ERTS – HW1

Question 1

a. Efficiency: Computation Time / Total Time = 4.5 / 50 = 0.09 = 9%

b. Response Times:

a. Task 1) min=1.000, avg=1.667, max=0.624

b. Task 2) min=1.500, avg=1.500, max=1.500

c. Task 3) min=7.000, avg=7.250, max=7.500

c. Preempted and Aborted:

a. Task 1) No preemptions nor abortions.

b. Task 2) No preemptions nor abortions.

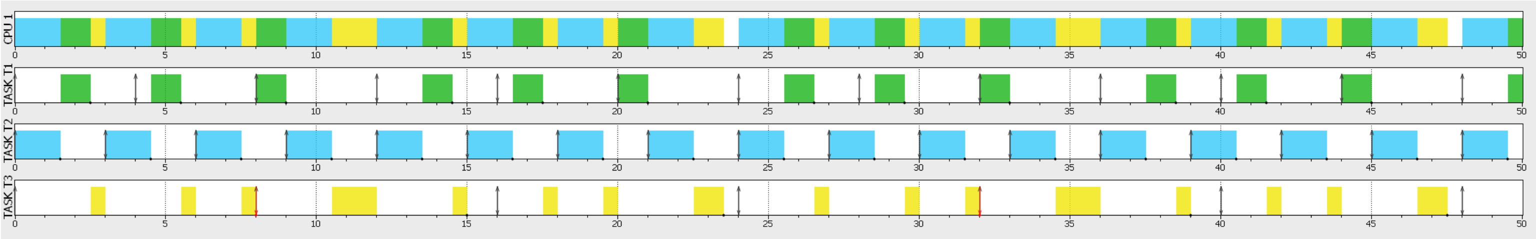
a. Task 3) 8s, 32s

d. Yes indeed. According to properties of Optimal algorithms, if a task set is not schedulable by an optimal algorithm, then it cannot be scheduled by any other algorithm.

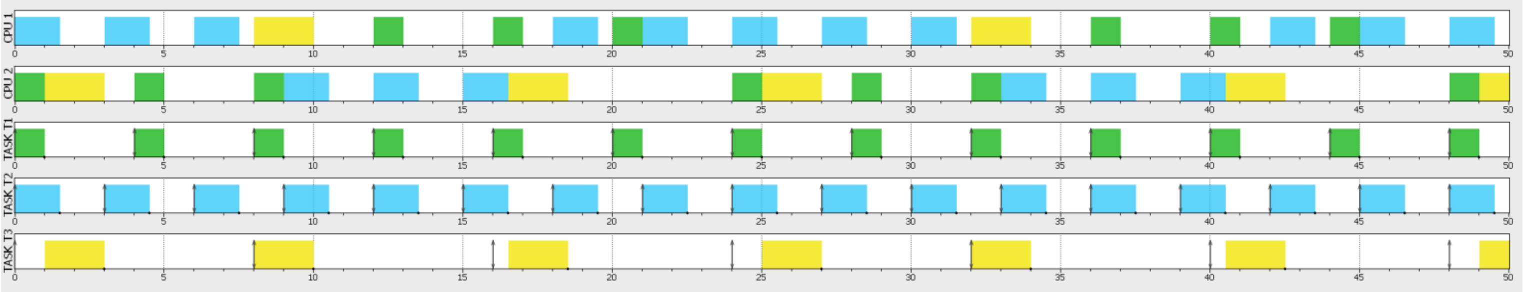
e. The pictures are also attached in the same .ZIP file...

With one processor, CPU is almost always doing something, but with 2 CPUs, there are times where one or both CPUs are Idle

Single Processor:



Dual Processor:



Question 2

b. Efficiency = Computation Time / Total Time = 6.1 / 50 = 0.12 = 12%

c. Response Times:

a. Task 1) min=0.400, avg=0.400, max=0.400

b. Task 2) min=2.000, avg=2.000, max=2.000

c. Task 3) - (they don’t get the time to execute)

d. Task 4) - (they don’t get the time to execute)

d. Preempted and Aborted:

a. Task 1) No preemptions nor abortions.

b. Task 2) Preempted or aborted at 4s, 8s, 12s, 16s, 20s, 24s, 28s, 32s, 36s, 40s, 44s, 48s

   c. Task 3) Preempted or aborted at 3s, 7.5s, 11.5s, 16.5s, 19.5s, 23.5s, 27.5s, 31.5s, 35.5s, 39.5s, 43.5s, 47.5

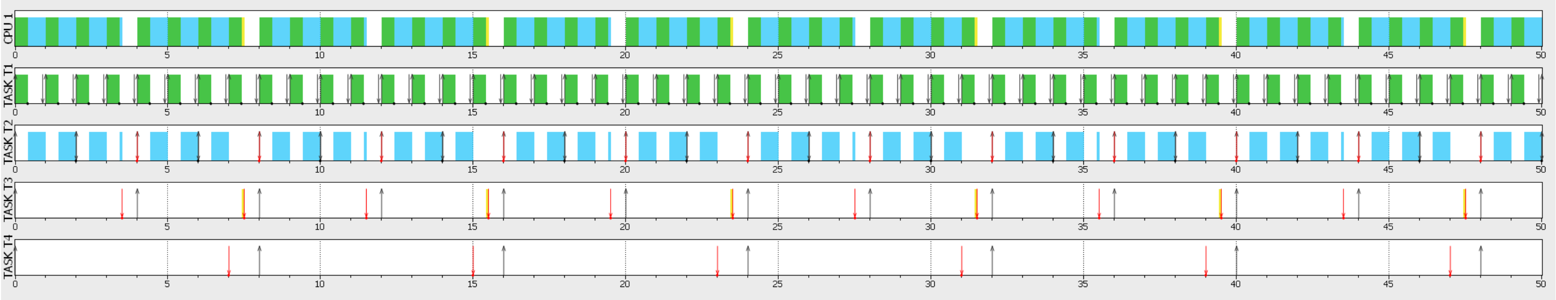
   d. Task 4) Preempted or aborted at 7s, 15s, 23s, 31s, 39s, 47s

e. Yes indeed. According to properties of Optimal algorithms, if a task set is not schedulable by an optimal algorithm, then it cannot be scheduled by any other algorithm.

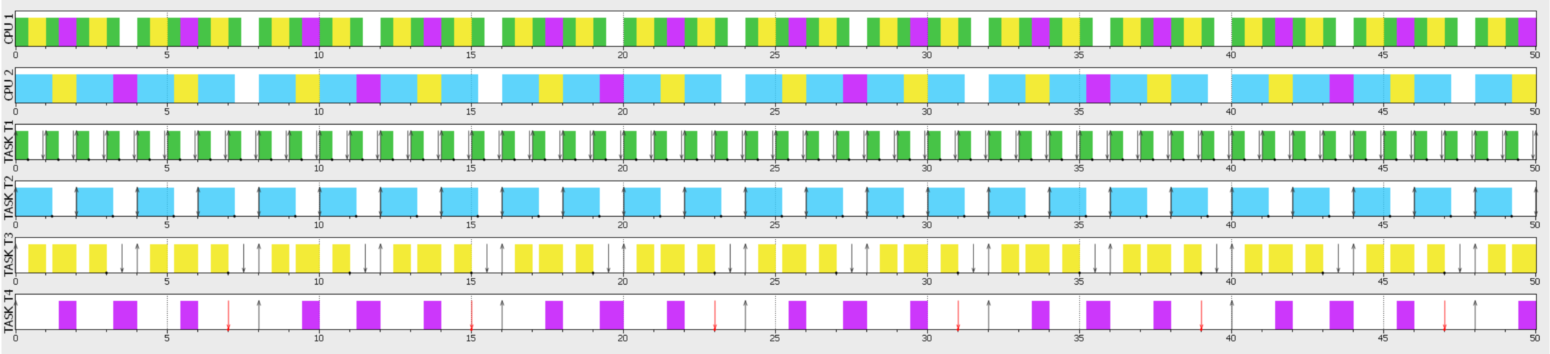
f. The pictures are also attached in the same .ZIP file...

   With two CPUs working together, we notice fewer preemptions and abortions!

   Single Processor:



   Dual Processor:



Question 3

a. Efficiency = Computation Time / Total Time = 11.6 / 100 = 0.116 = 11.6%

b. Response Times:

   a. RM:

      i. Task 1) min=0.400, avg=0.525, max=0.900

      ii. Task 2) min=2.000, avg=2.000, max=2.000

      iii. Task 3) - (they don't get the time to execute)

      iv. Task 4) - (they don't get the time to execute)

   b. EDF:

      i. Task 1) min=4.000, avg=4.000, max=4.000

      ii. Task 2) min=2.000, avg=2.000, max=2.000

      iii. Task 3) - (they don't get the time to execute)

      iv. Task 4) - (they don't get the time to execute)

   Tasks 3 and 4 don't execute at all on both algorithms, but average response time for RM algorithm is slightly lower than EDF.

c. Preempted and Aborted:
    a. RM:
        i. Task 1) No preemptions nor abortions.
        ii. Task 2) Preempted or aborted at 4s, 8s, 12s, 16s, 20s, 24s, 28s, 32s, 36s, 40s, 44s, 48s
        iii. Task 3) Preempted or aborted at 3s, 7.5s, 11.5s, 16.5s, 19.5s, 23.5s, 27.5s, 31.5s, 35.5s, 39.5s, 43.5s, 47.5
        iv. Task4) Doesn't execute at all
    b. EDF:
        i. Task 1) No preemptions nor abortions.
        ii. Task2) Preempted or aborted at 4s, 8s, 12s, 16s, 20s, 24s, 28s, 32s, 36s, 40s, 44s, 48s
        iii. Task3) Preempted or aborted at 3s, 7.5s, 11.5s, 16.5s, 19.5s, 23.5s, 27.5s, 31.5s, 35.5s, 39.5s, 43.5s, 47.5
        iv. Task4) Doesn't execute at all

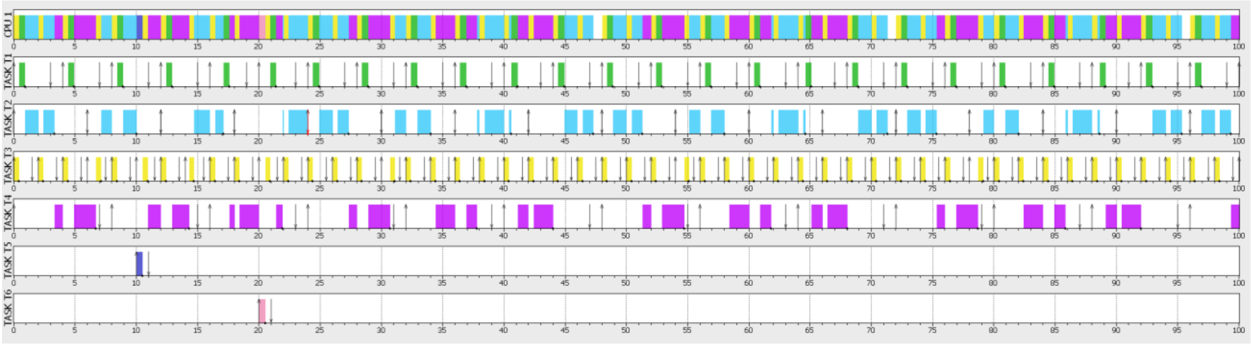
Both algorithms follow the same pattern of preemption and abortion.

d. Task 4 doesn't get executed on both algorithms, so the response time is None.

e. According to response times for task 1 and 2 on both algorithms, the average response time is lower on RM algorithm.

Question 4

a. The grant chart is as followed:



b. As followed:
    a. Task 1) 0s (no interventions)
    b. Task 2) 0s (no interventions)
    c. Task 3) 1s in total (twice, at 10 and 20s)
    d. Task 4) 0.5s in total (once, at 20s)

c. Yes, task 2 reaches deadline at 24s. We cannot since EDF is an optimal algorithm; This means if EDF fails, then all other algorithms will fail as well. No increasing the response time does not help in this case.

ERTS – HW1

Question 5

a. Since laptop itself is not considered an embedded system, then everything running on it is also **non-embedded**. Browsing is considered a **soft real-time** since it reduces use satisfaction on delay, but doesn’t interrupt the laptop operations.
b. YouTube platform is a **non-embedded system** and **firm real-time**, since it won’t be of any use if streaming speed or quality is no good.
c. Yes, the remote is an **embedded system**, since it’s got a dedicated hardware as well as a dedicated purpose to follow and is considered to be a <u>VERY</u> **hard real-time** system (:

Rewarded Question

یک سیستم real-time executive، لایه ای است که بین application و hardware & software قرار گرفته و به نوعی ماشین مجازی را برای استفاده آماده می‌کند که برنامه نویسی ساده تری دارد. همچنین این لایه جنبه های امنیتی بیشتری را ارائه می‌دهد. این سیستم را می‌توانیم مستقیم و یا بر روی یک سیستم عامل سوار کرده به طوری که نیاز حداقلی به سخت افزار دارد و می‌تواند peripheral ها را مدیریت کند. این سیستم قابلیت task scheduling و الگوریتم هایی ارائه میدهد تا cpu utility را حداکثر کند. هدف task scheduling بهینه سازی اجرای کار های مختلف توسط پردازنده است، پس عملا با یک مسءله بهینه سازی سر و کار داریم و لازم است الگوریتم مناسب این کار را انتخاب کنیم، زیرا هدف اصلی این الگوریتم به انجام رساندن تسک ها قبل ددلاین است که واژه “feasible” بهترین توصیف برای این کار است. یکی از رایج ترین روش های این کار در نرم افزار افزودن آنها به یک لیست پیوندی است به طوری که اولیت آنها توسط موقعیتشان در لیست توسط الگوریتم تعیین می‌شود.