

Unidade Curricular: Sistemas Embarcados (EMB22109).

Professor: Hugo Marcondes.

Aluno: Gustavo Vianna França.

Florianópolis, 1 de fevereiro de 2022.

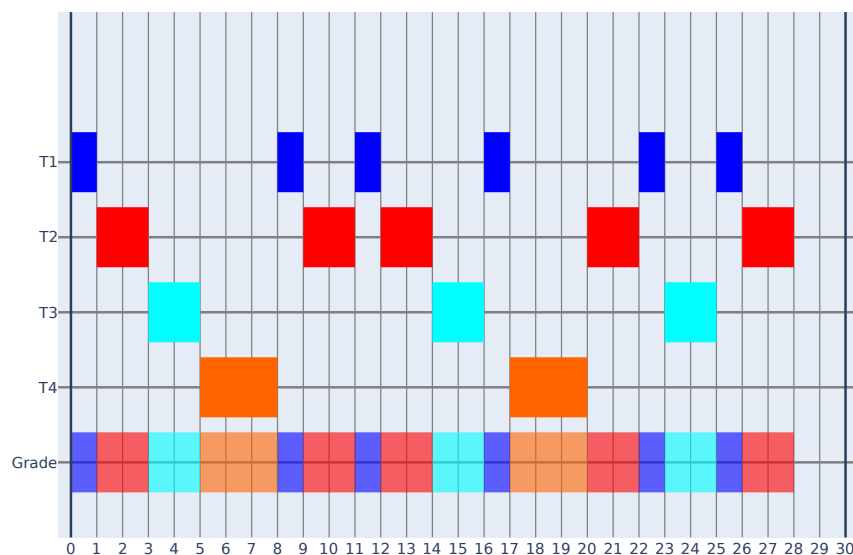
Resolução da Lista de Exercícios - Escalonamento em Tempo Real

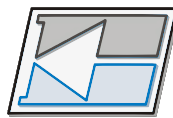
1. Considerando o conjunto de tarefas abaixo, realize o cálculo de utilização do conjunto de tarefas, e tente criar uma escala de tempo (executivo cíclico) para atender os deadlines solicitados. O conjunto de tarefas é escalonável por *Rate Monotonic*, com preempção e sem preempção?

	C	P	D
T1	1	5	5
T2	2	6	6
T3	2	10	10
T4	3	15	15

R: Inicialmente, a escala de tempo é realizada abaixo, que não necessita de preempção e cumpre todos os *deadlines*.

Escala de tempo (executivo cíclico) para o conjunto de tarefas do item 1





Em seguida para a análise de escalonabilidade por *Rate Monotonic*, como descrito por [Liu e Layland \(1973, p. 5\)](#), a utilização do conjunto de tarefas é dada pela soma da utilização do tempo de processamento de cada uma, de acordo com a equação abaixo:

$$U = \sum_{i=1}^N U_i = \sum_{i=1}^N \frac{C_i}{P_i} \quad (1)$$

Onde C_i , P_i , U_i e N são respectivamente a computação necessária, o período, a utilização de processamento de cada tarefa e o número total de tarefas e U é a utilização total de processamento das tarefas. Na tabela abaixo estão os valores percentuais de utilização de cada tarefa e do total:

	U
T1	20,0%
T2	33,33%
T3	20,0%
T4	20,0%
Total	93,33%

Assim, segundo [Liu e Layland \(1973, p. 8\)](#), um teste suficiente para averiguar a escalonabilidade do conjunto é apresentado a seguir:

$$\sum_{i=1}^N \frac{C_i}{P_i} \leq N \left(2^{\frac{1}{N}} - 1 \right) \quad (2)$$

Onde no conjunto atual de 4 tarefas tem um resultado de 0,7568, que é menor do que o valor de 0,9333, portanto não é possível garantir que ele seja escalonável. Desse modo, faz-se necessário a examinação dos piores cenários possíveis para a avaliar seus testes exatos.

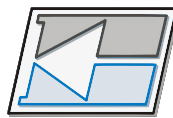
Primeiramente é verificado o cenário preemptivo. Assim, por meio das equações abaixo ([TINDELL; CLARK, 1994, p. 3](#)) são calculados os piores tempos de resposta do conjunto de tarefas, e caso eles ultrapassem os seus respectivos *deadlines*, o conjunto não é escalonável.

$$r_i = J_i + w_i \quad (3)$$

Onde:

$$w_i = C_i + B_i + \sum_{\forall j \in hp(i)} \left\lceil \frac{J_j + w_j}{P_j} \right\rceil C_j \quad (4)$$

Onde B_i , J e $hp(i)$ são, nessa ordem, o tempo de bloqueio que a tarefa i é bloqueada por uma tarefa de menor prioridade, o *release jitter* que é o atraso na liberação de uma tarefa e o último é a lista de tarefas de maior prioridade que a atual. Com isso se tem o pior tempo de resposta das tarefas, r_i , observado na tabela a seguir:



	r_i
T1	1
T2	3
T3	5
T4	17

Observa-se que a tarefa T4 possui um pior tempo de resposta maior que seu *deadline*, logo o conjunto não é escalonável.

Depois é considerado o cenário não preemptivo, que possui o mesmo resultado do teste suficiente para o caso preemptivo. Porém o seu pior tempo de resposta deve levar em conta as seguintes equações (GEORGE; RIVIERRE; SPURI, 1996, p. 29, 36 e 42):

$$L_i = \max_{j \in lp(i)} \{C_j - 1\} + \sum_{j \in hp(i) \cup \{i\}} \left\lceil \frac{L_i}{P_j} \right\rceil C_j \quad (5)$$

$$r_i = \max_{q=0, \dots, Q} \{w_{i,q} + C_i - qP_i\} \quad (6)$$

Onde:

$$w_{i,q} = qC_i + \sum_{j \in hp(i)} \left(1 + \left\lceil \frac{w_{i,q}}{P_j} \right\rceil\right) C_j + \max_{k \in lp(i)} \{C_k - 1\} \quad (7)$$

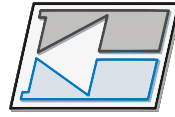
$$Q = \left\lceil \frac{L_i}{P_i} \right\rceil \quad (8)$$

Onde L_i , $w_{i,q}$, q , Q e $lp(i)$ são, de modo respectivo, o período de tempo de ocupação do processador por tarefas com maior ou mesma prioridade que a atual, o pior tempo de resposta da tarefa atual de determinada instância, as instâncias da tarefa atual, o número máximo de instâncias da tarefa atual e a lista de tarefas com prioridade menor que a atual. A partir dessas equações são obtidos os piores tempos de resposta possíveis para o conjunto, r_i , no que seus valores estão dispostos na tabela abaixo:

	r_i
T1	3
T2	5
T3	10
T4	11

Vê-se que nenhum pior tempo de resposta ultrapassa os *deadlines* de cada tarefa, consequentemente o conjunto é escalonável por *Rate Monotonic* sem preempção.

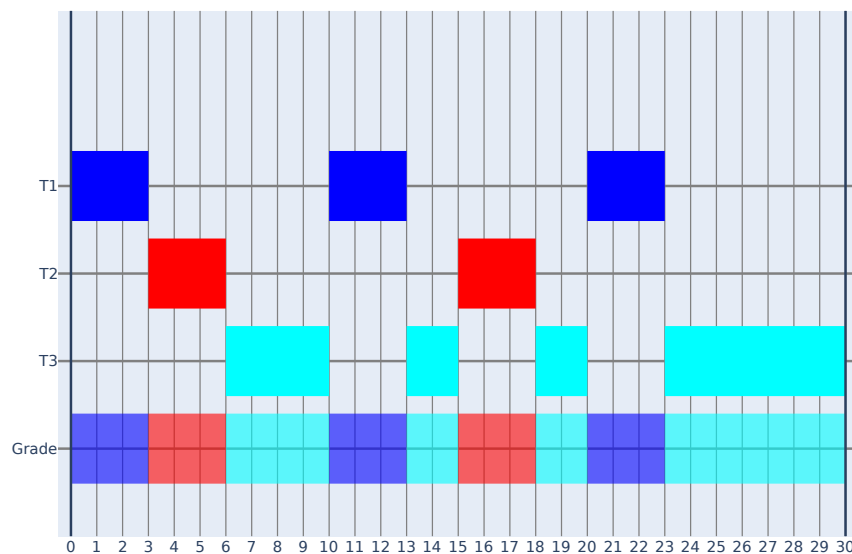
2. Considerando o conjunto de tarefas abaixo, tente criar uma escala de tempo (executivo cíclico) para atender os deadlines solicitados.



	C	P	D
T1	3	10	10
T2	3	15	15
T3	15	30	30

R: A escala de tempo requisitada é apresentada em sequência:

Escala de tempo (executivo cíclico) para o conjunto de tarefas do item 2



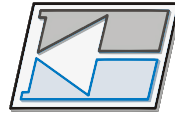
Nota-se que todos os *deadlines* foram obedecidos, porém é necessário o uso de preempção.

3. Em relação a escalonabilidade do conjunto de tarefas do exercício 2, pode-se dizer que:

- a) É impossível.
- b) **É possível, desde que algumas preempções sejam permitidas.**
- c) É possível, desde que nenhuma preempção seja permitida.
- d) É possível, dado que os períodos são múltiplos entre si.

R: O item b) é a resposta por ser necessário preempções sobre a 3ª tarefa e pelo fato de que nem todos os períodos são múltiplos entre si, e.g. as 1ª e 2ª tarefas.

4. No que diz respeito ao escalonamento de sistemas de tempo real, cite uma vantagem e uma desvantagem do executivo cíclico sobre o escalonamento com prioridades e teste de escalonabilidade.



R: Uma vantagem do escalonamento executivo cíclico sobre o escalonamento com prioridades e teste de escalonabilidade é o conhecimento de toda a escala de execução proporcionando determinismo. E uma desvantagem deste é a complexidade encontrada para escalonar estruturas e sistemas compostos de várias tarefas.

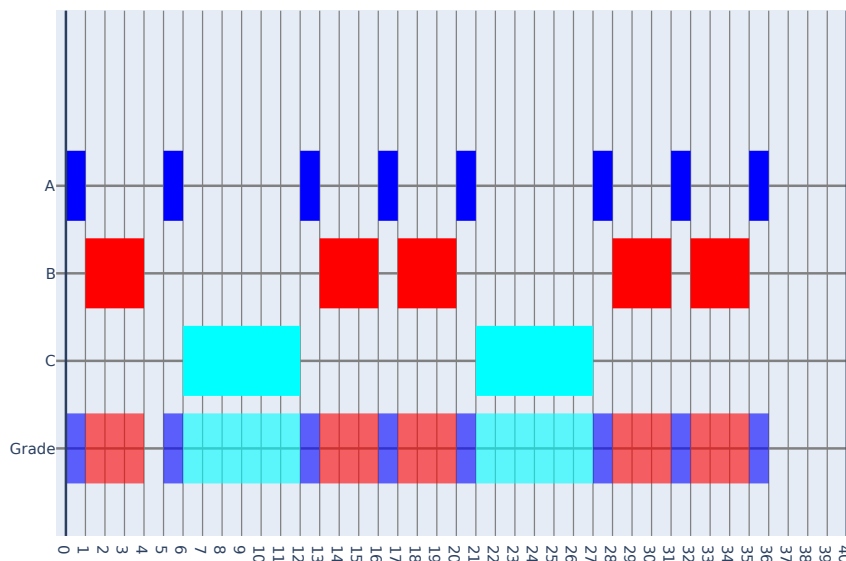
5. A tabela abaixo descreve um sistema a ser escalonado por um executivo cíclico.

	C	P	D
A	1	5	5
B	3	8	8
C	6	20	20

a) É possível resolver com uma grade de tempo não preemptiva? Justifique.

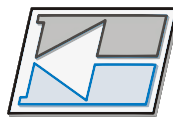
R: É factível que a grade possa ser não preemptiva, segue um exemplo na escala de tempo abaixo:

Escala de tempo (executivo cíclico) para o conjunto de tarefas do item 5.a)

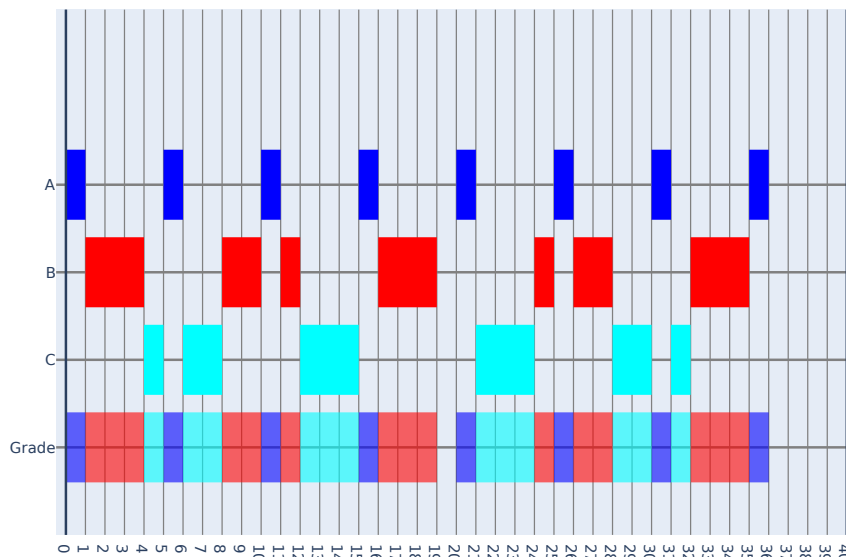


b) Construa uma grade de tempo que na verdade implementa um escalonamento EDF preemptivo.

R: Prossegue-se à tabela com o cumprimento dos critérios de escalonamento por EDF preemptivo:



Escala de tempo (EDF preemptivo) para o conjunto de tarefas do item 5.b)



- c) Caso a tarefa A seja esporádica, ainda assim é possível escalonar este sistema? Justifique a sua resposta.

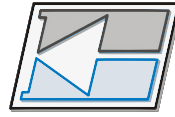
R: Quando a tarefa A é esporádica o pior cenário levantado é do seu intervalo mínimo entre ativações, fazendo com que possa ser considerada periódica. Logo, se a preempção for permitida, o teste exato para o escalonamento por EDF é válido, o qual é (GEORGE; RIVIERRE; SPURI, 1996, p. 8):

$$\sum_{i=1}^N \frac{C_i}{P_i} \leq 1 \quad (9)$$

Nisso o total de utilização de processamento é obtido a partir da tabela abaixo, no que seu valor é de 0,875, que é menor que 1, o que indica a sua escalonabilidade.

	U
A	20,0%
B	37,5%
C	30,0%
Total	87,5%

6. Considerando o conjunto de tarefas abaixo, tente criar uma escala de tempo (executivo cíclico) para atender os deadlines solicitados. As tarefas T1 e T2 compartilham uma mesma

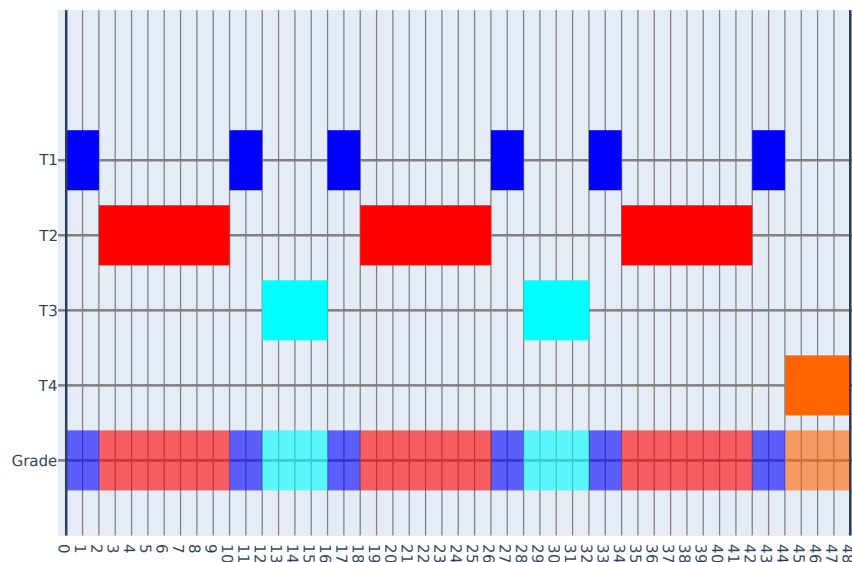


tabela, enquanto as tarefas T3 e T4 compartilham uma mesma lista encadeada. Procure minimizar os *overheads*.

	C	P	D
T1	2	8	8
T2	8	16	16
T3	4	24	24
T4	4	48	48

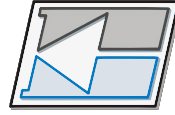
R: Dá-se continuidade à grade de tempo por meio de um escalonamento não preemptivo visando diminuir os *overheads*.

Escala de tempo (executivo cíclico) para o conjunto de tarefas do item 6



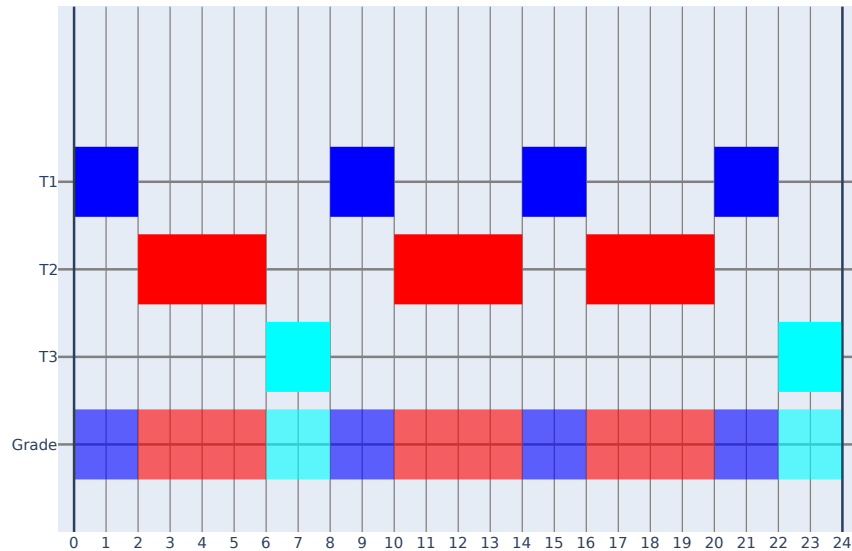
7. Considerando o conjunto de tarefas abaixo, tente criar uma escala de tempo (executivo cíclico) para atender os deadlines solicitados. As tarefas T1 e T2 compartilham a mesma estrutura de dados.

	C	P	D
T1	2	6	6
T2	4	8	8
T3	2	12	12



R: Apresenta-se abaixo a tabela com um escalonamento não preemptivo:

Escala de tempo (executivo cíclico) para o conjunto de tarefas do item 7



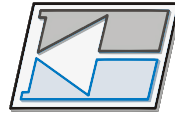
8. Considerando a tabela abaixo, usando a política de atribuição de prioridades *Deadline Monotonic*:

	C	P	D
A	4	20	10
B	3	30	15
C	11	40	30

Calcule os tempos de resposta e mostre a escalonabilidade (ou não escalonabilidade) desse conjunto de tarefas. Interrupções podem ficar desabilitadas por no máximo 1 u.t. e as tarefas A e B compartilham uma estrutura de dados protegida por mutex cuja seção crítica demora 2 u.t. para ser executada.

R: Toma-se conhecimento que o teste suficiente de escalonabilidade do *Deadline Monotonic* é similar ao do *Rate Monotonic* mostrado na [Equação 2](#). No entanto, quando os *deadlines* são menores que os períodos ela é modificada para:

$$\sum_{i=1}^N \frac{C_i}{D_i} \leq N \left(2^{\frac{1}{N}} - 1 \right) \quad (10)$$



Visto que na tabela em sequência o valor total do fator de utilização do processador é 0,9667, um valor maior que 0,7798, não se pode afirmar que o conjunto é escalonável.

Em virtude disso, faz-se necessária a análise do tempo de resposta. Desse modo são calculados os piores tempos de resposta para o caso preemptivo, que utilizam a [Equação 3](#) e a [Equação 4](#) considerando o *release jitter* como o tempo máximo em que as interrupções são desabilitadas e o tempo de bloqueio como a demora da seção crítica para as duas primeiras tarefas. Ao final os seus resultados estão dispostos na tabela abaixo:

	r_i
A	7
B	10
C	19

Observa-se que os tempos são todos menores que os *deadlines* das suas respectivas tarefas, e portanto é escalonável. Já para o cenário não preemptivo utilizam-se a [Equação 5](#), a [Equação 6](#), a [Equação 7](#) e a [Equação 8](#). Seus piores tempos estão delineados na tabela abaixo:

	r_i
A	14
B	17
C	18

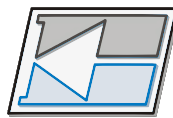
Com isso, verificam-se que as duas primeiras tarefas ultrapassam os seus respectivos *deadlines*, logo elas não são escalonáveis por *Deadline Monotonic* não preemptivo.

9. Considerando a tabela abaixo, usando a política de atribuição de prioridades *Deadline Monotonic*:

	C	P	D
T1	2	10	10
T2	3	30	30
T3	9	40	30

Calcule os tempos de resposta e mostre a escalonabilidade (ou não escalonabilidade) desse conjunto de tarefas. Interrupções podem ficar desabilitadas por no máximo 2 u.t. e as tarefas T2 e T3 compartilham uma estrutura de dados protegida por mutex cuja seção crítica demora 3 u.t. para ser executada. A T1 é esporádica. Interrupções do *timer* acontecem a cada 10 u.t. para ativar um tratador que demora no máximo 1 u.t. para executar.

R: De forma similar à questão anterior, é realizado o teste suficiente, porém incluindo uma tarefa adicional, T0, de período de 10 u.t., *deadline* de 3 u.t. (devido ao *release jitter*) e



tempo de computação igual a 1 u.t., com o intuito de representar o tratador de interrupções do *timer*. Também como discutido em uma questão prévia, o pior tempo de resposta para uma tarefa esporádica, no caso T1, é a situação em que ela ocorre periodicamente.

Além disso, o valor de comparação do seu teste é modificado para 1, como na [Equação 9](#), devido ao fato de que o conjunto possui apenas períodos múltiplos entre si (conjunto de tarefas harmônicas). Essas contribuições resultam em um valor de utilização de processamento total de 0,9333, que é menor do que 1, e por consequência o conjunto é escalonável.

E os tempos de resposta para essa política no caso preemptivo estão na tabela abaixo:

	r_i
T0	3
T1	5
T2	14
T3	26

Denota-se que nenhuma tarefa ultrapassou o seu respectivo *deadline*, fazendo com o que o conjunto seja escalonável com preempção. E por fim, os piores tempos de resposta sem preempção estão localizados na tabela abaixo:

	r_i
T0	9
T1	11
T2	17
T3	15

A atraso do tratador de interrupções do *timer* não é atendido e nem o *deadline* da tarefa T1, o que torna o conjunto não escalonável por *Deadline Monotonic* sem preempção.

Referências

GEORGE, L.; RIVIERRE, N.; SPURI, M. Preemptive and non-preemptive real-time uniprocessor scheduling. 01 1996. Disponível em: <<https://hal.inria.fr/inria-00073732/PDF/RR-2966.pdf>>. Acesso em: 26 jan. 2022. Citado 2 vezes nas páginas 3 e 6.

LIU, C.; LAYLAND, J. **Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment**. 1973. Disponível em: <<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.36.8216>>. Acesso em: 26 jan. 2022. Citado na página 2.

TINDELL, K.; CLARK, J. A. Holistic schedulability analysis for distributed hard real-time systems. **Microprocess. Microprogramming**, v. 40, p. 117–134, 1994. Disponível em: <<http://beru.univ-brest.fr/cheddar/publications/tindell94.pdf>>. Acesso em: 26 jan. 2022. Citado na página 2.