D (whichever was not executed at the previous interval) and B have the same deadline, and thus have been assigned equal priority.



			A	C			C	D	A				
A	B	C			B	A							
			C	A			D	C	B				

At 45 ms, processes A, B, C and D have the same deadline, so they are assigned equal priority. Either one can execute, as long as it has not executed at the previous interval.



			A	C			C	D	A	B			
A	B	C			B	A							
			C	A			D	C	B	A			

At 50 ms, processes A, B, C and D have the same deadline, so they are assigned equal priority. Either one can execute, as long as it has not executed at the previous interval.

A	▨			▨			▨			▨		↔	▨	
B	▨				▨				▨			↔	▨	
C	▨	▨					▨	▨				↔	▨	▨
D	▨											↔	▨	
	0	5	10	15	20	25	30	35	40	45	50	55	60	

			A	C			C	D	A	B	C		
A	B	C			B	A							
			C	A			D	C	B	A	C		

At 55 ms, processes A, B, C and D have the same deadline, so they are assigned equal priority. Either one can execute, as long as it has not executed at the previous interval.