

**Sistemas de Tempo Real:
Análise de Escalonabilidade**

Jean-Marie Farines
Joni da Silva Fraga
Rômulo Silva de Oliveira

LCMI - Laboratório de Controle e Microinformática
DAS - Departamento de Automação e Sistemas
UFSC - Universidade Federal de Santa Catarina

Escalonamento Baseado em Prioridades Fixas

- Aplicação composta por tarefas (processos)
- Estados de uma tarefa:
 - Liberada (pronta para executar)
 - Suspensa esperando por evento ou passagem do tempo
 - Bloqueada
- Em geral escalonamento é preemptivo
- Tarefas possuem prioridade fixa definida em projeto
- Garantia exige
 - Tarefas periódicas ou esporádicas
 - Tempo máximo de computação conhecido
 - Teste de escalonabilidade apropriado

Teste de Escalonabilidade

- Teste de Escalonabilidade pode ser
 - Suficiente
 - Exato
 - Necessário

	Teste Necessário	Teste Exato	Teste Suficiente
– 1	OK	OK	OK
– 2	OK	OK	OK
– 3	OK	OK	NÃO
– 4	OK	NÃO	NÃO
– 5	OK	NÃO	NÃO
– 6	NÃO	NÃO	NÃO

Rate Monotonic

- Quanto menor o período, maior a prioridade
- Ótimo quando
 - Tarefas são periódicas
 - Deadline é sempre igual ao período
- Exemplo:

– Tarefas	T1	T2	T3
– Períodos	P1=30	P2=40	P3=50
– Prioridades	p1=1	p2=2	p3=3
- Cuidado!
 - Número menor indica prioridade maior
 - Muitas vezes é o contrário

Análise da Utilização

- Utilização de uma tarefa:
 - Tempo máximo de computação dividido pelo período
 - T1 tem $C1=12$ e $P1=50$, então $U1 = 12 / 50 = 0.24$
- Teste para Rate Monotonic, sistema é escalonável se:

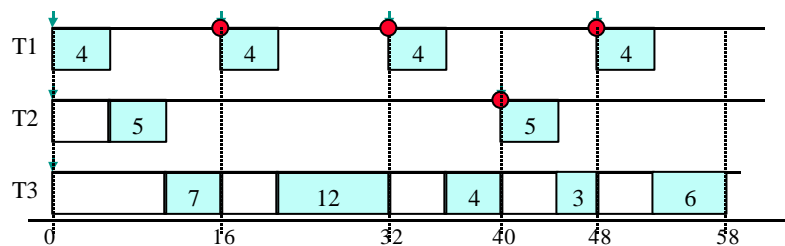
$$\sum_{i=1}^N \frac{C_i}{P_i} \leq N(2^{1/N} - 1)$$

- Para $N=1$ utilização máxima é 100%
- Para $N=10$ utilização máxima é 71.8%
- Para N grandes utilização máxima tende para 69.3%
- Baseado no conceito de Instante Crítico
- Teste é suficiente mas não necessário

Análise da Utilização



- Exemplo:

	T1	T2	T3
– Períodos	$P1=16$	$P2=40$	$P3=80$
– Computação	$C1=4$	$C2=5$	$C3=32$
– Utilização	$U1=0.250$	$U2=0.125$	$U3=0.400$
– Prioridades	$p1=1$	$p2=2$	$p3=3$
- Utilização total é 0.775, abaixo do limite 0.780

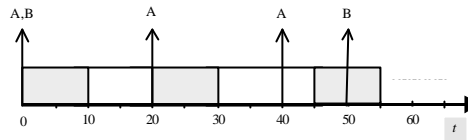


EDF

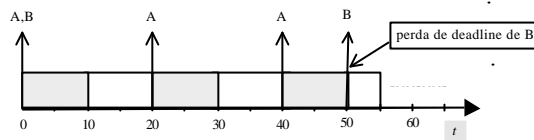
tarefas periódicas	C_i	P_i	D_i
tarefa A	10	20	20
tarefa B	25	50	50

tarefa A - 
tarefa B - 

$$U = \sum_{i=1}^n C_i / P_i \leq 1. \quad [3]$$



(a) Escalonamento EDF



(b) Escalonamento RM

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

Sistemas de Tempo Real - Jean-Marie Farines, Joni da Silva Fraga, Rômulo Silva de Oliveira

7

Análise do Tempo de Resposta

- Limitações da análise baseada em Utilização
 - Não é exata
 - Aplicável apenas a modelos de tarefas muito simples
- Análise baseada em **Tempo de Resposta**
 - Abordagem analítica calcula tempo de resposta no pior caso
 - Tempo de resposta de cada tarefa é comparada com o deadline
- Para a tarefa mais prioritária temos $R1 = C1$
- Demais tarefas sofrem **Interferência** das tarefas com prioridade maior
- Neste caso, $R_i = C_i + I_i$
- Interferência é máxima a partir do **Instante Crítico**
 - Todas as tarefas são liberadas simultaneamente
 - Suposto instante zero na análise

Sistemas de Tempo Real - Jean-Marie Farines, Joni da Silva Fraga, Rômulo Silva de Oliveira

8

Análise do Tempo de Resposta

- Seja T_j uma tarefa com prioridade maior que T_i
- Quantas vezes T_j pode acontecer durante a execução de T_i ?

$$\left\lceil \frac{R_i}{P_j} \right\rceil$$

- Qual a interferência total de T_j sobre T_i ?

$$\left\lceil \frac{R_i}{P_j} \right\rceil \times C_j$$

- Qual a interferência total sobre T_i ?

$$\sum_{j \in hp(i)} \left\lceil \frac{R_i}{P_j} \right\rceil \times C_j$$

Análise do Tempo de Resposta

- O tempo máximo de resposta de T_i é $R_i = C_i + I_i$

$$R_i = C_i + \sum_{j \in hp(i)} \left\lceil \frac{R_i}{P_j} \right\rceil \times C_j$$

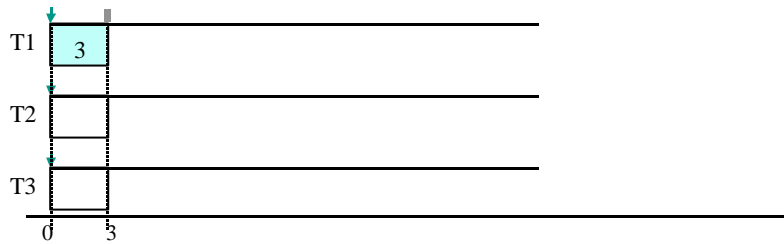
- Equação é recursiva
- Calculada através de iterações sucessivas, até:
 - Tempo de resposta passar do deadline
 - Resultado convergir, iteração $x+1$ igual a iteração x

$$w_i^{x+1} = C_i + \sum_{j \in hp(i)} \left\lceil \frac{w_i^x}{P_j} \right\rceil \times C_j \quad w_i^0 = C_i$$

Análise do Tempo de Resposta

- Exemplo:

	T1	T2	T3
– Períodos	P1=7	P2=12	P3=20
– Computação	C1=3	C2=3	C3=5
– Prioridades	p1=1	p2=2	p3=3
- R1 = C1 = 3

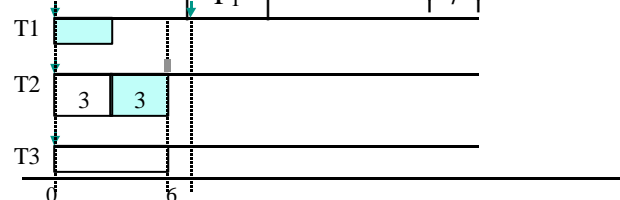


Análise do Tempo de Resposta

- Análise da tarefa T2: $w_2^0 = C_2 = 3$

$$w_2^1 = C_2 + \left\lceil \frac{w_2^0}{P_1} \right\rceil \times C_1 = 3 + \left\lceil \frac{3}{7} \right\rceil \times 3 = 6$$

$$w_2^2 = C_2 + \left\lceil \frac{w_2^1}{P_1} \right\rceil \times C_1 = 3 + \left\lceil \frac{6}{7} \right\rceil \times 3 = 6$$



Análise do Tempo de Resposta

- Análise da Tarefa T3:

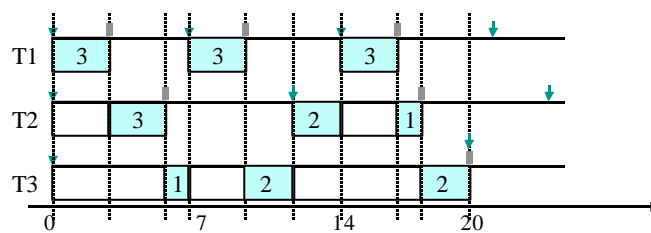
$$w_3^0 = C_3 = 5$$

$$w_3^1 = C_3 + \sum_{j \in hp(3)} \left\lceil \frac{w_3^0}{P_j} \right\rceil \times C_j = 5 + \left\lceil \frac{5}{7} \right\rceil \times 3 + \left\lceil \frac{5}{12} \right\rceil \times 3 = 11$$

$$w_3^2 = 5 + \left\lceil \frac{11}{7} \right\rceil \times 3 + \left\lceil \frac{11}{12} \right\rceil \times 3 = 14 \quad w_3^3 = 5 + \left\lceil \frac{14}{7} \right\rceil \times 3 + \left\lceil \frac{14}{12} \right\rceil \times 3 = 17$$

$$w_3^4 = 5 + \left\lceil \frac{17}{7} \right\rceil \times 3 + \left\lceil \frac{17}{12} \right\rceil \times 3 = 20 \quad w_3^5 = 5 + \left\lceil \frac{20}{7} \right\rceil \times 3 + \left\lceil \frac{20}{12} \right\rceil \times 3 = 20$$

Análise do Tempo de Resposta



- Exemplo:

	T1	T2	T3
- Períodos	P1=7	P2=12	P3=20
- Computação	C1=3	C2=3	C3=5
- Prioridades	p1=1	p2=2	p3=3
- Tempo Máximo de Resposta	R1=3	R2=6	R3=20

Análise do Tempo de Resposta

- Teste de escalonabilidade **exato**
- Deadline pode ser menor que o período
 - Basta comparar o tempo de resposta com o deadline
- Deadline maior que o período exige análise mais complexa
 - Tarefa pode interferir com ela mesma
- Tarefas esporádicas podem ser tratadas como periódicas
 - Intervalo mínimo entre ativações é usado como período
- A forma como prioridades são atribuídas **NÃO** é importante
 - Funciona pois $hp(i)$ sempre indica as tarefas mais prioritárias

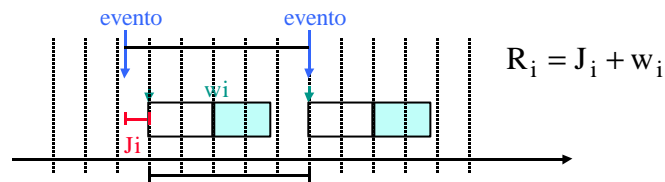
Deadline Monotonic

- Quanto menor o deadline, maior a prioridade
- Ótimo quando deadline é menor ou igual ao período
- Exemplo:
 - Tarefas
 - Períodos
 - Tempo máximo de computação
 - Deadline
 - Prioridades
 - Tempo máximo de resposta
 - Caso fosse RM

	T1	T2	T3	T4
– Períodos	P1=20	P2=15	P3=10	P4=20
– Tempo máximo de computação	C1=3	C2=3	C3=4	C4=3
– Deadline	D1=5	D2=7	D3=10	D4=20
– Prioridades	p1=1	p2=2	p3=3	p4=4
– Tempo máximo de resposta	R1=3	R2=6	R3=10	R4=20
– Caso fosse RM	10	7	4	20

Release Jitter

- Suponha uma tarefa esporádica liberada por evento externo
 - Eventos podem ser amostrados periodicamente
 - Sinalização do evento pode ter atraso variável



- **Release Jitter:** Atraso máximo na liberação da tarefa

$$w_i = C_i + \sum_{j \in hp(i)} \left\lceil \frac{w_i + J_j}{P_j} \right\rceil \times C_j$$

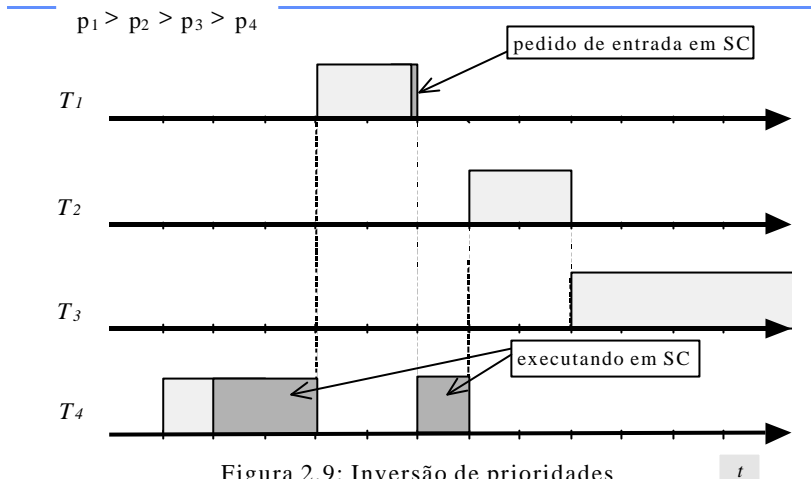
Bloqueios

- Podem ocorrer bloqueios devido a relações de exclusão mútua
 - Estruturas de dados compartilhadas
 - Dispositivos compartilhados
- Suponha T1 e T2, T1 com maior prioridade
- Se T2 fica bloqueada, esperando por T1
 - Ok, T1 tem mesmo prioridade superior
- Se T1 fica bloqueada, esperando por T2
 - Cálculo do tempo de resposta deve incluir a espera máxima B_i

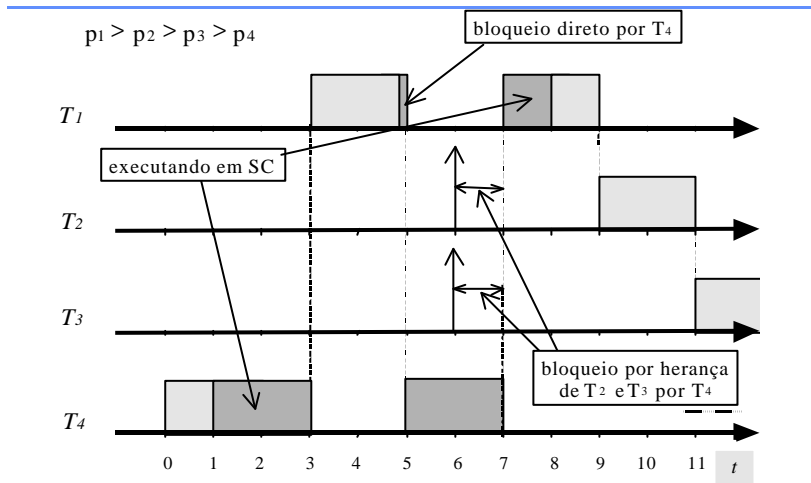
$$w_i = C_i + B_i + \sum_{j \in hp(i)} \left\lceil \frac{w_i + J_j}{P_j} \right\rceil \times C_j$$

$$R_i = J_i + w_i$$

Inversão de Prioridades



Herança de Prioridades



Herança de Prioridade

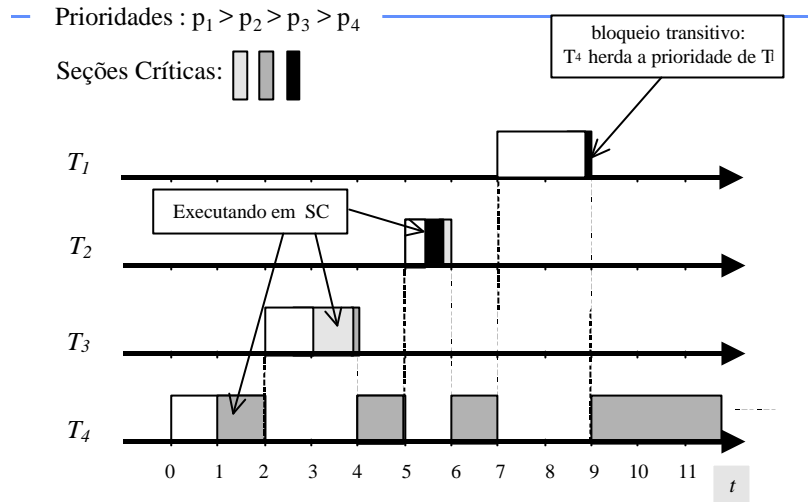


Figura 2.11: Bloqueio transitivo

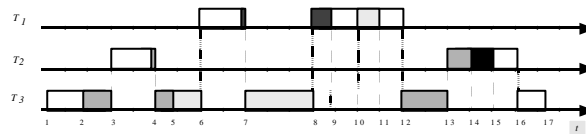
Sistemas de Tempo Real - Jean-Marie Farnes, Joni da Silva Fraga, Rômulo Silva de Oliveira

21

Prioridade Teto

Prioridades : $p_1 > p_2 > p_3$ (com $p_1 = 1$, $p_2 = 2$ e $p_3 = 3$)

Semáforos : S₁ █ S₂ █ S₃ █



Eventos	Descrição
1	T ₃ começa sua execução
2	T ₃ entra em seção crítica (fecha o Semáforo S ₃)
3	T ₂ inicia seu processamento interrompendo T ₃ (preempção de T ₃).
4	T ₂ sofre bloqueio direto: S ₃ "fechado" por T ₃ que reassume e herda prioridade de T ₂ .
5	T ₃ entra na seção crítica aninhada ao fechar semáforo S ₂
6	T ₁ inicia e interrompe T ₃ . T ₁ é mais prioritária que T ₃ com a prioridade herdada de T ₂ .
7	T ₁ tenta fechar semáforo S ₁ , é bloqueado por "ceiling"; possui prioridade igual ao maior "ceiling" de semáforo já fechado pelas outras tarefas (C(S ₂)=1).
8	T ₃ libera semáforo S ₂ . T ₁ é reativado e interrompe T ₃ . T ₁ entra em S ₁ .
9	T ₁ libera S ₁ .
10	T ₁ fecha Semáforo S ₂ .
11	T ₁ libera S ₂ .
12	T ₁ termina e T ₃ reassume na prioridade herdada de T ₂ .
13	T ₃ libera S ₃ retornando a sua prioridade estática. T ₂ interrompe T ₃ e fecha S ₃ .
14,15,16,17	T ₂ libera S ₃ ; fecha e libera S ₁ ; e completa. T ₃ reassume e completa.

Figura 2.12: Exemplo de uso do PCP

Sistemas de Tempo Real - Jean-Marie Farnes, Joni da Silva Fraga, Rômulo Silva de Oliveira

22

Servidor em Background

tarefas	C_i	P_i	D_i	p_i
tarefa periódica A	4	10	10	1
tarefa periódica B	8	20	20	2
tarefa aperiódica C	1	-	-	3
tarefa aperiódica D	1	-	-	3

tarefa A - 
 tarefa B - 
 tarefa C - 
 tarefa D - 

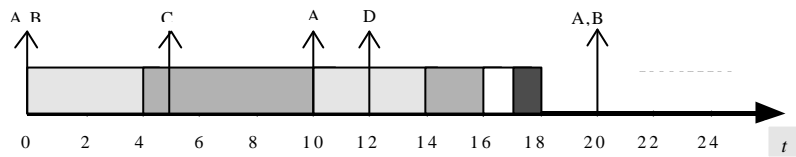






Figura 2.14: Servidora de “Background”

Servidor Polling

tarefas	C_i	P_i	D_i	p_i
tarefa periódica A	4	10	10	3
tarefa periódica B	8	20	20	2
tarefa servidora PS	1	5	-	1
tarefa aperiódica C	1	-	-	-
tarefa aperiódica D	0.5	-	-	-

tarefa A - 
 tarefa B - 
 tarefa C - 
 tarefa D - 

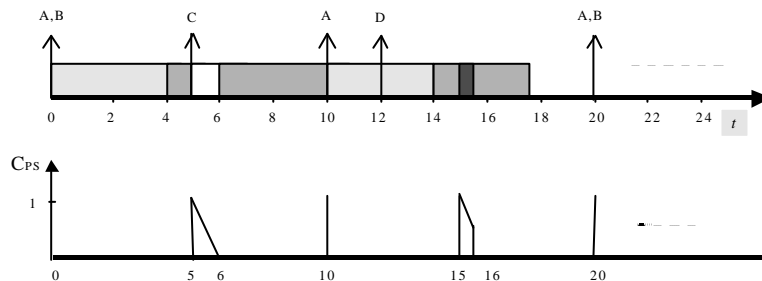
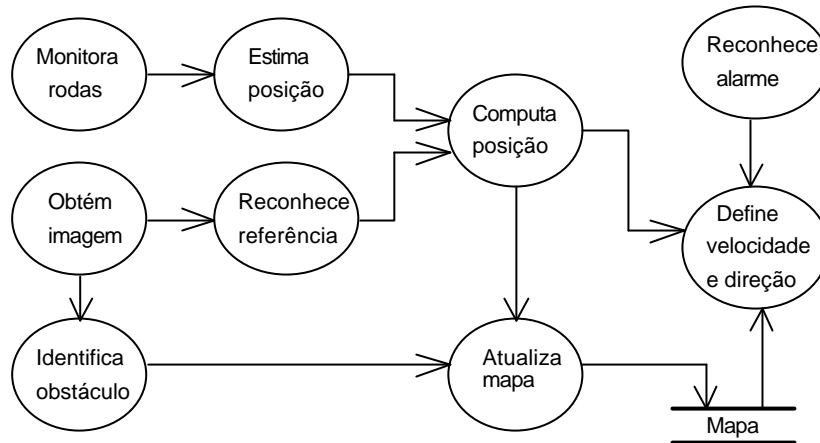


Figura 2.15: Algoritmo “Polling Server”

Diagrama da Aplicação Exemplo



Lista de Tarefas - versão inicial

Tarefa	Tipo	P (ms)	D (ms)	C (ms)	Predecessor	Bloqueio (ms)
C_P	Periódica	100	100	20		1 (c/R_R)
L_I	Periódica	500	500	20		
A_M	Periódica	500	500	100	L_I	3 (c/D_V_D)
R_R	Periódica	1300	1300	200		1 (c/C_P)
D_V_D	Periódica	100	100	30	C_P	3 (c/A_M)
E_D	Esporádica	2000	20	1		
R	Soft					

Tabela 5.1 - Definição das tarefas, versão inicial

Algumas considerações

- O tempo de execução de alguns elementos do sistema operacional são computados juntos com o tempo de execução das tarefas
- O temporizador em hardware ("*timer*") do sistema gera interrupções periodicamente
- Algumas seções críticas do "*kernel*" desabilitam as interrupções por breves instantes. A maior seção de código do "*kernel*" que executa com interrupções desabilitadas o faz durante $B_k = 0.1$ ms.
- Caso uma interrupção do "*timer*" aconteça enquanto o "*kernel*" está com interrupções desabilitadas, ela somente será tratada com um atraso (latência) de B_k .
- Outras

Lista de Tarefas - versão final

Tarefa	Tipo	P (ms)	D (ms)	C (ms)	J (ms)	Predecessor	Bloqueio (ms)
timer	periódica	10	10	0,1	B_k		
C_P	periódica	100	100	20	B_k		1 ($B_{R,R}$)
L_I	periódica	500	500	20	B_k		
A_M	periódica	500	500	100	$R_{L,I}$	L_I	
R_R	periódica	1300	1300	200	B_k		
D_V_D	periódica	100	100	30	$R_{C,P}$	C_P	3 ($B_{A,M}$)
E_D	esporádica	2000	20	1	B_k		B_m
R	servidora	10000	80	5	B_k		

Tabela 5.2: Definição das Tarefas, Versão Final