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

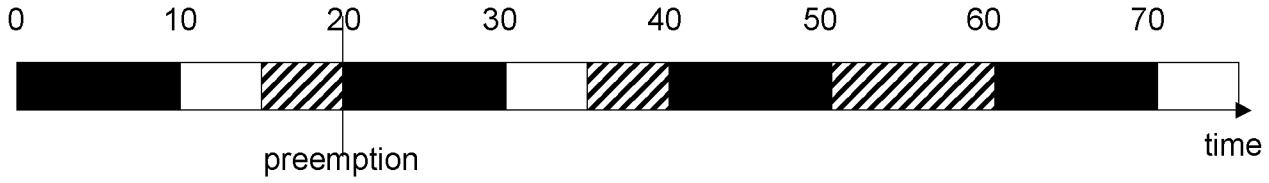
- On-line
- Preemptive
- Priority-based with static priorities

- 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)
- RM algorithm or RMS

Example

	P1	P2	P3
Period (T_i)	20	50	30
WCET (e_i)	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 $G = n * (2^{1/n} - 1)$
- Sufficient condition
- 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 lowest priority
- Otherwise:
  - change U by reducing C_i (code optimization, faster processor, ...)
  - or increase T_i for some process (possible?)

Theorems

- $G=n * (2^{1/n} - 1)$
- Optimality:
  - RMS is optimal among methods with fixed priority (in what sense?)
- Lowest upperbound:
  - for arbitrarily large n, it suffices that processor utilization is < 0.69
  - (proof in various RT books, e.g., Buttazzo)
- Example:

	P1	P2	P3
– period	20	50	30
– WCET	7	10	5

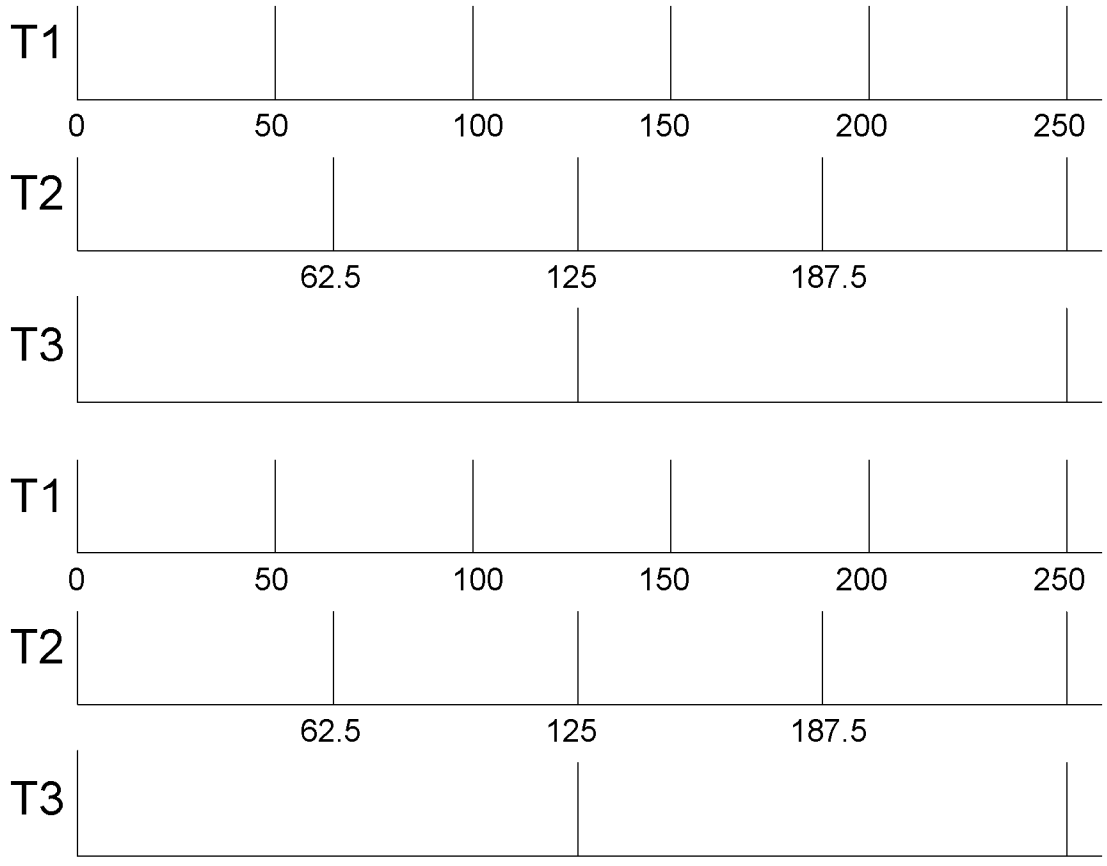
Exact Schedulability Test

```
for (each task T_j) {
    I=0;
    do {
        R = I + c_j;
        if (R > d_j) return (UNSCHEDULABLE);
        I = Σ⌈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


T1
0
50
100
150
200
250
T2
62.5
125
187.5
T3
T1
0
50
100
150
200
250
T2
62.5
125
187.5
T3