

Lucas Martins

12011ECP022

lucas.martins@ufu.br

Coursera: Real Time Systems - Assignment 5

Instruções

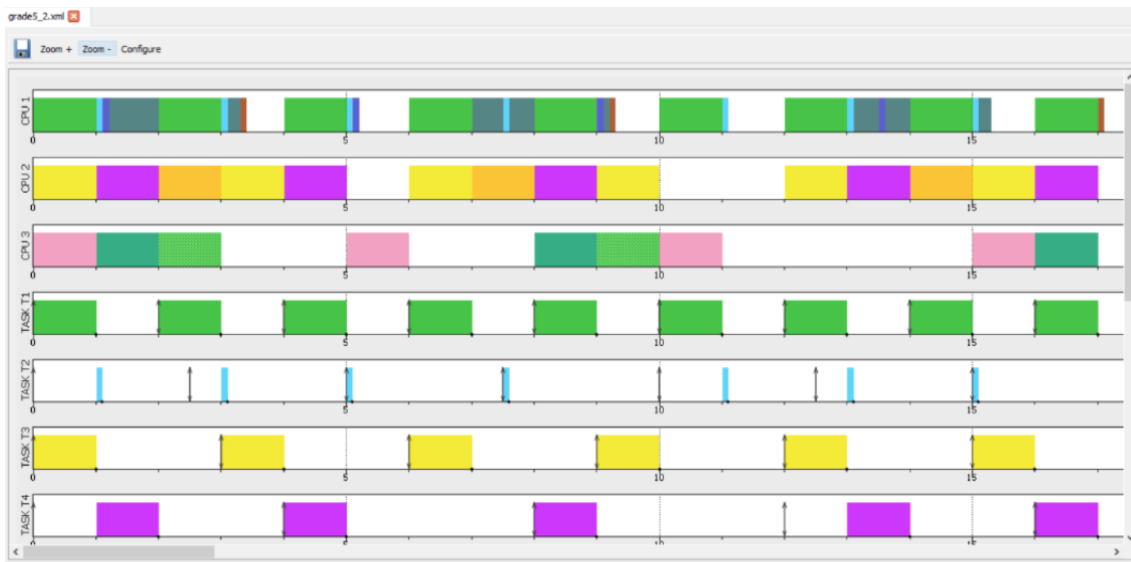
Área de upload do assignment 5 do curso Development of Real-Time Systems (<https://www.coursera.org/learn/real-time-systems>)

Simulação: modificar um simulador em tempo real para verificar a viabilidade de um conjunto de tarefas

- Uma imagem (screenshot) da programação final usando o programador modificado
- O código-fonte do algoritmo de agendamento

```
main.c D:\...Trabalho_04  C main.c D:\...Trabalho_05  PartitionedScheduler.py X  Settings
> UFU > SCTR > Trabalho_05 > PartitionedScheduler.py
1  from simso.core.Scheduler import SchedulerInfo
2  from simso.utils import PartitionedScheduler
3  from simso.schedulers import scheduler
4
5  @scheduler("simso.schedulers.P_RM")
6  class P_RM(PartitionedScheduler):
7      def init(self):
8          PartitionedScheduler.init(
9              self, SchedulerInfo("simso.schedulers.RM_mono"))
10
11     def packer(self):
12         # First Fit
13         cpus = [[cpu, 0] for cpu in self.processors]
14         numCPUs = len(cpus)
15         print "CPU num: ", numCPUs
16         taskNUM = [0] * numCPUs
17         Urm = 0.0
18         U = 0.0
19         for task in self.task_list:
20             #m = cpus[0][1]
21             j = 0
22             # Encontre o processador com a menor load.
23             for i, c in enumerate(cpus):
24                 Urm = (taskNUM[i]+1.0) * ((pow(2.0, 1/(taskNUM[i]+1.0))) - 1.0)
25                 U = (c[1] + (task.wcet / task.period))
26                 print "CPU U = ",c[1]
27                 print "U after scheduling = ",U
28                 print "Urm = ", Urm
29                 if U < Urm:
30                     j = i
31                     break
32
33             taskNUM[j] = taskNUM[j] + 1
34             print "CPU scheduled = ",j
35             print "Tasks = ", taskNUM
36
37             # Affect para a task.
38             self.affect_task_to_processor(task, cpus[j][0])
39
40             # Atualiza.
41             cpus[j][1] += float(task.wcet) / task.period
42
43     return True
```

Resultados:



Programação: crie tarefa específica e a comunicação por filas

- **Inclua uma imagem (screenshot) da execução.**

[illegible]