

Lab 2 for uC/OS-II: EDF Scheduler

Prof. Li-Pin Chang
ESSLab@NCTU

Objective

- To implement an EDF scheduler in uC/OS-II

Fixed-Priority Scheduling

- uC/OS-II supports fixed-priority scheduling
 - Easy to implement RM
- There is no EDF support
 - Tasks have dynamic priorities
 - Job's “urgency” are determined upon their arrivals
 - Must associate every job with a deadline

Adding Support for EDF

- Identify where/when scheduling decisions are made
 - OS_Sched, OSIntExit(), OSStart()
- Add proper deadline information to task information (i.e., in TCB)
- Add code to pickup a ready job with the earliest deadline at the re-scheduling points

Deadlines and Priorities

- Task creation should remain the same for a minimal invasive modification
- Your scheduler will pick up a ready task whose deadline is the earliest
 - Unlike priorities, the value domain of deadlines are infinite
 - The best solution would be using a heap
 - But in this project, performance is not a concern; it is okay to use linear search

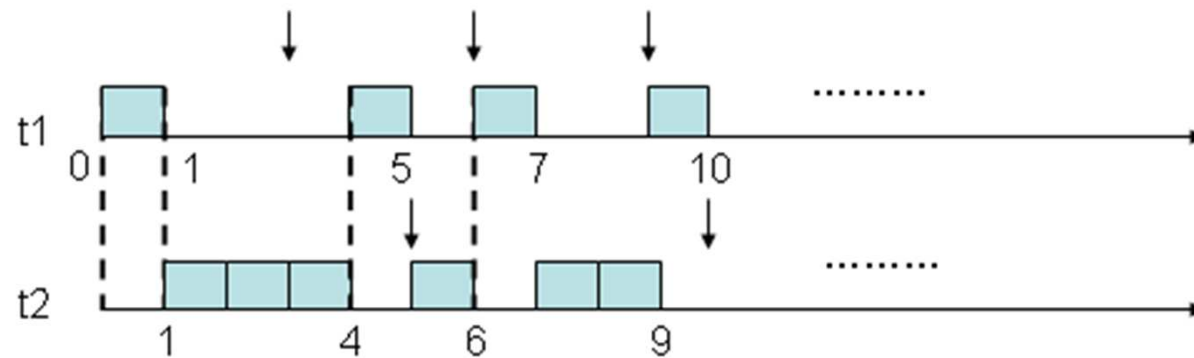
Periodic Tasks

- Reuse your code of Lab 0 to simulate periodic tasks

```
while(1)
{
    while (OSTCBCur->CompTime > 0)
    {
        // do nothing
    }
    ...
    OSTimeDly(...);
}
```

Taksets for Tests

- Task set1 = { $t1(1,3)$, $t2(3,5)$ }



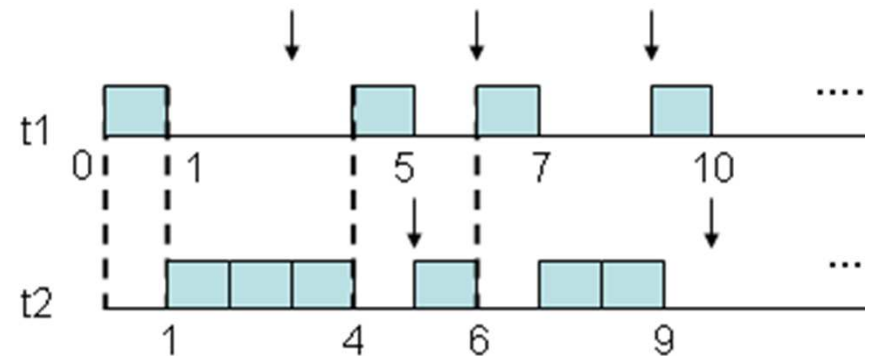
- Task set2 = { $t1(1,4)$, $t2(2,5)$, $t3(2,10)$ }

Output

- Following the same format of Project 0

Time event from to

0	Preempt	63	1
1	Complete	1	2
4	Complete	2	1
5	Complete	1	2
6	Preempt	2	1



More Hints

- Add deadline information in the TCB
 - You can pass them to tasks upon creation via the user-provided parameter
- Upon re-scheduling, visit the TCB list linearly; find the ready task whose deadline is the earliest
 - `ptcb->OSTCBStat` is `OS_STAT_RDY`?
 - Rescheduling points are `OSIntExit`, `OS_Sched`, `OSStart`
 - Linear search is actually a bad practice... but is excused here...
- Before a task delays for the next period, advance its deadline to the next period