

Universidade Federal do Amazonas
Faculdade de Tecnologia
Programa de Pós-graduação em Engenharia Elétrica



Escalonamento (Tarefas Esporádicas)

Lucas Cordeiro

lucascordeiro@ufam.edu.br

Notas de Aula



Baseado nas notas de aula do Prof. Francisco Vasques, da Faculdade de Engenharia da Universidade do Porto.

<http://www.fe.up.pt/~vasques>

Baseado no livro de “Sistemas de Tempo Real” dos Professores Jean-Marie Farines, Joni da Silva Fraga e Rômulo Silva de Oliveira da 12ª Escola de Computação, IME-USP, São Paulo-SP, 24 a 28 de julho de 2000.

Tarefas Esporádicas



- Por definição, uma tarefa esporádica pode estar um **longo período de tempo sem ser ativada**. No entanto, após a sua ativação, esta deverá ser executada a tempo
 - Em consequência, o algoritmo DM é particularmente adequado para o escalonamento de conjuntos de tarefas com tarefas esporádicas
- De uma forma genérica, para um conjunto de tarefas com pelo menos uma tarefa esporádica, o teste de escalonabilidade através da análise do **tempo de resposta** será um teste **suficiente**

Análise de Escal. de Tarefas Esporádicas

1. **Abordagem da resposta garantida:** todas as tarefas com metas temporais críticas deverão ser escalonáveis para os seus tempos máximos de execução e as suas taxas máximas de ativação (pior caso)
 - A 1ª regra garante o **respeito das metas temporais** para todas as **tarefas críticas**
2. **Abordagem do melhor esforço:** todas as tarefas (críticas e não críticas) deverão ser escalonáveis quando são considerados os seus tempos médios de execução e as suas taxas médias de ativação (caso médio)
 - Uma consequência da 2ª regra é que possivelmente **nem todas as metas temporais das tarefas não críticas** serão **cumpridas** no caso de uma sobrecarga transitória

Tarefas Esporádicas

- Considerando que, na maior parte dos casos, **os valores de T** (intervalo mínimo de tempo entre ativações consecutivas) para as tarefas esporádicas **são muito menores que os intervalos de tempo reais** entre ativações consecutivas:
 - O cálculo de **testes de escalonabilidade** baseado nos valores de T será muito **pessimista**
 - A **utilização total** admissível para o sistema será **muito reduzida**
 - A **utilização de servidores** é aconselhável para efetuar o escalonamento de tarefas esporádicas

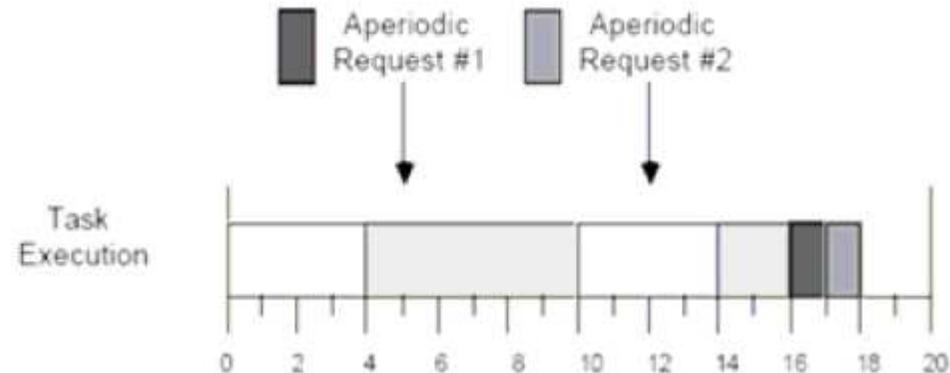
Uso de Servidores

- Tarefas cujo intervalo de tempo entre ativações consecutivas não tem mínimo definido
- 1ª solução: atribuir os menores valores de prioridade às tarefas aperiódicas (ou esporádicas)
 - Consequência: difícil garantir o respeito das metas temporais
- Para poder ser considerada a utilização de tarefas aperiódicas em sistemas de tempo-real, torna-se necessário impor um **limite superior** à sua utilização de recursos computacionais através da utilização de servidores

Utilização de Servidores *Background* (1)

- As tarefas aperiódicas são executadas unicamente quando o processador está ocioso

<i>tarefa</i>	<i>C</i>	<i>T</i>	<i>d</i>	<i>U</i>
τ_1	4	10	10	0,4
τ_2	8	20	20	0,4
				<i>U total: 0,8</i>



Utilização de Servidores *Background* (2)

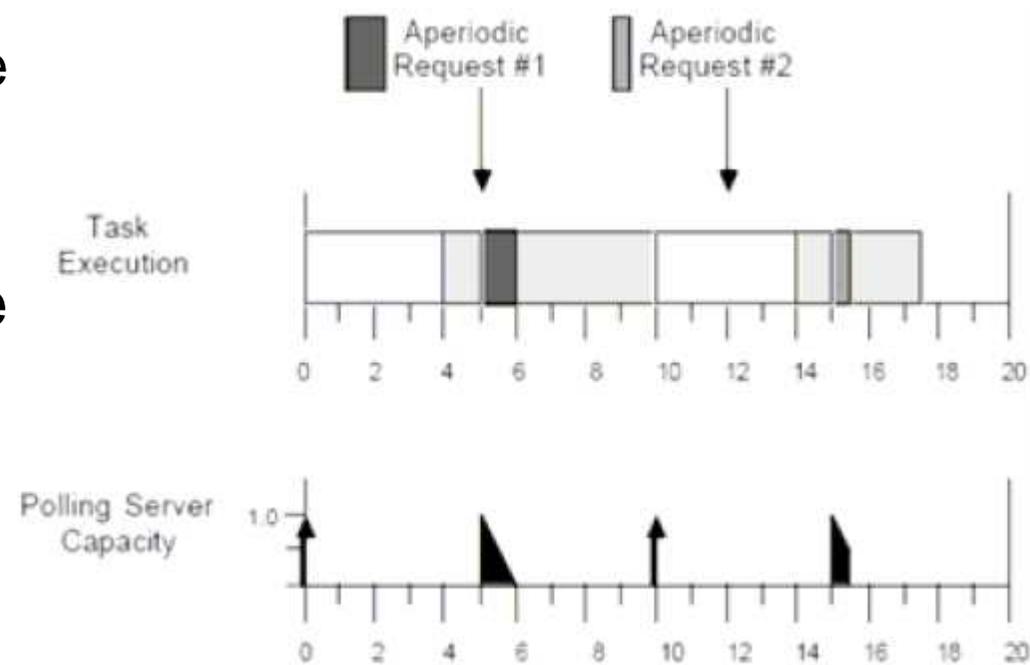


- Vantagens
 - Tem um **impacto nulo** sobre o escalonamento das tarefas periódicas (condições de escalonabilidade mantêm-se)
 - **Simplicidade** de implementação
- Desvantagens
 - A **capacidade de processamento** atribuída às tarefas aperiódicas depende da carga imposta pelas tarefas periódicas
 - O **tempo de resposta** a pedidos de ativação de tarefas aperiódicas pode ser muito **longo**

Utilização de Servidores *Polling* (1)

- Uma tarefa periódica extra: o servidor de “*polling*” é adicionado ao conjunto de tarefas a escalonar
 - Tarefa com capacidade C_s e periodicidade T_s
- Durante o tempo de execução do servidor de “*polling*” (C_s), as tarefas aperiódicas serão executadas

<i>tarefa</i>	C	T	d	U
$\tau_{Polling}$	1	5	5	0,2
τ_1	4	10	10	0,4
τ_2	8	20	20	0,4
				$U \text{ total: } 1$



Utilização de Servidores *Polling* (2)

- Teste de escalonabilidade para as tarefas periódicas:
 - Teste suficiente de escalonabilidade (RM)

$$U = \sum_{i=1}^n \frac{C_i}{T_i} + \frac{C_s}{T_s} \leq (n+1)(\sqrt[2]{2} - 1)$$

- Extensão a m servidores com prioridades diferentes para o escalonamento de tarefas aperiódicas com relevâncias diferentes

$$U = \sum_{i=1}^n \frac{C_i}{T_i} + \sum_{j=1}^m \frac{C_j}{T_j} \leq (n+m)(\sqrt[2]{2} - 1)$$

Utilização de Servidores *Polling* (3)

- Vantagens:
 - **Simplicidade**, quando se considera o teste de escalonabilidade para as tarefas periódicas
 - Fornece um **melhor serviço** às tarefas aperiódicas, quando comparado com o “*background server*”
- Desvantagens:
 - A **capacidade do servidor é perdida**, caso não existam tarefas aperiódicas com pedidos de execução ativados
 - **Não é capaz de fornecer uma resposta imediata** às tarefas aperiódicas

Exercício

- Verifique se o conjunto de tarefas abaixo é escalonável usando a técnica de servidores *polling*. Caso positivo, apresente o diagrama temporal considerando duas solicitações aperiódicas no tempo $t=4$ e $t=9$ com durações de 1 u.t.

Tarefa	Computação	Período	Deadline
T_{polling}	1	4	4
T_2	2	6	6
T_3	3	10	9

Servidores *Sporadic*



- O algoritmo SS implementa uma tarefa periódica (servidor) de alta prioridade para servir pedidos aperiódicos, que **mantém a sua capacidade de execução** até que um pedido de ativação de uma tarefa aperiódica ocorra
- É equivalente ao algoritmo DM, exceto no que diz respeito aos instantes em que a capacidade do servidor é recolocada no seu valor máximo

Definições (1)

- Definições:

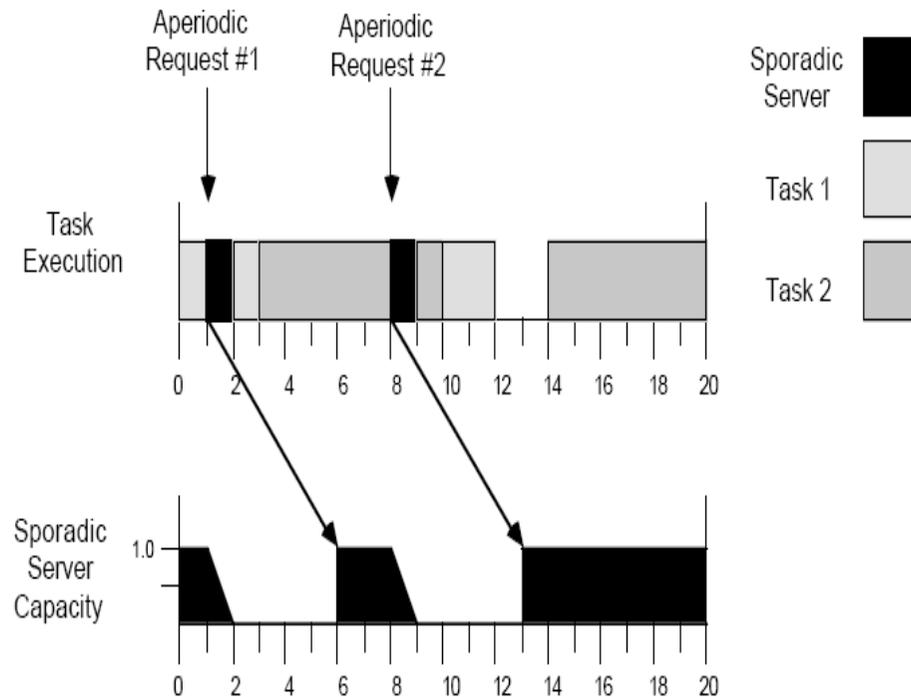
- P_s : nível de prioridade da tarefa em curso de execução
- P_i : nível de prioridade de tarefa
 - P_1 é a maior prioridade do sistema
- RT_i : Instante de recarregamento para o nível de prioridade P_i
- Ativo: O nível de prioridade P_i está ativo se a prioridade atual do sistema, P_s , é maior ou igual que a prioridade P_i :
 $P_s \geq P_i$
- Ocioso: O nível de prioridade P_i está ocioso (“idle”) se
 $P_s < P_i$

Definições (2)

- Para um servidor esporádico que execute a um nível de prioridade P_i
 - Se o servidor ainda tiver tempo de execução disponível, o seu instante de recarregamento RT_i é definido no instante t em que o nível de prioridade P_i fica ativo
 - Se a sua **capacidade já estiver esgotada**, o seu instante de recarregamento RT_i será definido quando a capacidade do servidor for de novo não nula e P_i estiver ativo
 - Em ambos os casos: $RT_i = t + T_i$
 - O recarregamento da capacidade do servidor será igual ao tempo de execução consumido desde a ultima vez que P_i mudou de ocioso para ativo

Utilização de Servidores *Sporadic* (1)

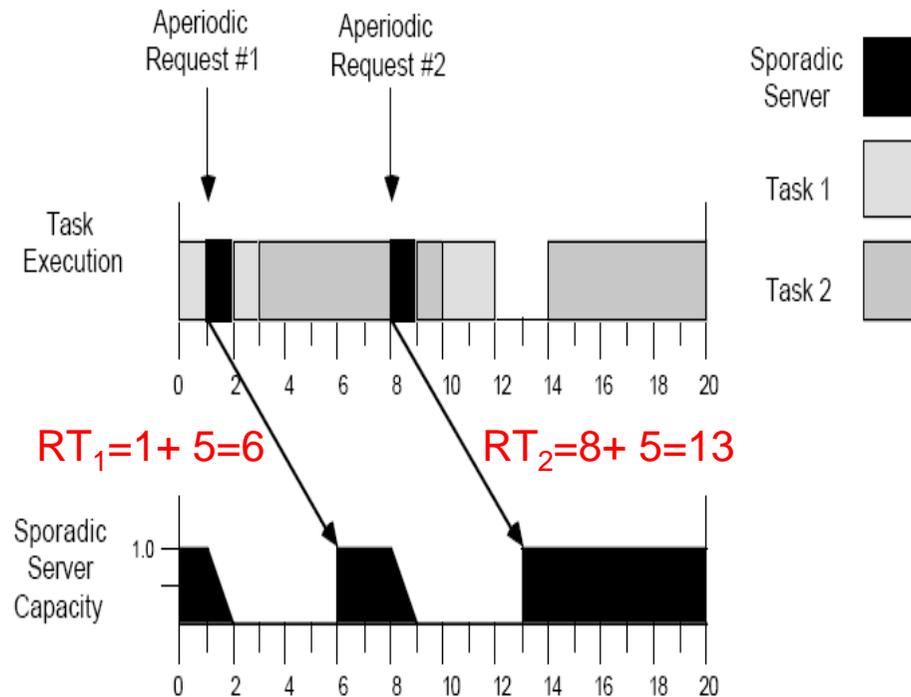
- SS com o nível de prioridade mais elevado: P1
 - Como o SS é a única tarefa com o nível de prioridade mais elevado, P1 só fica ativo quando existe um pedido de execução de uma tarefa aperiódica.
 - Logo, RT_1 só é definido neste instante
 - Logo, a capacidade do servidor é reposta um período após o pedido de execução de uma tarefa aperiódica



<u>Task</u>	<u>Exec Time</u>	<u>Period</u>	<u>Utilization</u>
SS	1	5	20.0%
τ_1	2	10	20.0%
τ_2	6	14	42.9%

Utilização de Servidores *Sporadic* (2)

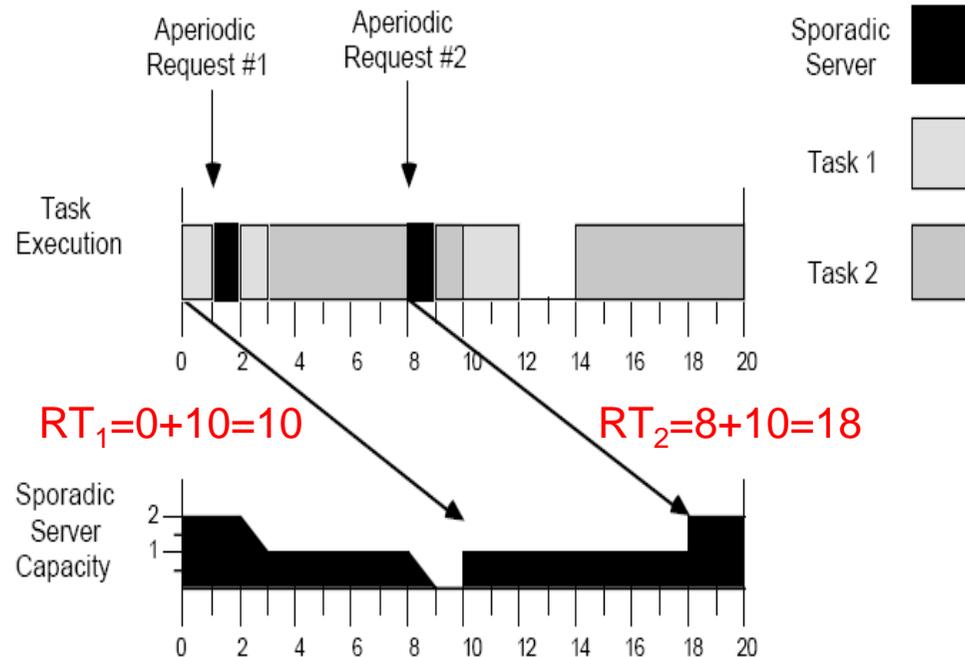
- **SS com o nível de prioridade mais elevado: P1**
 - Como o SS é a única tarefa com o nível de prioridade mais elevado, P1 só fica ativo quando existe um pedido de execução de uma tarefa aperiódica
 - Logo, RT_1 só é definido neste instante
 - Logo, a **capacidade do servidor é reposta um período após o pedido de execução de uma tarefa aperiódica**



<u>Task</u>	<u>Exec Time</u>	<u>Period</u>	<u>Utilization</u>
SS	1	5	20.0%
τ_1	2	10	20.0%
τ_2	6	14	42.9%

Utilização de Servidores *Sporadic* (3)

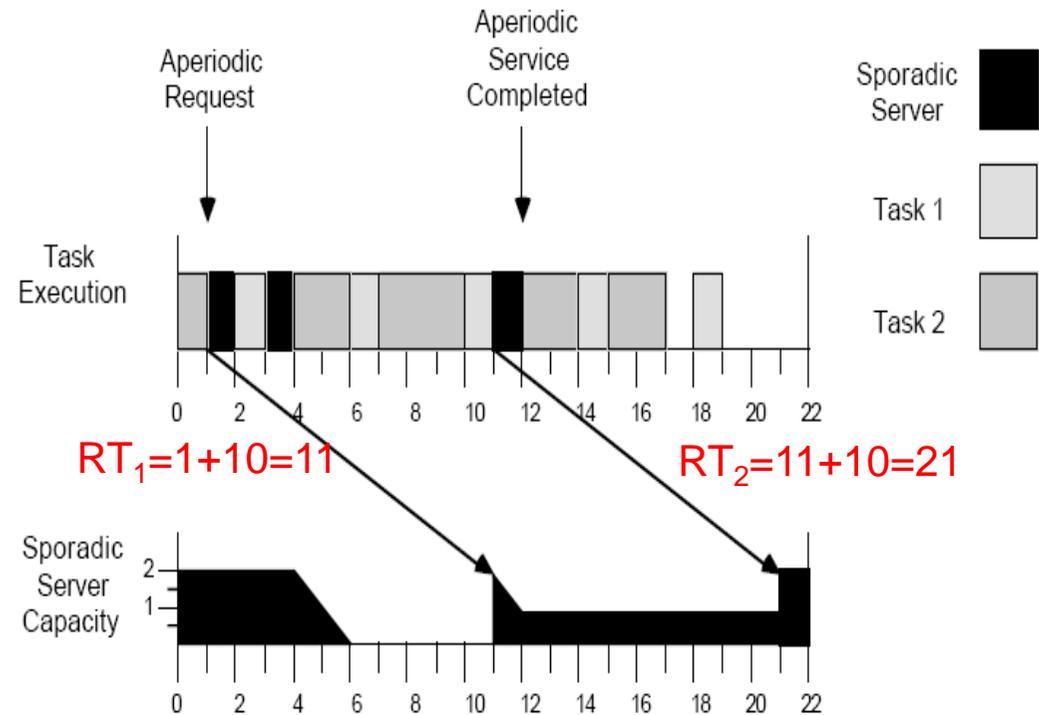
- Outra tarefa com o mesmo nível de prioridade do servidor (P1).**
 - No instante $t = 0$, τ_1 inicia a sua execução, P_1 fica ativo e RT_1 é definido
 - No instante $t = 1$, o primeiro pedido aperiódico é servido
 - No instante $t = 3$, τ_1 termina a sua execução, P_1 passa a ocioso, e o recarregamento no instante $t = 10$ é definido: 1 unidade de tempo



Task	Exec Time	Period	Utilization
SS	2	10	20.0%
τ_1	2	10	20.0%
τ_2	6	14	42.9%

Utilização de Servidores *Sporadic* (4)

- Efeito da exaustão da capacidade do servidor, devido a pedido aperiódico para um tempo de execução superior à capacidade do servidor

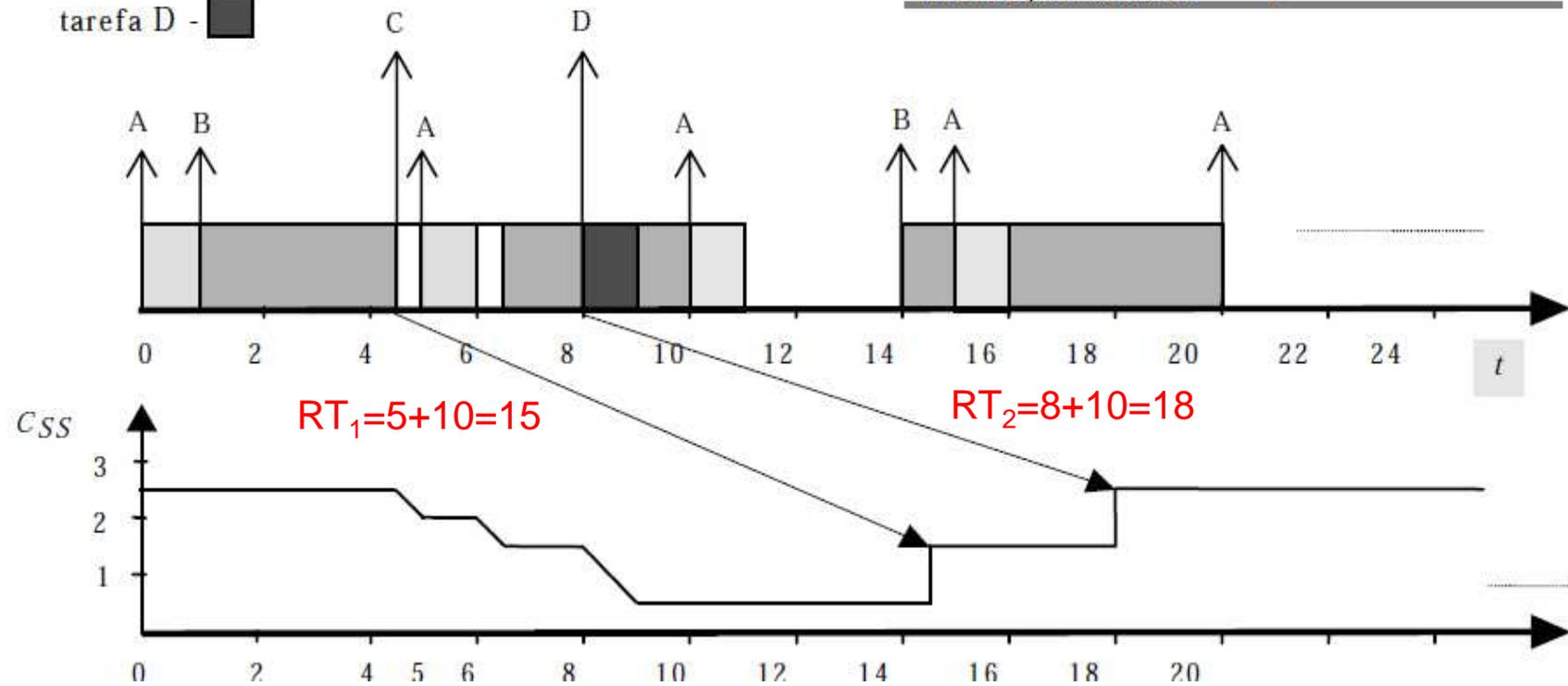


Task	Exec Time	Period	Utilization
τ_1	1	4	25%
SS	2	10	20%
τ_2	10	40	25%

Utilização de Servidores *Sporadic* (5)

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

<i>tarefas</i>	C_i	P_i	D_i	p_i
tarefa periódica A	1	5	5	1
tarefa periódica B	6	14	14	3
tarefa servidora SS	2,5	10	-	2
tarefa aperiódica C	1	-	-	-
tarefa aperiódica D	1	-	-	-



Exercício

- Calcule a capacidade do servidor *sporadic* para o seguinte conjunto de tarefas. Considere que as solicitações aperiódicas ocorrem nos tempos $t=12$ e $t=18$ com duração de 4 e 8 u.t. respectivamente.

Tarefa	Computação	Período	Deadline	Prioridade
T_1	4	12	12	2
T_2	4	20	20	3
T_{ss}	8	32	10	1
T_3	8	10	10	4