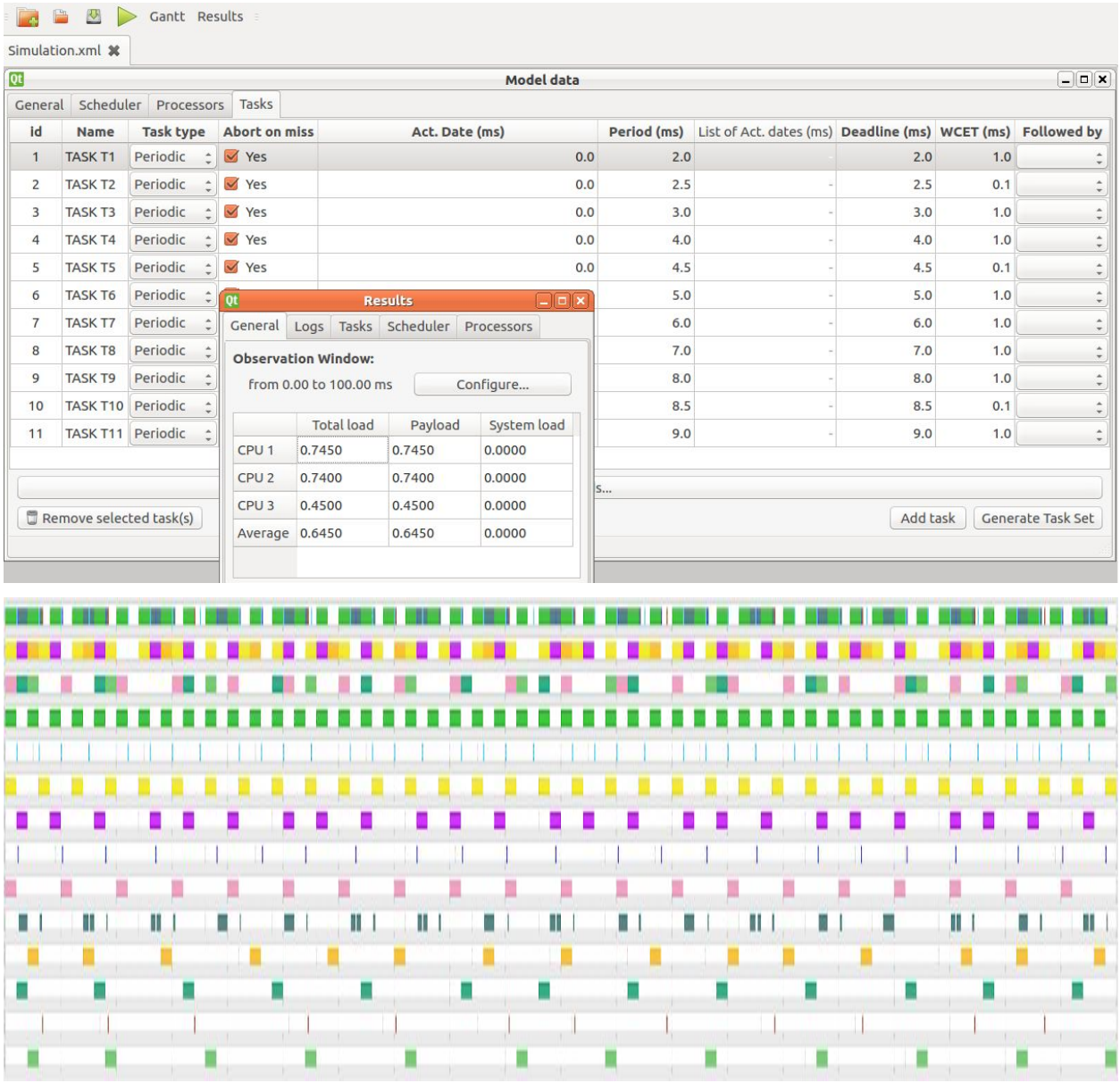
Development of Real-Time system

Assignment 5:

Simulation assignment:

- A picture of the final schedule using the modified scheduler



I have attached the task scheduling image for better visibility in the folder.

- The source code of the scheduling algorithm

```
"""
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
```

Programming assignment: