Scheduling real time tasks in multiprocessor and distributed systems

- Scheduling real time tasks consists of two problems:
  - Task assignment
  - The task assignment problem is concerned with how to partition a set of tasks and then how to assign these to the processors. Task assignment can either be static or dynamic.
  - Task allocation
  - After task assignment to the processors, consider the tasks on each processor individually and therefore the second phase of the multiprocessor and distributed systems reduces to the scheduling problem.

Multiprocessor Task Allocation

- Utilization Balancing Algorithm
- Next Fit Algorithm

Utilization Balancing Algorithm

- This algorithm is suitable when the number of processors in a multiprocessor is fixed.
- Maintains the tasks in increasing order of their utilizations.
- It removes the tasks one by one from the head of the queue and allocates them to the least utilized processor each time. (To achieve balanced utilization of the different processors).
- The objective of selecting the least utilized processor each time is to balance utilization of the different processors.
- In a perfectly balanced system, utilization at each processor u_i = $\bar{u}$ overall utilization of the processors.
- The objective of balancing algorithm is to minimize $\sum_{i=1}^{n} |(\bar{u} - u_i)|$
  - n is number of processors in the system
  - $\bar{u}$ is average utilization of processors
  - u_i is utilization of processor i.

Next Fit Algorithm (RMA)

- This algorithm attempts to use as few processors as possible.
- The algorithm classifies the different tasks into a few classes based on the utilization of the task.
- One or more processors are assigned exclusively to each class of tasks.
- The tasks with similar utilization values are scheduled on the same processor.
- If the tasks are to be divided into m classes, a task T_i belongs to a class j, $0 \leq j < m$, iff

$$(2^{\frac{1}{j+1}} - 1) < e_i/p_i \leq (2^{\frac{1}{j}} - 1)$$

- Suppose the tasks of a system are to be partitioned into four classes

$$\textbf{Class 1: } (2^{\frac{1}{2}} - 1) < C_1 \leq (2^{\frac{1}{1}} - 1)$$
$$\textbf{Class 2: } (2^{\frac{1}{3}} - 1) < C_2 \leq (2^{\frac{1}{2}} - 1)$$
$$\textbf{Class 3: } (2^{\frac{1}{4}} - 1) < C_3 \leq (2^{\frac{1}{3}} - 1)$$
$$\textbf{Class 4: } 0 < C_4 \leq (2^{\frac{1}{4}} - 1)$$

- Class 1 : (0.41,1)
- Class 2: (0.26, 0.41)
- Class 3: (0.19, 0.26)
- Class 4: (0, 0.19)

Problem

The following table shows the execution time (in ms) and periods (in ms) of a set of 10 periodic real time tasks. Assume that the tasks need to run on a multiprocessor with 4 processors. Allocate the tasks to processors using the next fit algorithm

Task	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}
e_i	5	7	3	1	10	16	1	3	9	17
p_i	10	21	22	24	30	40	50	55	70	100

Solution

Task	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}
e_i	5	7	3	1	10	16	1	3	9	17
p_i	10	21	22	24	30	40	50	55	70	100
u_i	0.5	0.33	0.14	0.04	0.33	0.4	0.02	0.05	0.13	0.17
Class	1	2	4	4	2	2	4	4	4	3

Bin Packing Algorithm for EDF

- This algorithm attempts to allocate tasks to the processors such that the tasks on the individual processors can be successfully scheduled using EDF.
- Tasks are to be assigned to processors such that the utilization at any processor does not exceed 1.
- First fit random algorithm
  - Tasks are selected randomly and assigned to processors in an arbitrary manner as long as the utilization of a processor does not exceed 1.
- First fit decreasing algorithm
  - The tasks are sorted in non-increasing order of their CPU utilization, the tasks are selected one by one from the ordered list and assigned to the processor to which it can fit in