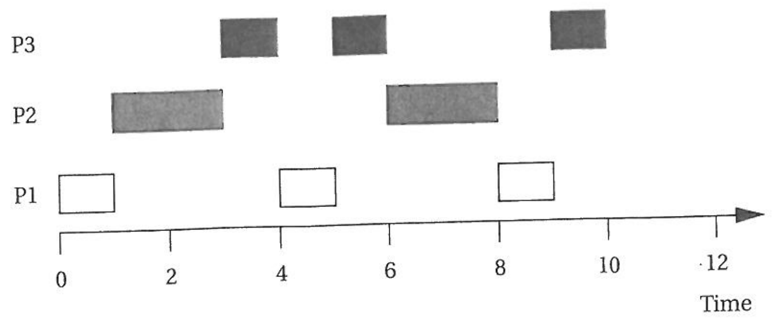
Example 6.3

Rate-monotonic scheduling

Here is a simple set of processes and their characteristics.

Process	Execution time	Period
P1	1	4
P2	2	6
P3	3	12

Applying the principles of RMA, we give P1 the highest priority, P2 the middle priority, and P3 the lowest priority. To understand all the interactions between the periods, we need to construct a time line equal in length to hyperperiod, which is 12 in this case.



All three periods start at time zero. P1's data arrive first. Since P1 is the highest-priority process, it can start to execute immediately. After one time unit, P1 finishes and goes out of the ready state until the start of its next period. At time 1, P2 starts executing as the highest-priority ready process. At time 3, P2 finishes and P3 starts executing. P1's next iteration starts at time 4, at which point it interrupts P3. P3 gets one more time unit of execution between the second iterations of P1 and P2, but P3 does not get to finish until after the third iteration of P1.

Consider the following different set of execution times for these processes, keeping the same deadlines.

Process	Execution time	Period
P1	2	4
P2	3	6
P3	3	12

In this case, we can show that there is no feasible assignment of priorities that guarantees scheduling. Even though each process alone has an execution time significantly less than its period, combinations of processes can require more than 100% of the available CPU cycles. For example, during one 12 time-unit interval, we must execute P1 three times, requiring 6 units of CPU time; P2 twice, costing 6 units of CPU time; and P3 one time, requiring 3 units of CPU time. The total of 6 + 6 + 3 = 15 units of CPU time is more than the 12 time units available, clearly exceeding the available CPU capacity.

Example 6.4

Earliest-deadline-first scheduling

Consider the following processes:

Process	Execution time	Period
P1	1	3
P2	1	4
P3	2	5

The hyperperiod is 60. In order to be able to see the entire period, we write it as a table:

Time	Running process	Deadlines
0	P1	
1	P2	
2	P3	P1
3	P3	P2
4	P1	P3
5	P2	P1
6	P1	
7	P3	P2
8	P3	P1
9	P1	P3
10	P2	
11	P3	P1, P2
12	P1	
13	P3	
14	P2	P1, P3
15	P1	P2
16	P2	
17	P3	P1
18	P1	
19	P3	P2, P3
20	P2	P1
21	P1	
22	P3	
23	P3	P1, P2
24	P1	P3
25	P2	
26	P3	P1
27	P1	P2
28	P3	
29	P2	P1, P3
30	idle	
31	P1	P2
32	P3	P1
33	P3	
34	P1	P3
35	P2	P1, P2
36	P1	
37	P2	
38	P3	P1
39	P3	P2, P3
40	P1	
41	P2	P1
42	P1	
43	P3	P2
44	P3	P1, P3
45	P1	
46	P2	
47	P3	P1, P2
48	P3	
49	P1	P3
50	P2	P1
51	P1	P2
52	P3	
53	P3	P1
54	P2	P3
55	P1	P2
56	P2	P1
57	P1	
58	P3	
59	P3	P1, P2, P3

(Continued)

There is one time slot left at $t = 30$, giving a CPU utilization of 59/60.