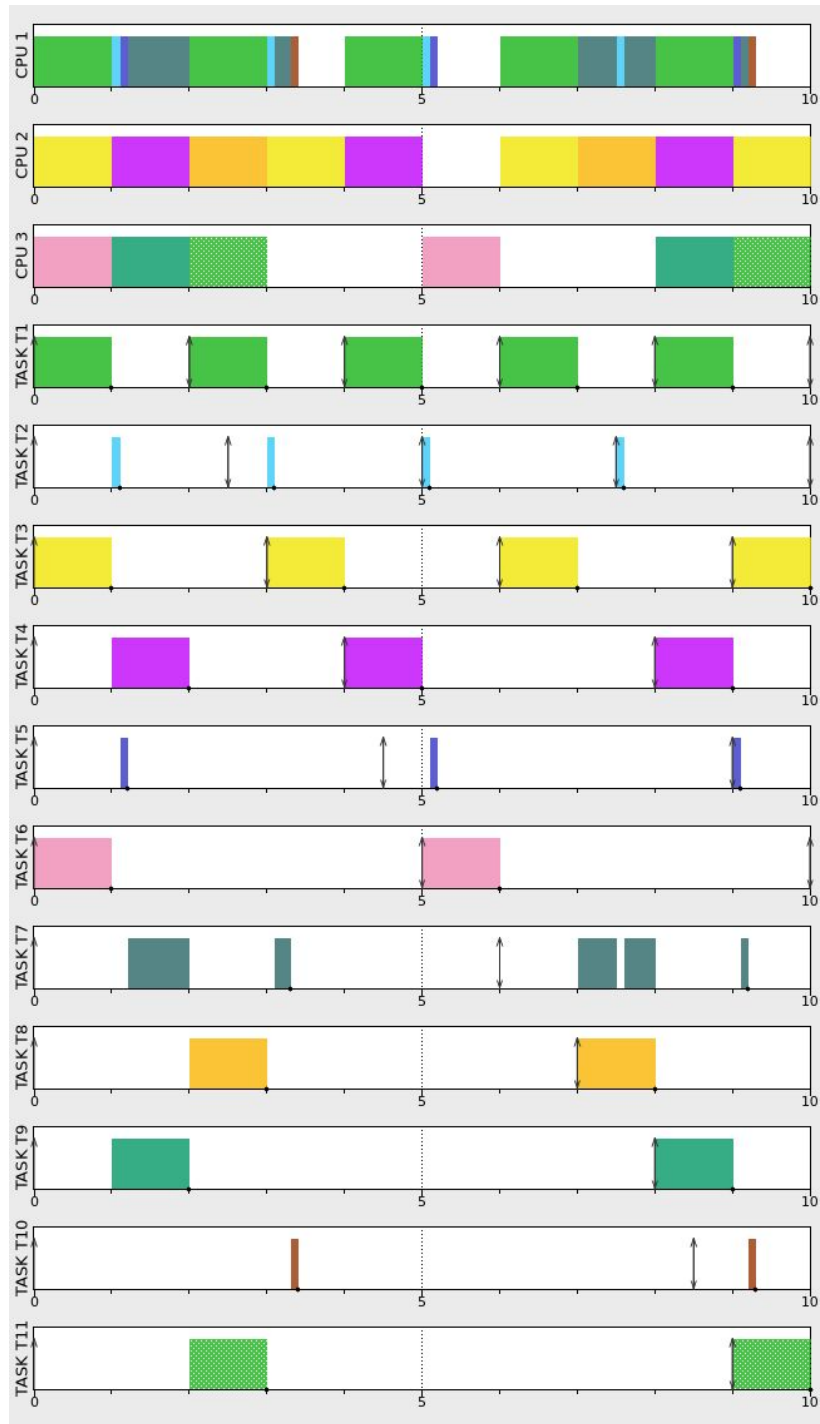


Simulation assignment

- A picture of the final schedule using the modified scheduler



- The source code of the scheduling algorithm

```
"""Rate Monotonic First Fit scheduler 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, 0] for cpu in self.processors]
        for task in self.task_list:
            m = cpus[0][1]
            j = 0

            # Find the processor with the first fit of total utilization less than
            Urm.

            for i, c in enumerate(cpus):
                # Calculate Urm
                k = c[2] + 1
                Urm = k * (2 ** (1 / k) - 1)

                # Calculate U total utilization
                U = c[1] + float(task.wcet) / task.period

                if U < Urm:
                    j = i
                    break

            # Affect it to the task.
            self.affect_task_to_processor(task, cpus[j][0])

            # Update utilization.
            cpus[j][1] += float(task.wcet) / task.period

            # Update number of tasks
            cpus[j][2] += 1

        return True
```

Programming assignment

- A screenshot of the execution

