

# Algoritmos de Escalonamento II

Eduardo Ferreira dos Santos

Ciência da Computação  
Centro Universitário de Brasília – UniCEUB

Março, 2016

## Sumário

- 1 Restrições temporais
- 2 Escalonamento Taxa Monotônica
- 3 Escalonamento EDF

- 1 Restrições temporais
- 2 Escalonamento Taxa Monotônica
- 3 Escalonamento EDF



# Definição de tempo

- As tarefas a serem escalonadas em tempo real serão descritas através da tripla  $J_i, P_i, C_i, D_i$

$J_i$  *Release Jitter*, ou tempo de ativação no **pior caso**;

$P_i$  Período, ou duração da tarefa;

$C_i$  Tempo de Computação (execução) da tarefa;

$D_i$  *Deadline*, ou tempo **máximo** que a tarefa pode durar.

## Tarefas de Tempo Real I

- Legenda
  - Tempo de Chegada
  - Tempo de Liberação
  - Tempo de Início
  - Tempo de Término

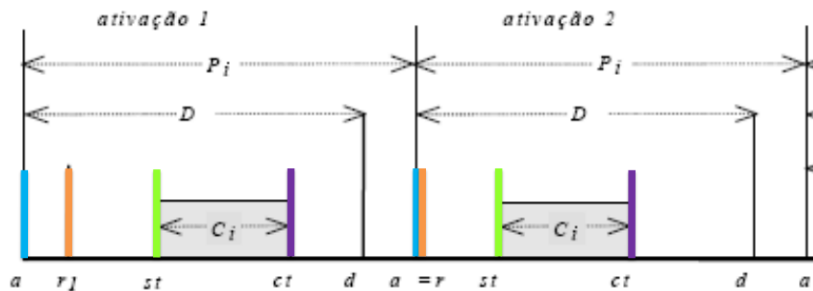


Figura 1.2: Ciclo de Tarefas 1 [Chagas, 2016]

## Tarefas de Tempo Real II

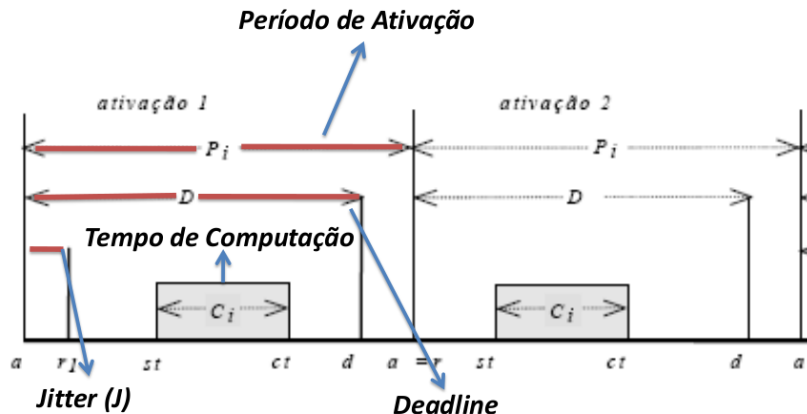


Figura 1.3: Ciclo de Tarefas 2 [Chagas, 2016]

- 1 Restrições temporais
- 2 Escalonamento Taxa Monotônica
- 3 Escalonamento EDF



## Premissas do RM [FARINES and MELO, 2000]

- 1 As tarefas são **periódicas** e **independentes**;
- 2 O *deadline* de cada tarefa coincide com seu período ( $D_i = P_i$ );
- 3 O tempo de computação ( $C_i$ ) de cada tarefa é conhecido e constante (*Worst computation time*);
- 4 O tempo de chaveamento entre as tarefas é considerado **nulo**.

Qual o problema das premissas 1 e 2?

# Atribuição de prioridades no RM

- Ordenação baseada nos valores dos períodos:
  - As prioridades decrescem em função do aumento dos períodos;
  - Tarefas mais frequentes têm maior prioridade.
- Condição suficiente para o atendimento das prioridades de  $n$  tarefas:

$$U = \sum_i^n \frac{C_i}{P_i} \leq n(2^{\frac{1}{n}} - 1) \quad (1)$$

Equação 1: teste de condição suficiente para atendimento das tarefas, onde  $U$  é a utilização de CPU.

# Teste de escalonamento RM

- O limite teórico da utilização de CPU é 0,69.
- Quando o período das tarefas coincide com um múltiplo da tarefa mais prioritária, podemos reduzir o teste à uma condição **necessária e suficiente**.

$$U = \sum_i^n \frac{C_i}{P_i} \leq 1 \quad (2)$$

## Exemplo 01

Tarefa	$P_i$	$D_i$	$C_i$
A	100	100	20
B	150	150	40
C	350	350	100

Tabela 2.1: Restrições de tempo do Exemplo 01

## Prioridades

Tarefa	$P_i$	$C_i$	$p_i$	$U_i$
A	100	20	1	0,2
B	150	40	2	0,267
C	350	100	3	0,286

Tabela 2.2: Cálculo de prioridades do Exemplo 01

## Escala RM

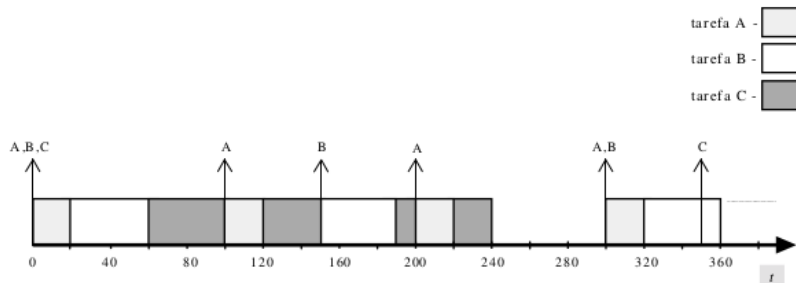


Figura 2.1: Escala RM do Exemplo 01 [FARINES and MELO, 2000]

## Exemplo 02

Tarefa	$P_i$	$D_i$	$C_i$
A	20	20	10
B	50	50	25

Tabela 2.3: Restrições de tempo do Exemplo 02

## Prioridades Exemplo 02

Tarefa	$P_i$	$C_i$	$p_i$	$U_i$
A	20	10	1	0,5
B	50	25	2	0,5

Tabela 2.4: Cálculo de prioridades do Exemplo 02



- 1 Restrições temporais
- 2 Escalonamento Taxa Monotônica
- 3 Escalonamento EDF

# Definição

- Escalonamento *Earliest Deadline First*: baseado em prioridades;
  - 1 Escalonamento de prioridade dinâmica;
  - 2 Escalonamento *on-line*.
- Considerado um algoritmo **ótimo** nos algoritmos de prioridade dinâmica.

# Premissas do EDF [FARINES and MELO, 2000]

- 1 As tarefas são **periódicas** e **independentes**;
- 2 O *deadline* de cada tarefa coincide com seu período ( $D_i = P_i$ );
- 3 O tempo de computação ( $C_i$ ) de cada tarefa é conhecido e constante (*Worst computation time*);
- 4 O tempo de chaveamento entre as tarefas é considerado **nulo**.

# Atribuição de prioridades no EDF

- Atribuição **dinâmica** de prioridades com base no *deadline* absoluto:
  - 1 A tarefa mais prioritária é a que tem o *deadline* ( $d_i$ ) mais próximo do tempo atual;
  - 2 Quando chega uma tarefa, a fila de pronto é reordenada com base na nova distribuição de prioridades;
  - 3 A cada ativação de uma tarefa, um novo *deadline* absoluto é determinado considerando o **número de períodos** que antecede a atual ativação ( $k$ )

$$d_{ik} = kP_i \quad (3)$$

## Voltando ao exemplo 02

Tarefa	$P_i$	$D_i$	$C_i$
A	20	20	10
B	50	50	25

Tabela 3.1: Restrições de tempo do Exemplo 02

## Escalonamento EDF x RM

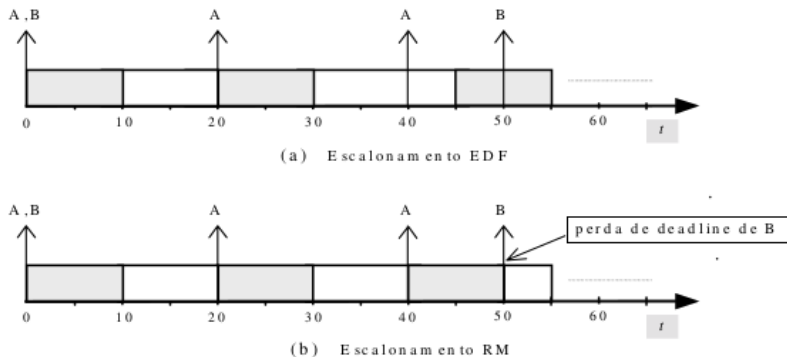


Figura 2.6: Escalas produzidas pelo (a) EDF e (b) RM



Figura 3.1: Comparação do escalonamento EDF com o RM  
[FARINES and MELO, 2000]

## Comparação EDF x RM [FARINES and MELO, 2000]

*“A escalonabilidade é também verificada em tempo de projeto, tomando como base a utilização do processador” (EDF).*

*“O escalonamento Taxa Monotônica ('Rate Monotonic') produz escalas em tempo de execução através de escalonadores preemptivos, dirigidos a prioridades.”*

- O EDF produz uma melhor utilização de CPU para o exemplo analisado;
- EDF produz menos preempções que RM;
- O RM é mais simples de implementar.

-  Chagas, F. (2016).  
Notas de aula do Prof. Fernando Chagas.
-  FARINES, J. M. and MELO, R. (2000).  
*Sistemas de Tempo Real*, volume 1.  
IME-USP.



OBRIGADO!!!  
PERGUNTAS???