Priority-Based Scheduling (Periodic Tasks)

- "A preemptive method where the priority of the process determines whether it continues to run or is disrupted" (most important process first)
- On-line scheduler (does not precompute schedule)
- Fixed priorities:
  - same priority to all jobs in a task
- Dynamic priorities:
  - different priorities to individual jobs in each task
  - task-level dynamic priorities
  - job-level fixed priorities

RMS: Rate Monotonic Scheduling

- On-line
- Preemptive
- Priority-based with static priorities

- Period T_i that is the shortest interval between its arrival times
- Processes are assigned priorities dependent on length of T_i, the shorter it is, the higher the priority (or the higher the rate, the higher the priority)

$$T_i < T_j \Rightarrow P_i > P_j$$

- RM algorithm or RMS

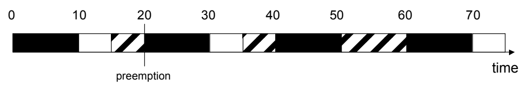
Example Priority Assignment

Process	Period	Priority
a	25	
b	60	
c	42	
d	105	
e	75	

Example

	P1	P2	P3
• Period (Ti)	20	50	30
• WCET (ei)	10	10	5
• Priority	high	low	medium

arrival time	process
t=0	P1,P2,P3
t=20	P1
t=30	P3
t=40	P1
t=50	P2
t=60	P1,P3



Schedulability Test

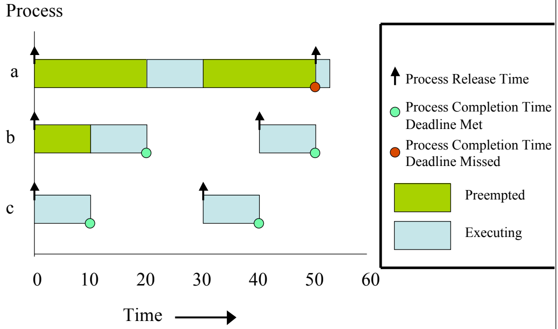
- For n processes, RMS will guarantee their schedulability if the total utilization U does not exceed the guarantee level $G = n * (2^{1/n} - 1)$
- When test fails:
  - try with the worst case: assume that all processes are released simultaneously at time 0, and then arrive according to their periods
  - check whether each process meets its deadline for all releases before the first deadline for the process with the lowest priority
- Otherwise:
  - change U by reducing c_i (code optimization, faster processor, …)
  - or increase T_i for some process (possible?)

Process Set A

Process	Period T	ComputationTime C	Priority P	Utilization U
a	50	12	1	0.24
b	40	10	2	0.25
c	30	10	3	0.33

- The combined utilization is 0.82 (or 82%)
- This is above the threshold for three processes (0.78) and, hence, this process set fails the utilization test

Timeline for Process Set A



Process Set B

Process	Period T	ComputationTime C	Priority P	Utilization U
a	80	32	1	0.400
b	40	5	2	0.125
c	16	4	3	0.250

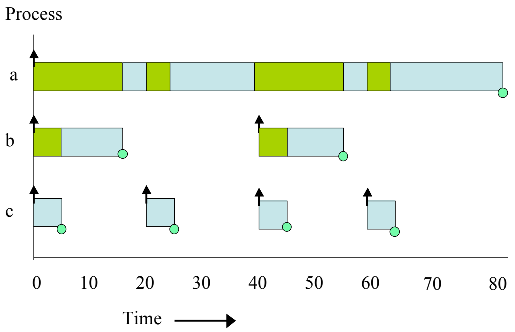
- The combined utilization is 0.775
- This is below the threshold for three processes (0.78) and, hence, this process set will meet all its deadlines

Process Set C

Process	Period T	ComputationTime C	Priority P	Utilization U
a	80	40	1	0.50
b	40	10	2	0.25
c	20	5	3	0.25

- The combined utilization is 1.0
- This is above the threshold for three processes (0.78) but the process set will meet all its deadlines

Timeline for Process Set C



Response Time Analysis

- Here task i's worst-case response time, R, is calculated first and then checked (trivially) with its deadline

$$R_i \le D_i$$

$$R_i = C_i + I_i$$

Where I is the interference from higher priority tasks

Calculating R

During R, each higher priority task j will execute a number of times:

$$\text{Number of Releases} = \left\lceil \frac{R_i}{T_j} \right\rceil$$

The ceiling function gives the smallest integer greater than the fractional number on which it acts. So the ceiling of 1/3 is 1, of 6/5 is 2, and of 6/3 is 2.

Total interference is given by:

$$\left\lceil \frac{R_i}{T_j} \right\rceil C_j$$

Reponse Time Equation

$$R_i = C_i + \sum_{j \in hp(i)} \left\lceil \frac{R_i}{T_j} \right\rceil C_j$$

Where $hp(i)$ is the set of tasks with priority higher than task i

Solve by forming a recurrence relationship:

$$w_i^{n+1} = C_i + \sum_{j \in hp(i)} \left\lceil \frac{w_i^n}{T_j} \right\rceil C_j$$

The set of values $w_i^0, w_i^1, w_i^2, \ldots, w_i^n, \ldots$ is monotonically non decreasing
When $w_i^n = w_i^{n+1}$ the solution to the equation has been found, w_i^0 must not be greater than R_i (e.g. 0 or C_i)

Process Set D

Process	Period T	ComputationTime C	Priority P
a	7	3	3
b	12	3	2
c	20	5	1

$R_a = 3$

$$w_b^0 = 3$$
$$w_b^1 = 3 + \left\lceil \frac{3}{7} \right\rceil 3 = 6$$
$$w_b^2 = 3 + \left\lceil \frac{6}{7} \right\rceil 3 = 6$$
$$R_b = 6$$

$$w_c^0 = 5$$
$$w_c^1 = 5 + \left\lceil \frac{5}{7} \right\rceil 3 + \left\lceil \frac{5}{12} \right\rceil 3 = 11$$
$$w_c^2 = 5 + \left\lceil \frac{11}{7} \right\rceil 3 + \left\lceil \frac{11}{12} \right\rceil 3 = 14$$
$$w_c^3 = 5 + \left\lceil \frac{14}{7} \right\rceil 3 + \left\lceil \frac{14}{12} \right\rceil 3 = 17$$
$$w_c^4 = 5 + \left\lceil \frac{17}{7} \right\rceil 3 + \left\lceil \frac{17}{12} \right\rceil 3 = 20$$
$$w_c^5 = 5 + \left\lceil \frac{20}{7} \right\rceil 3 + \left\lceil \frac{20}{12} \right\rceil 3 = 20$$
$$R_c = 20$$

Revisit: Process Set C

Process	Period	ComputationTime	Priority	Response time
	T	C	P	R
a	80	40	1	80
b	40	10	2	15
c	20	5	3	5

- The combined utilization is 1.0
- This was above the ulilization threshold for three processes (0.78), therefore it failed the test
- The response time analysis shows that the process set will meet all its deadlines
- RTA is necessary and sufficient

Response Time Analysis

- Is sufficient and necessary
- If the process set passes the test they will meet all their deadlines; if they fail the test then, at run-time, a process will miss its deadline (unless the computation time estimations themselves turn out to be pessimistic)

Exact Schedulability Test

```
for (each task T_j) {
    I = 0;
    do {
        R = I + c_j
        if (R > d_j) return UNSCHEDULABLE;
        I = ∑⌈R/p_k⌉c_k; /* k = 1..j-1 */
    } while (I + c_j > R)
    return SCHEDULABLE;
}
```

Deadline-Monotonic Algorithm (DM)

- Fixed-priority
- Uses relative deadlines: the shorter the relative deadline, the higher the priority
- RM and DM are identical if the relative deadline is proportional to its period
- Otherwise DM performs better in the sense that it can sometimes produce a feasible schedule when RM fails, while RM always fails when DM fails

Example