ASSIGNMENT 5
Simulation Assignment

Name: K.Vishwanath
Course Title: Development of Real-Time System
Date: 05-07-2022

Assignment:
Modify the python code in P_RM.py to use firstfit instead of the current algorithm. Please follow the steps in This document.

-Hint: Instead of scheduling the task to the CPU with the lowest utilisation chose the first one which has a lower utilisation than Urm(x+1) where x is the already scheduled tasks on the CPU

-Hint2: have a look at the def packer(self) function in the file P_RM.py

-Schedule the following task set on three processors using your modified algorithm.

T1(2,1) T2(2.5,0.1) T3(3,1) T4(4,1) T5(4.5,0.1) T6(5,1) T7(6,1) T8(7,1) T9(8,1) T10(8.5,0.1) T11(9,1)

P_RM.py source code:

```
"""
Partitionned EDF using PartitionedScheduler.
"""
from simso.core.Scheduler import SchedulerInfo
from simso.utils import PartitionedScheduler
from simso.schedulers import scheduler

@scheduler("simso.schedulers.P_RM")

class P_RM(PartitionedScheduler):
    def init(self):
        PartitionedScheduler.init(
            self, SchedulerInfo("simso.schedulers.RM_mono"))
    def packer(self):

        # First Fit

        cpus = [[cpu, 0] for cpu in self.processors]

        numCPUs = len(cpus)

        print("CPU num: ", numCPUs)
```

```
        taskNUM = [0] * numCPUs
        Urm = 0.0
        U = 0.0
        for task in self.task_list:
            #m = cpus[0][1]
            j = 0
            # Find the processor with the lowest load.
            for i, c in enumerate(cpus):
                Urm = (taskNUM[i]+1.0) * ((pow(2.0, 1/(taskNUM[i]+1.0))) -
1.0)
                U = (c[1] + (task.wcet / task.period))
                print("CPU U = ",c[1])
                print("U after scheduling = ",U)
                print("Urm = ", Urm)
                if U < Urm:
                    j = i
                    break

            taskNUM[j] = taskNUM[j] + 1
            print("CPU scheduled = ",j)
            print("Tasks = ", taskNUM)
            # Affect it to the task.
            self.affect_task_to_processor(task, cpus[j][0])
            # Update utilization.
            cpus[j][1] += float(task.wcet) / task.period
        return True
```

Model data

General | Scheduler | Processors | Tasks

id	Name	Task type	Abort on miss	Act. Date (ms)	Period (ms)	List of Act. dates (ms)	Deadline (ms)	WCET (ms)	Followed by
1	TASK T1	Periodic	Yes	0	2.0	-	2.0	1.0	
2	TASK T2	Periodic	Yes	0	2.5	-	2.5	0.1	
3	TASK T3	Periodic	Yes	0	3.0	-	3.0	1.0	
4	TASK T4	Periodic	Yes	0	4.0	-	4.0	1.0	
5	TASK T5	Periodic	Yes	0	4.5	-	4.5	0.1	
6	TASK T6	Periodic	Yes	0	5.0	-	5.0	1.0	
7	TASK T7	Periodic	Yes	0	6.0	-	6.0	1.0	
8	TASK T8	Periodic	Yes	0	7.0	-	7.0	1.0	
9	TASK T9	Periodic	Yes	0	8.0	-	8.0	1.0	
10	TASK T10	Periodic	Yes	0	8.5	-	8.5	0.1	
11	TASK T11	Periodic	Yes	0	9.0	-	9.0	1.0	

Edit data fields...

Remove selected task(s) | Add task | Generate Task Set

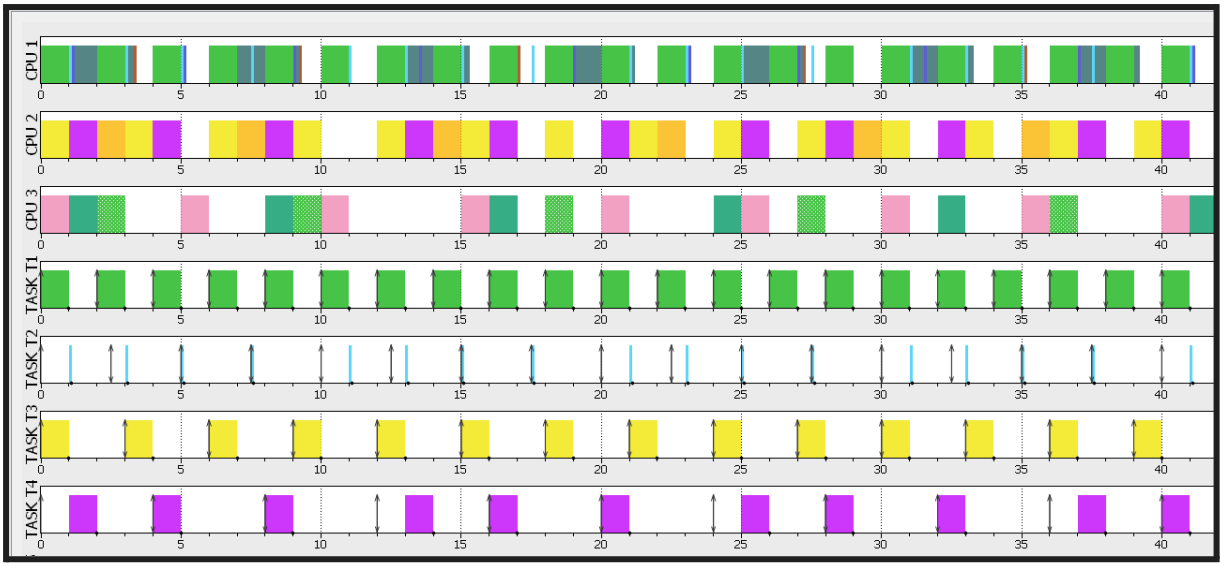
Results

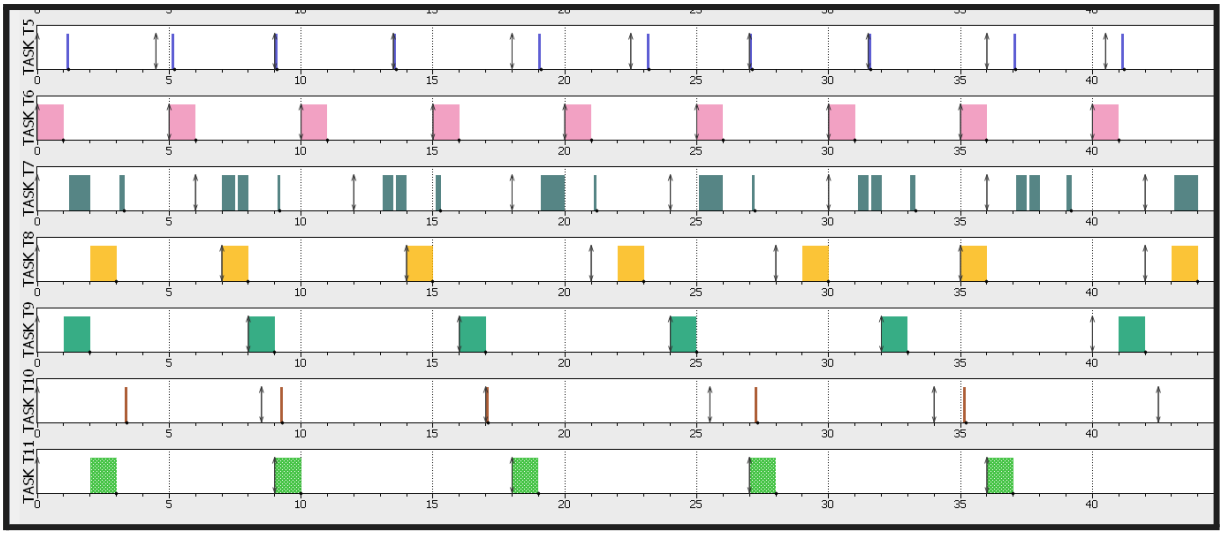
General | Logs | Tasks | Scheduler | Processors

Observation Window:

from 0.00 to 100.00 ms | Configure...

	Total load	Payload	System load
CPU 1	0.7450	0.7450	0.0000
CPU 2	0.7400	0.7400	0.0000
CPU 3	0.4500	0.4500	0.0000
Average	0.6450	0.6450	0.0000


CPU 1
CPU 2
CPU 3
TASK T1
TASK T2
TASK T3
TASK T4
0
5
10
15
20
25
30
35
40


TASK T5
TASK T6
TASK T7
TASK T8
TASK T9
TASK T10
TASK T11
0
5
10
15
20
25
30
35
40