



FUNDAÇÃO EDSON QUEIROZ  
UNIVERSIDADE DE FORTALEZA  
ENSINANDO E APRENDENDO

## T569 –SISTEMAS DE TEMPO REAL

---

# Aula 11

Prof. Marcelo Sousa



## Agenda

- Event Driven Scheduler
  - Earliest Dead Line First (EDF)
    - Primeiro Deadline Primeiro
  - Rate Monotonic Algorithm (RMA)
    - Algoritmo de Taxa Monotônica



## Earliest Deadline First (EDF)

- A cada ponto de escalonamento de tarefas, a tarefa com deadline mais próximo é posta em execução
- O teste de escalonabilidade:  $\sum_{i=1}^n \frac{e_i}{p_i} = \sum_{i=1}^n u_i \leq 1$ 
  - $p_i = d_i$
- Considerado um escalonador ótimo para sistemas uniprocessados



## Earliest Deadline First (EDF)

- Se  $p_i > d_i$ 
  - Logo o tempo de computação é realizado a cada:
    - $\min(p_i, d_i)$
- Então: 
$$\sum_{i=1}^n \frac{e_i}{\min(p_i, d_i)} \leq 1$$
- Se  $p_i < d_i$  :
  - É possível que mesmo com o teste acima falso o conjunto de tasks ser escalonável.



## Earliest Deadline First (EDF)

- Exemplo:
  - Considere o conjunto de três tasks periódicas a serem escalonadas em um escalonador de tempo real.  
 $T_1=(e_1=10, p_1=20)$  ;  $T_2=(e_2=5, p_2=50)$  ;  
 $T_3=(e_3=10, p_3=35)$ . Determine se o sistema é escalonável.



## Earliest Deadline First (EDF)

- É realmente um algoritmos de prioridades dinâmicas?
  - Para considerar isso, é necessário que o escalonador tenha conhecimento o tempo todo de qual o valor da prioridade de cada task.
  - E o EDF não há a necessidade de conhecimento de nenhuma prioridade de task.



## Earliest Deadline First (EDF)

- Implementação do EDF
  - É necessário colocar em uma fila todas as tasks prontas para execução:
  - Caso alguma task nova esteja pronta pra execução (ready) ela é colocada no final da fila.
  - Cada task deve conter o valor absoluto do *deadline*
  - A cada ponto de escalonamento todas as tasks são analisadas para avaliar qual possui o menor *deadline*
  - *Nota: Implementação ineficiente*
    - *Necessidade de varrer a fila sempre*



## Earliest Deadline First (EDF)

- Implementação eficiente do EDF
  - Deve manter todas as tasks ordenadas em uma fila de prioridades de execução, que é mantida sempre ordenada.
  - Cada vez que uma task é inserida, ela é inserida na posição adequada.
  - A cada ponto de escalonamento, a próxima task a ser executada se encontra na primeira posição da fila.





## Rate Monotonic Algorithm (RMA)

- Algoritmo estático de prioridades
- Assinala uma prioridade para a task baseada na sua taxa de ocorrências.
- Quanto menor taxa de ocorrência de uma task, menor será sua prioridade
- É provado com uma algoritmo ótimo considerando se escalonadores de prioridades estáticas



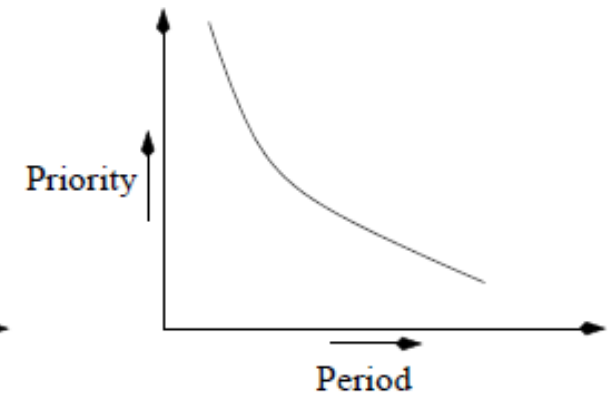
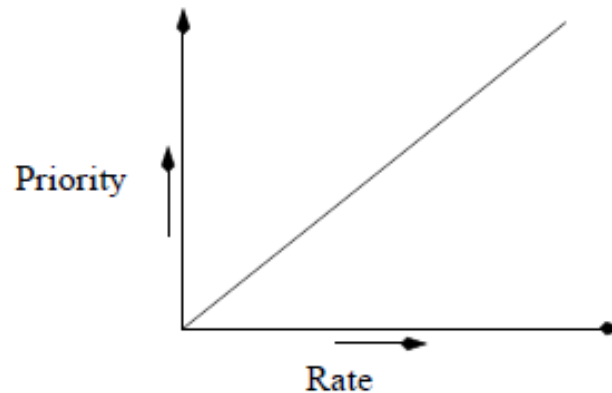
## Rate Monotonic Algorithm (RMA)

- A prioridade de uma task é diretamente proporcional a sua taxa, ou inversamente proporcional ao seu período:

$$priority = \frac{k}{p_i}$$

$p_i = \text{período}$

$k = \text{constante}$





# Rate Monotonic Algorithm (RMA)

- Teste de Escalonabilidade p/ RMA:

- Condições:
  - Necessária:  $\sum_{i=1}^n \frac{e_i}{p_i} = \sum_{i=1}^n u_i \leq 1$
  - Suficiente:  $\sum_{i=1}^n u_i \leq n \left( 2^{\frac{1}{n}} - 1 \right)$

- Condição de Liu e Layland



# Rate Monotonic Algorithm (RMA)

- Considerando o número de tasks do sistema

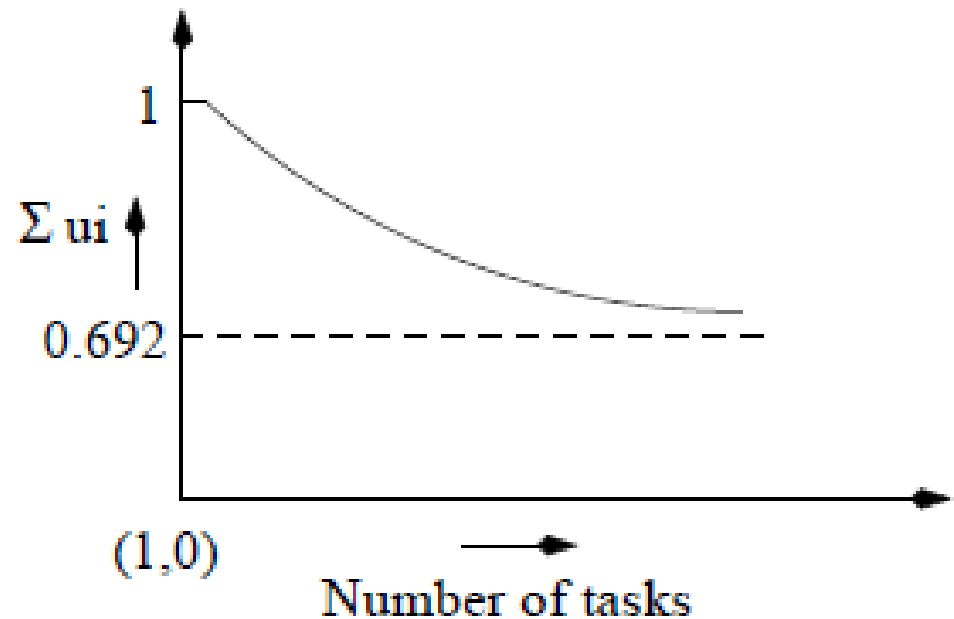
$$n = 1 \Rightarrow \sum_{i=1}^n u_i \leq 1$$

$$n = 2 \Rightarrow \sum_{i=1}^n u_i \leq 0,824$$

$$n = 3 \Rightarrow \sum_{i=1}^n u_i \leq 0,78$$

...

$$n = 3 \Rightarrow \sum_{i=1}^n u_i \leq \infty.0 \Rightarrow 0,692$$





## Rate Monotonic Algorithm (RMA)

- Exemplo: Verifique se o conjunto de tasks a seguir é escalonável sob o RMA em um único processador:
  - $T_1(e_1=20, p_1=100)$ ;
  - $T_2(e_2=30, p_2=150)$ ;
  - $T_3(e_3=60, p_3=200)$ ;



## Rate Monotonic Algorithm (RMA)

- Exemplo: Verifique se o conjunto de tasks a seguir é escalonável sob o RMA em um único processador:
  - $T_1(e_1=20, p_1=100)$ ;
  - $T_2(e_2=30, p_2=150)$ ;
  - $T_3(e_3=90, p_3=200)$ ;



## Rate Monotonic Algorithm (RMA)

- Se um conjunto de task passar no teste de Liu Layard é garantido que é escalonável através de RMA.
- Por outro lado, se mesmo não passando, a task pode ser escalonável através de RMA.
- É necessário realizar o teste de Lehoczky



# Rate Monotonic Algorithm (RMA)

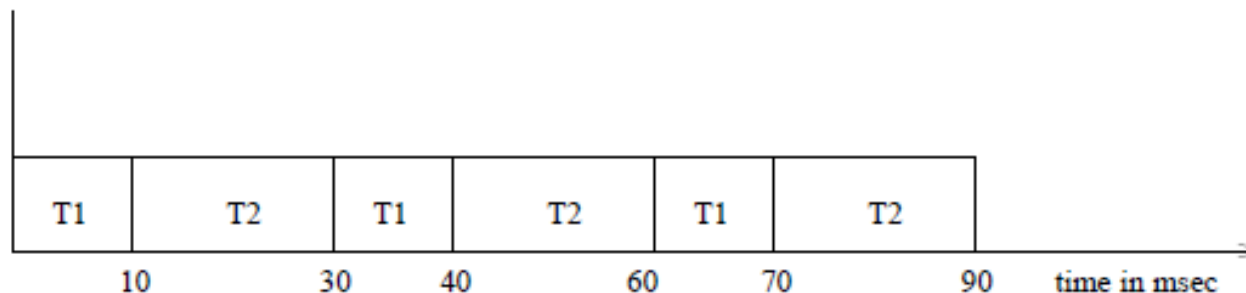
- Teste de Lehoczky
  - Um conjunto de tarefas periódicas de tempo real é escalonável por RMA se todas as tarefas alcançarem seus respectivos deadlines quando possuírem fase 0.



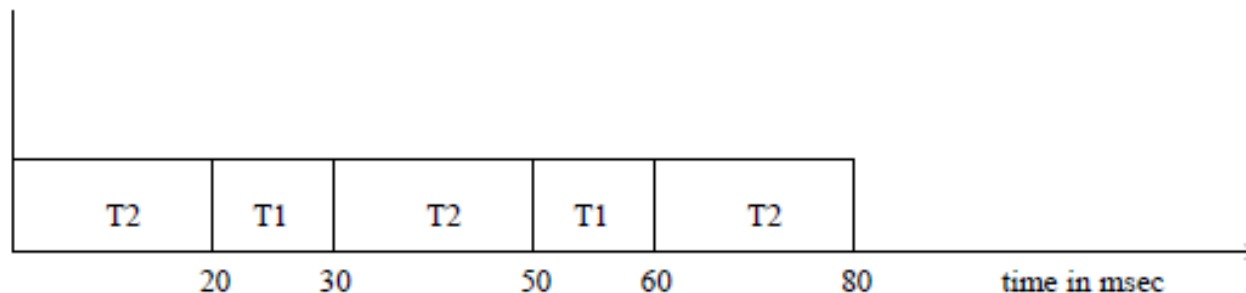


## Rate Monotonic Algorithm (RMA)

- Exemplo:
  - $T1=(10,30)$  – Maior Prioridade
  - $T2=(60,120)$  – Menor Prioridade



(a) T1 is in phase with T2



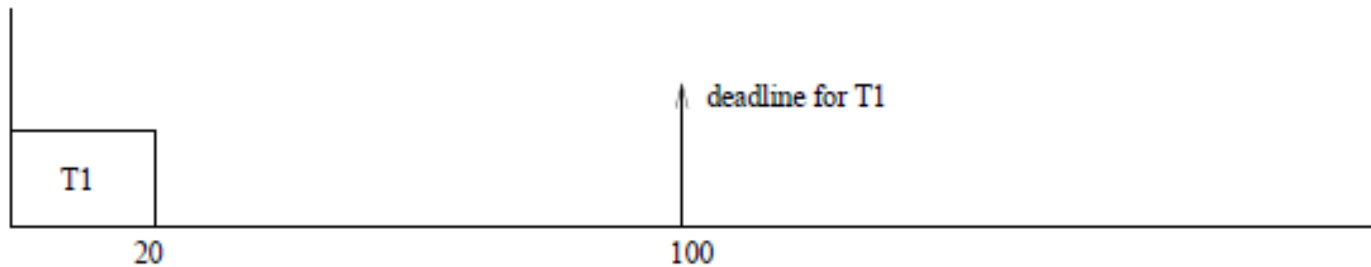


## Rate Monotonic Algorithm (RMA)

- Exemplo: Verifique se exemplo anterior é escalonável por RMA através do teorema de Lehoczky :
  - $T_1=(e_1=20,p_1=100);$
  - $T_2(e_2=30,p_2=150);$
  - $T_3(e_3=90,p_3=200);$



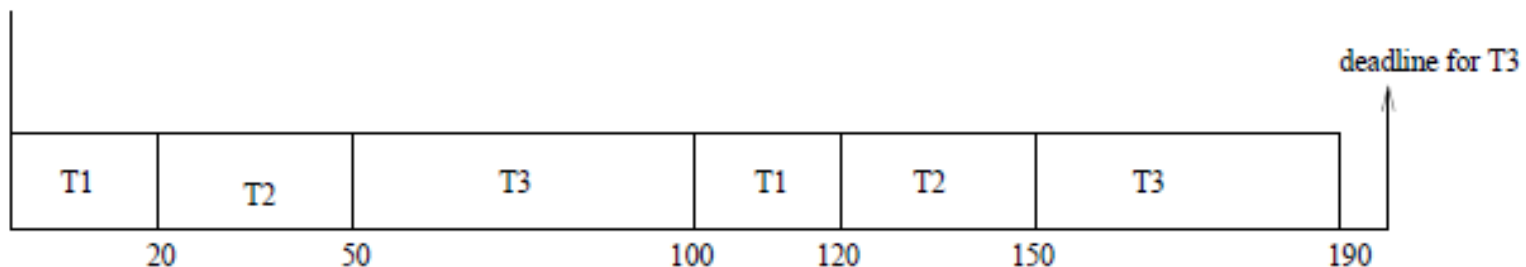
## Rate Monotonic Algorithm (RMA)



(a) T1 meets its first deadline



(b) T2 meets its first deadline



(c) T3 meets its first deadline



## Rate Monotonic Algorithm (RMA)

- Exemplo: Considerando o conjunto de três tasks periódicas de tempo real
  - $T_1=(10,20)$ ;
  - $T_2=(15,60)$ ;
  - $T_3=(20,120)$ ;