Real Time Systems 2

Task Assignment

Next-fit algorithm for RM

Task Assignment

- The optimal assignment of tasks to CPUs is an NP-complete problem. We must therefore make do with heuristics.
- These heuristics can not guarantee that an allocation will be found that permits all tasks to be feasibly scheduled.
- All that we hope to do is to allocate the tasks, check their feasibility and, if the allocation is not feasible, modify the allocation to try to render its schedules feasible.

Task Assignment

- When checking an allocation for feasibility, we must account for communication costs.
- For example, suppose that T1 PRECEDES T2. Task T2 can not start before receiving that output of T1.
- That is, if fi denotes the completion time of task Ti and Cij is the time to communicate from Ti to Tj

$R2 >= f1+C12$

- If tasks T1 and T2 are allocated to the same CPU, then C12=0.
- If they are allocated to separate CPU, C12 is positive and must be taken into account.

Task Assignment/ Next-fit algorithm for RM

- There is a utilization-based allocation heuristic that is meant specifically to be used in conjunction with RM scheduling algorithm.
- The task set is (independent, Pre-emptible, and periodic)
- The multiprocessor is assumed to consist of identical CPUs.
- Tasks are assumed to require no resources other than CPU time.

Task Assignment/ Next-fit algorithm for RM

- Define M>3 classes as follows:, where M is picked by the user.
- Task Ti is in class j <M if
- $2^{1/(j+1)} - 1 < e_i / P_i <= 2^{1/j} - 1$, and in class M otherwise.
- Corresponding to each task class is a set of CPUs that is only allocated to tasks of that class.
- We allocate tasks one by one to the appropriate CPU class until all the tasks have been scheduled.
- Adding CPUs to classes if that is needed for RM- schedulability.

Task Assignment/ Next-fit algorithm for RM

Example:

Suppose M=4 classes.

Then the following table lists the utilization bounds corresponding to each class.

Class	Bound
C1	(0.41,1]
C2	(0.26,0.41]
C3	(0.19,0.26]
C4	(0,0.19]

Next-fit algorithm for RM/ Example

- Consider the following periodic task set:

	Exi	Pi	U(i)=exi/Pi	Class
T1	5	10	0.5	C1
T2	7	21	0.33	C2
T3	3	22	0.14	C4
T4	1	24	0.04	C4
T5	10	30	0.33	C2
T6	16	40	0.40	C2
T7	1	50	0.02	C4
T8	3	55	0.05	C4
T9	9	70	0.13	C4
T10	17	90	0.19	C4
T11	21	95	0.22	C3

Next-fit algorithm for RM/ Example

- Since we have at least one task in each of the four classes, lets begin by earmarking one processor for each class.
- Let processor CPUi be reserved for tasks in class Ci. $1<=i<=4$
- T1 is assigned to CPU1
- T2 is assigned to CPU2
- T3 is assigned to CPU4
- T4 is assigned also to CPU4
- And since {T2,T5} is RM-schedulable on the same CPU, we assign T5 to CPU2

Next-fit algorithm for RM/ Example

- Don't forget, RM schedulable if utilization $< n(2^{1/n} - 1)$
- T6 also C2 but we can not assign it to CPU2 also because it will be unschedulable under RM on the same CPU, so we assign an additional CPU5 to C2 and assign T6 to CPU5.
- T7 belongs to C4 and {T3,T4,T7} is RM-schedulable on the same CPU.
- The same for T8,T9,T10 to CPU4.
- T11 belongs to C3 and assigned to CPU3.

Next-fit algorithm for RM/ Example

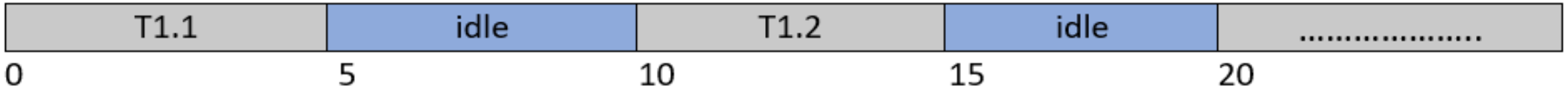
- We summarize that we need 5 CPUs as the following:

Processor	Tasks
CPU1	T1
CPU2	T2,T5
CPU3	T11
CPU4	T3,T4,T7,T8,T9,T10
CPU5	T6

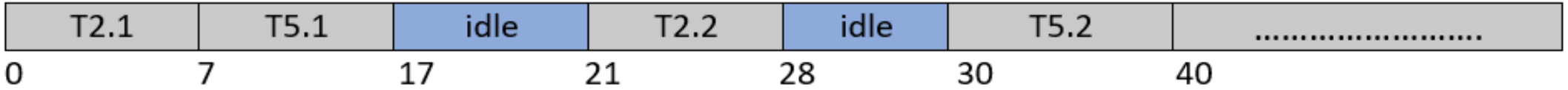
  - For every CPU draw the chart for scheduling it's tasks.

Next-fit algorithm for RM/ Example

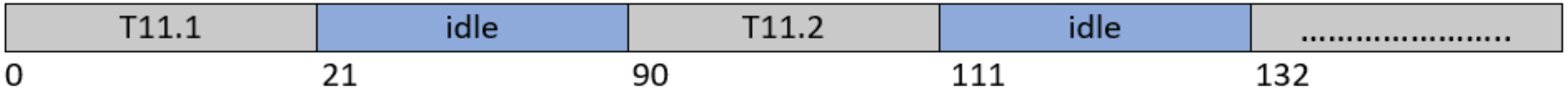
CPU1:



CPU2:

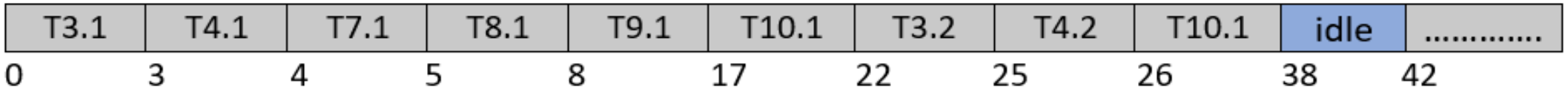


CPU3:

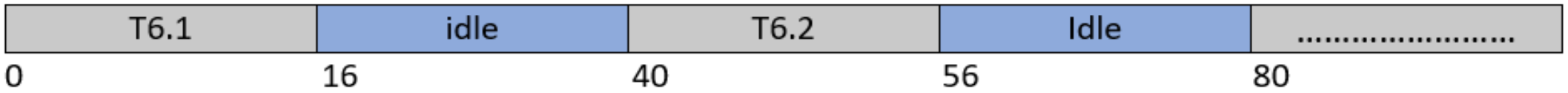


CPU4:

$Per_{T3} < Per_{T10}$ $Per_{T4} < Per_{T10}$



CPU5:



Next-fit algorithm for RM/ Hw

- Consider the following periodic task set:
- Assign them to CPUs using Next-fit algorithm.
- M=4

Tasks	Exi	Pi	U(i)=exi/Pi	Class
T1	5	10		
T2	17	25		
T3	7	35		
T4	10	60		
T5	15	20		
T6	6	20		
T7	7	30		
T8	10	60		
T9	5	30		
T10	10	30		