

# CIn-UFPE STR

Escalonamento de Tarefas Aperiódicas de STR Hard

---

Anderson L. S. Moreira

André L. M. Silva

Daniel N. Leite

Edgard N. A. G. Lima

Guilherme Álvaro R. M. Esmeraldo

Jordana L. Seixas

# Agenda (Geral)

---

- Background Server e Polling Server;
- Deferrable Server e Priority Exchange Server;
- Sporadic Server
- Teoremas e Análise do SS.

# Agenda

---

- Escalonamento de Tarefas Aperiódicas;
- Background Server;
- Polling Server.

# Escalonamento de tarefas Aperiódicas 1/3

---

- Sistemas reais: Tarefas Periódicas e Aperiódicas;
- Tarefas aperiódicas apresentam ocorrência não determinística;

# Escalonamento de tarefas Aperiódicas 2/3

---

- Tarefas Periódicas são consideradas críticas. Necessitam de garantia em tempo de projeto;
- Tarefas Aperiódicas: Bom tempo médio de resposta;
- Em geral:
  - Tarefas Aperiodicas: Soft Deadline;
  - Tarefas Periódicas: Hard Dealine;

# Escalonamento de tarefas Aperiódicas 3/3

---

- Solução:
  - Abordagem híbrida;
  - Baseada nas sobras de processador na escala de carga periódica;

# Background Server (BS) 1/2

---

- O ***Background Server*** é o mais simples dos servidores.
- Sua estratégia é processar as requisições aperiódicas quando o processador estiver desocupado, isto é, processar as tarefas aperiódicas apenas quando não houver tarefas periódicas pendentes.

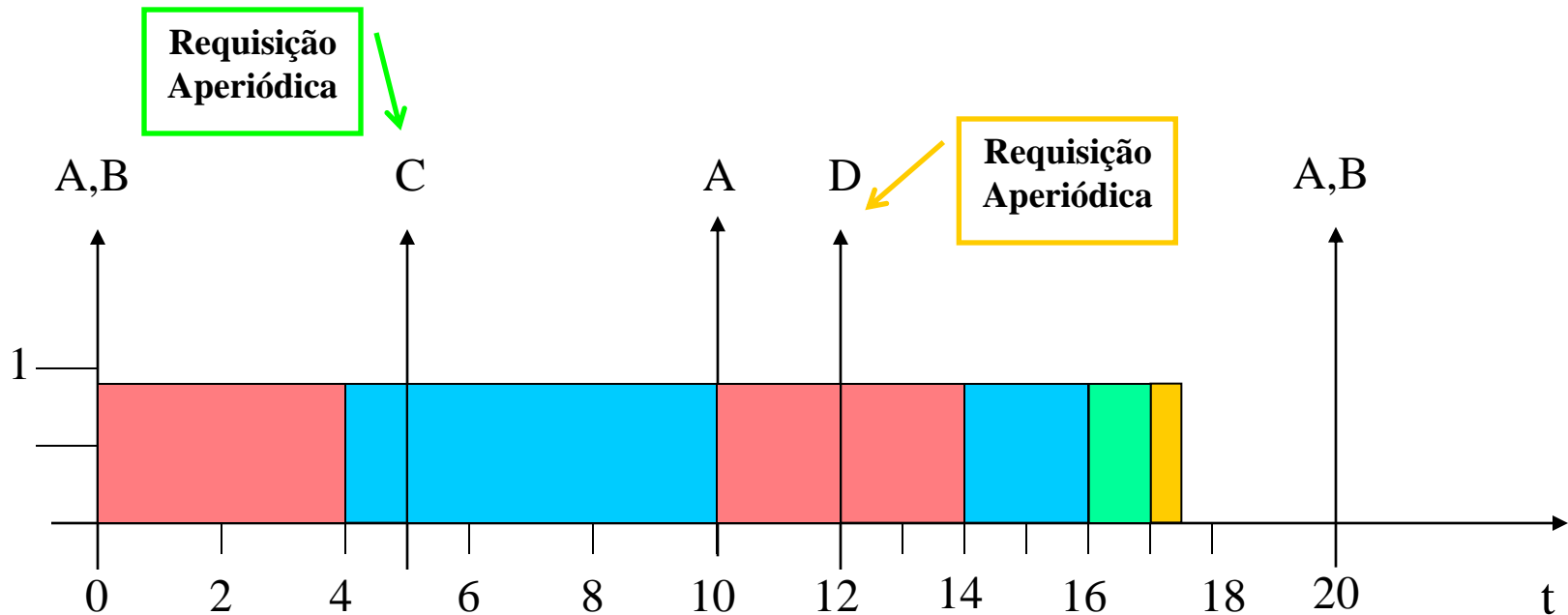
# Background Server (BS) 2/2

---

- Implementação bastante simples;
- Grande carga periódica resulta em um alto tempo de resposta para tarefas aperiódicas;
- Requer:
  - Tarefas aperiódicas não críticas;
  - Baixa carga periódica;



| Tarefas      | Tempo de Computação $C_i$ | Período $P_i$ | Deadline $D_i$ | Prioridade $p_i$ |
|--------------|---------------------------|---------------|----------------|------------------|
| $T_A$        | 4                         | 10            | 10             | 3                |
| $T_B$        | 8                         | 20            | 20             | 2                |
| Aperiódica C | 1                         | -             | -              | -                |
| Aperiódica D | 0,5                       | -             | -              | -                |



Podemos observar neste exemplo que as tarefas aperiódicas só serão executadas depois de toda a carga periódica, causando um longo tempo de resposta.

# Polling Server (PS) 1/3

---

- O Polling Server consiste na criação de uma tarefa servidora periódica para atender a carga aperiódica;
- Esta tarefa servidora tem período, prioridade e tempo de computação definidos;
- Em cada ativação, a tarefa servidora executa as tarefas aperiódicas pendentes durante seu tempo de computação;

# Polling Server (PS) 2/3

---

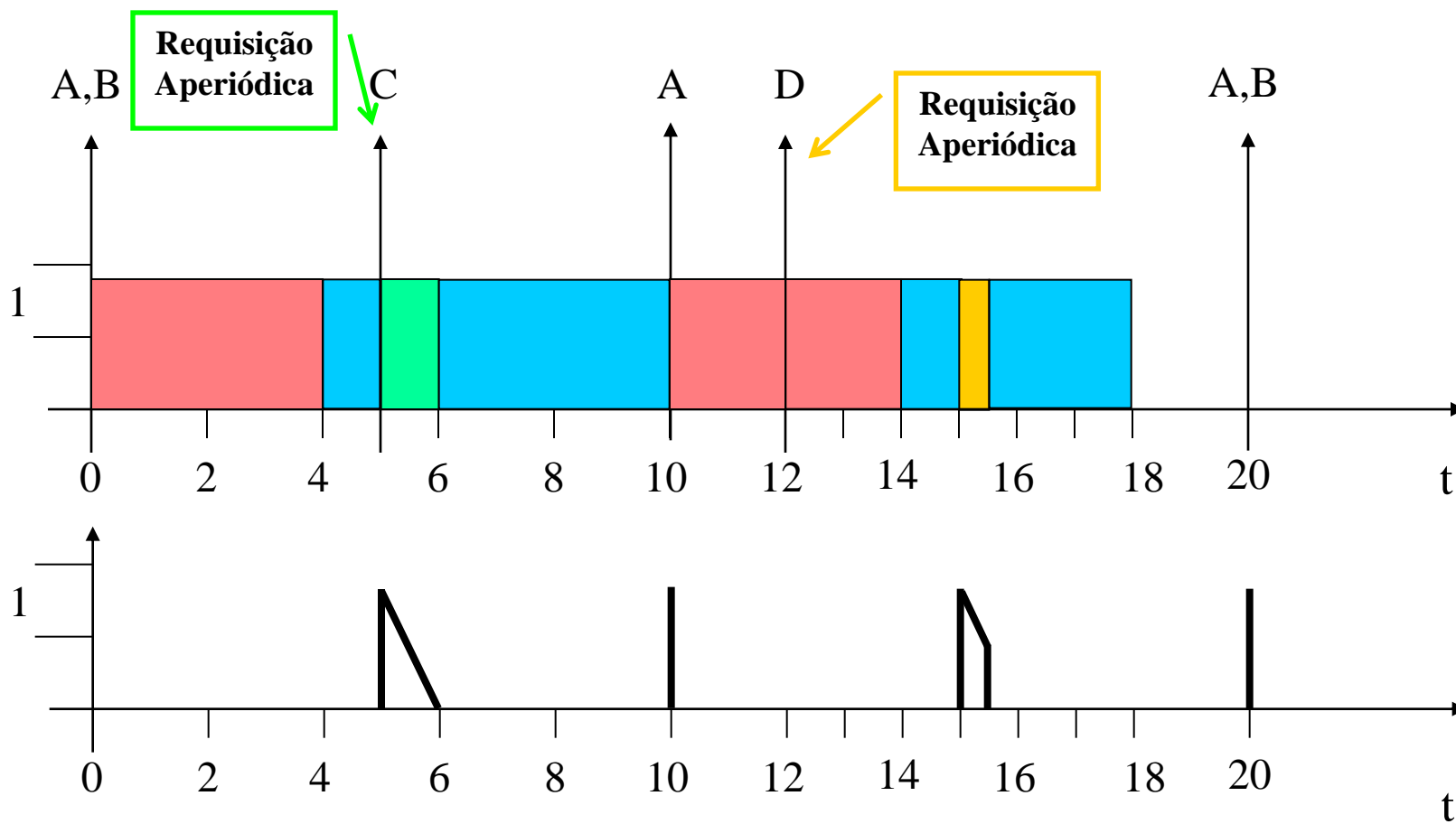
- Caso não haja tarefas aperiódicas pendentes, a tarefa servidora é suspensa até sua próxima ativação e o processamento volta para as tarefas periódicas;
- Se uma tarefa aperiódica é requisitada logo após a suspensão da tarefa servidora, a tarefa aperiódica deve esperar até a próxima ativação da tarefa servidora.

# Polling Server (PS) 3/3

---

- A interferência da tarefa servidora no sistema é no pior caso igual a interferência causada por uma tarefa com o mesmo tempo de computação e período.
- O em relação ao BS, o PS melhora o tempo de resposta médio de tarefas aperiódicas, porém não possui serviço de resposta imediato.

| Tarefas        | Tempo de Computação<br>$C_i$ | Período<br>$P_i$ | Deadline<br>$D_i$ | Prioridade<br>$p_i$ |
|----------------|------------------------------|------------------|-------------------|---------------------|
| $T_A$          | 4                            | 10               | 10                | 2                   |
| $T_B$          | 8                            | 20               | 20                | 3                   |
| T Servidora PS | 1                            | 5                | -                 | 1                   |
| Aperiódica C   | 1                            | -                | -                 | -                   |
| Aperiódica D   | 0,5                          | -                | -                 | -                   |



# Agenda

---

- Deferrable Server (DS);
- Características do P. Exchange (PE);
- DS x PE.

# Deferrable Server (DS)

---

- O algoritmo *Deferrable Server*, supera as desvantagens das requisições aperiódicas associadas aos servidores *Polling Server* (PS) e *Background Server* (BS);
- Como o PS, o algoritmo de DS cria tarefa aperiódica, geralmente de alta prioridade, para servidores de requisições aperiódicas;
- Diferente do PS, o DS preserva o tempo de execução das tarefas aperiódicos ( $C_{DS}$ ), conserva a sua capacidade;

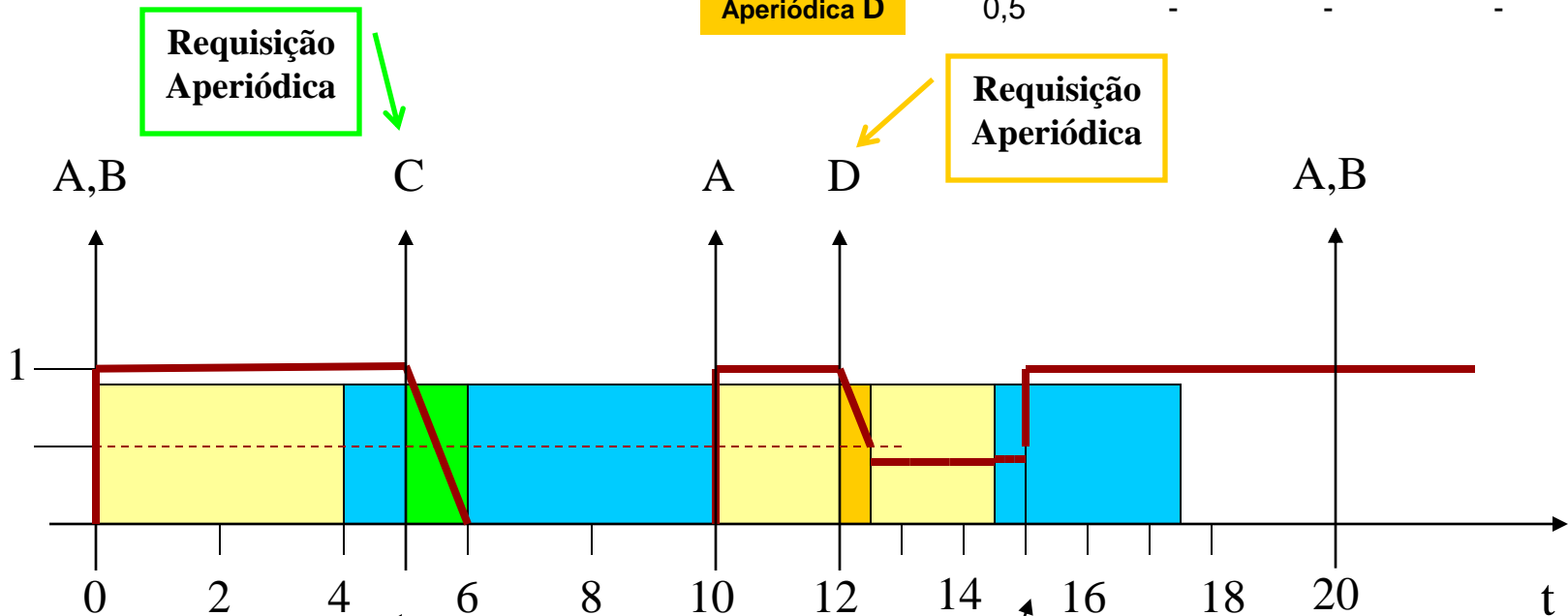
# Deferrable Server (DS)

---

- O *Deferrable Server* mantém o tempo de execução aperiódico, pela duração do período da tarefa servidora (DS);
- Assim, requisições aperiódicas podem ser atendidas com alta prioridade no tempo, enquanto o tempo de execução da tarefa servidora não tenha terminado;
- No início de cada período da tarefa servidora (DS), o tempo de execução é recuperado para sua capacidade completa.



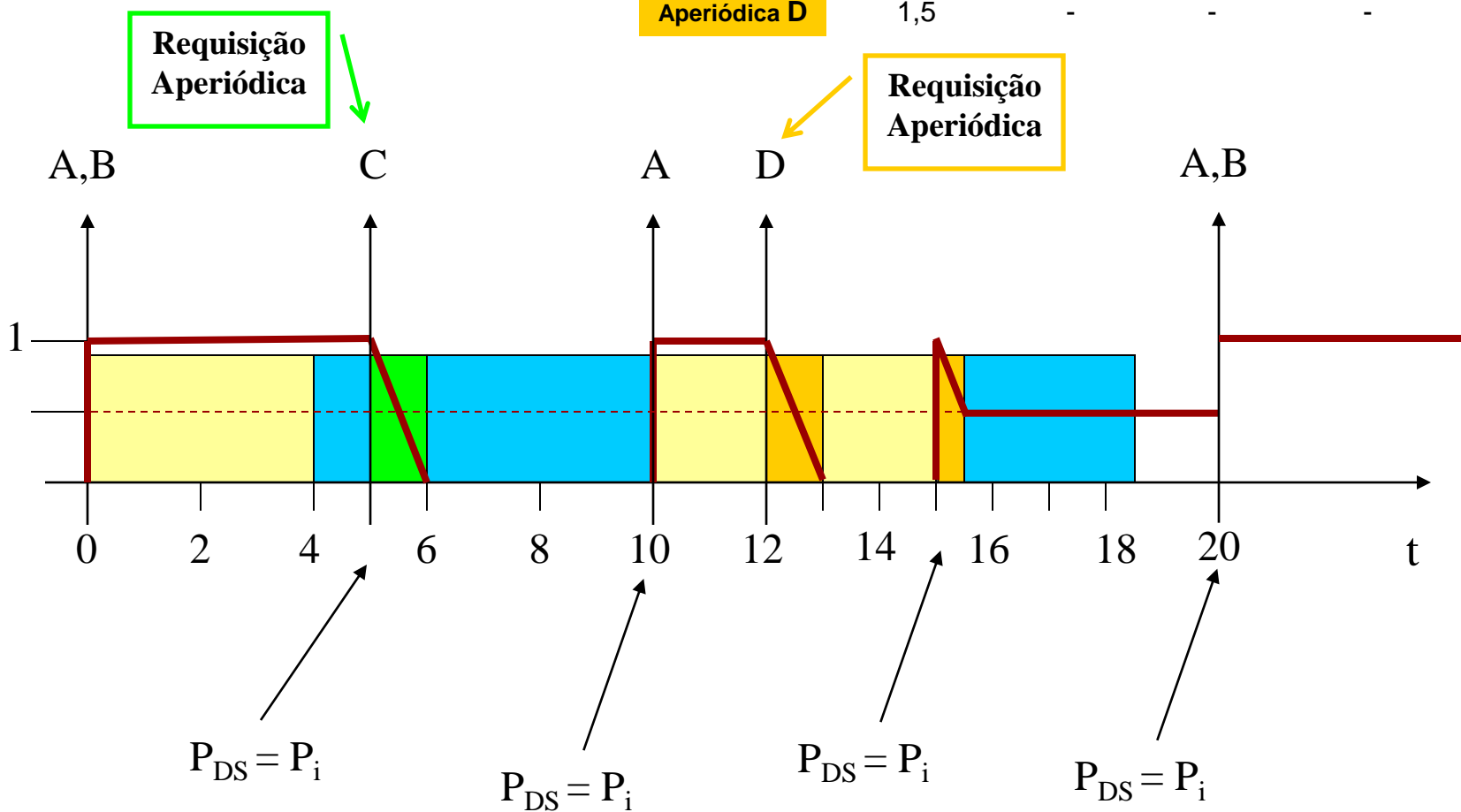
| Tarefas      | Tempo de Computação $C_i$ | Período $P_i$ | Deadline $D_i$ | Prioridade $p_i$ |
|--------------|---------------------------|---------------|----------------|------------------|
| Servidora DS | 1                         | 5             | -              | 1                |
| $T_A$        | 4                         | 10            | 10             | 3                |
| $T_B$        | 8                         | 20            | 20             | 2                |
| Aperiódica C | 1                         | -             | -              | -                |
| Aperiódica D | 0,5                       | -             | -              | -                |



Podemos observar neste exemplo o melhor desempenho do servidor DS, comparando-se aos **servidores Polling Server (PS) e Background Server (BS)**, em relação ao tempo de resposta e serviço de resposta imediata.

$P_{DS} = P_i$      
  $P_{DS} = P_i$      
  $P_{DS} = P_i$

| Tarefas      | Tempo de Computação $C_i$ | Período $P_i$ | Deadline $D_i$ | Prioridade $p_i$ |
|--------------|---------------------------|---------------|----------------|------------------|
| Servidora DS | 1                         | 5             | -              | 1                |
| $T_A$        | 4                         | 10            | 10             | 3                |
| $T_B$        | 8                         | 20            | 20             | 2                |
| Aperiódica C | 1                         | -             | -              | -                |
| Aperiódica D | 1,5                       | -             | -              | -                |



# Deferrable Server (DS)

- Segundo [Farines, et al. 2000] quando é usada a política RM, a influência da tarefa servidora DS sobre a utilização da carga periódica não pode ser determinada de maneira tão simples como no caso do PS;
- Isso porque a capacidade da servidora  $C_{DS}$  (que é preservada) é considerada nos cálculos da utilização da servidora;
- Com ajuste no teste do RM para captar o comportamento singular da servidora DS, é apresentada uma relação entre:

$$U_P(U_{DS}) = U_{DS} + l_n \frac{U_{DS} + 2}{2U_{DS} + 1}$$

Utilização da  
carga periódica

Utilização da  
servidora

Onde:

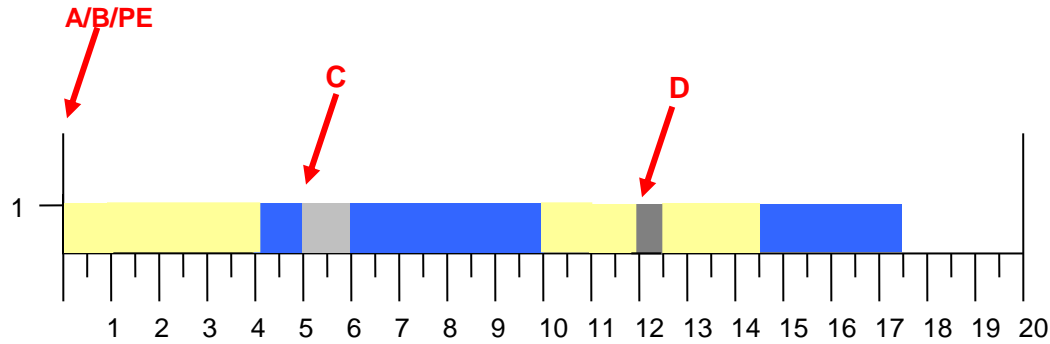
$$U_{DS} = \frac{C_{DS}}{P_{DS}} \text{ e}$$

$$U_P = \sum \frac{C_i}{P_i} \leq n(2^{1/n} - 1)$$

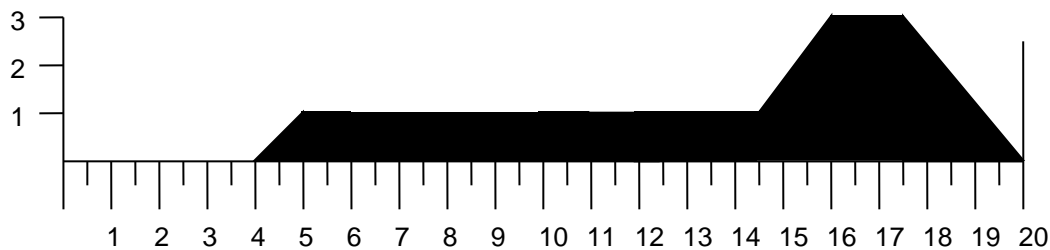
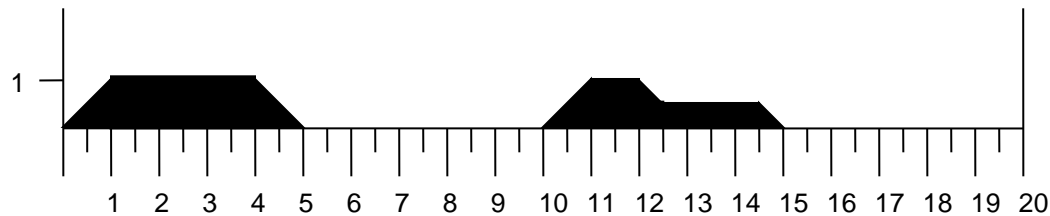
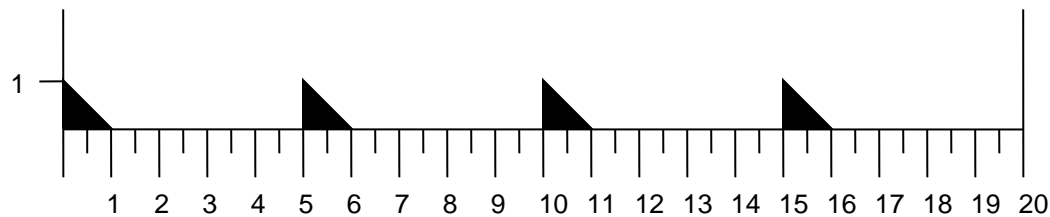
# Características do P.Exchange

---

- Tarefa Servidora continua com prioridade máxima
  - Atende requisições aperiódicas com tempo de resposta rápido
- Preserva Tempo de Computação
  - Usa os níveis de prioridades das tarefas periódicas



| Tarefas             | Ci  | Pi | Pi |
|---------------------|-----|----|----|
| tarefa periódica A  | 4   | 10 | 2  |
| tarefa periódica B  | 8   | 20 | 3  |
| tarefa servidora PE | 1   | 5  | 1  |
| tarefa aperiódica C | 1   | .. | .. |
| tarefa aperiódica D | 0,5 | .. | .. |



# DS x PE

- Implementação:
  - DS mais simples
- Escalonabilidade:
  - Limite do PE maior

$$U_{DS}(U_S) = U_S + \ln \frac{U_S + 2}{(2U_S + 1)}$$

$$U_{PE}(U_S) = U_S + \ln \frac{2}{(U_S + 1)}$$

**Exemplo:**

$$U_P = 60\%$$

$$U_{DS} = 7\%$$

$$U_{PE} = 10\%!!!$$

# Agenda

---

- Escalonamento de Tarefas Aperiódicas (soft deadline);
- Algoritmo do Servidor Esporádico (SS);
- Exemplo do Algoritmo SS;
- Servidor Esporádico Dinâmico (DSS);
- Escalonabilidade de Servidores Esporádicos;
- Performance de Servidores Esporádicos;
- Escalonamento de tarefas esporádicas.

# Escalonamento de tarefas aperiódicas (soft deadline)

---

- Prover um método que garanta os deadlines de tarefas periódicas;
- Não pode haver perda dos deadlines das periódicas;
- Tempo de resposta rápido para tarefas aperiódicas.



# O Algoritmo do Servidor Esporádico

---

- Problemas nos servidores passados:
  - DS: não eficiente para grandes cargas de tarefas;
  - PE: muito complexo de implementar.
- Algoritmo SE ou SS:
  - Cria uma tarefa com prioridade alta
  - Também preserva o tempo de execução para tarefas aperiódicas
  - Diferente no processo de recarregar o tempo de execução (RT).

# O Algoritmo do Servidor Esporádico

---

- $P_S$  - Nível da tarefa que o sistema está atualmente executando;
- $P_i$  - Um dos níveis de prioridade do sistema;
- *Ativo* - Um nível de prioridade  $P_i$  é considerado ativo se o nível de prioridade  $P_S$  é maior ou igual que a prioridade de  $P_i$  ( $P_S \geq P_i$ );
- *Ocioso* - É o oposto do *Ativo*, ou seja  $P_S < P_i$ ;
- *RT* - Tempo de carregamento que o servidor usará.

# O Algoritmo do Servidor Esporádico

---

Regarregamento do tempo de execução do SS em uma prioridade  $P_i$  obedece as seguintes premissas:

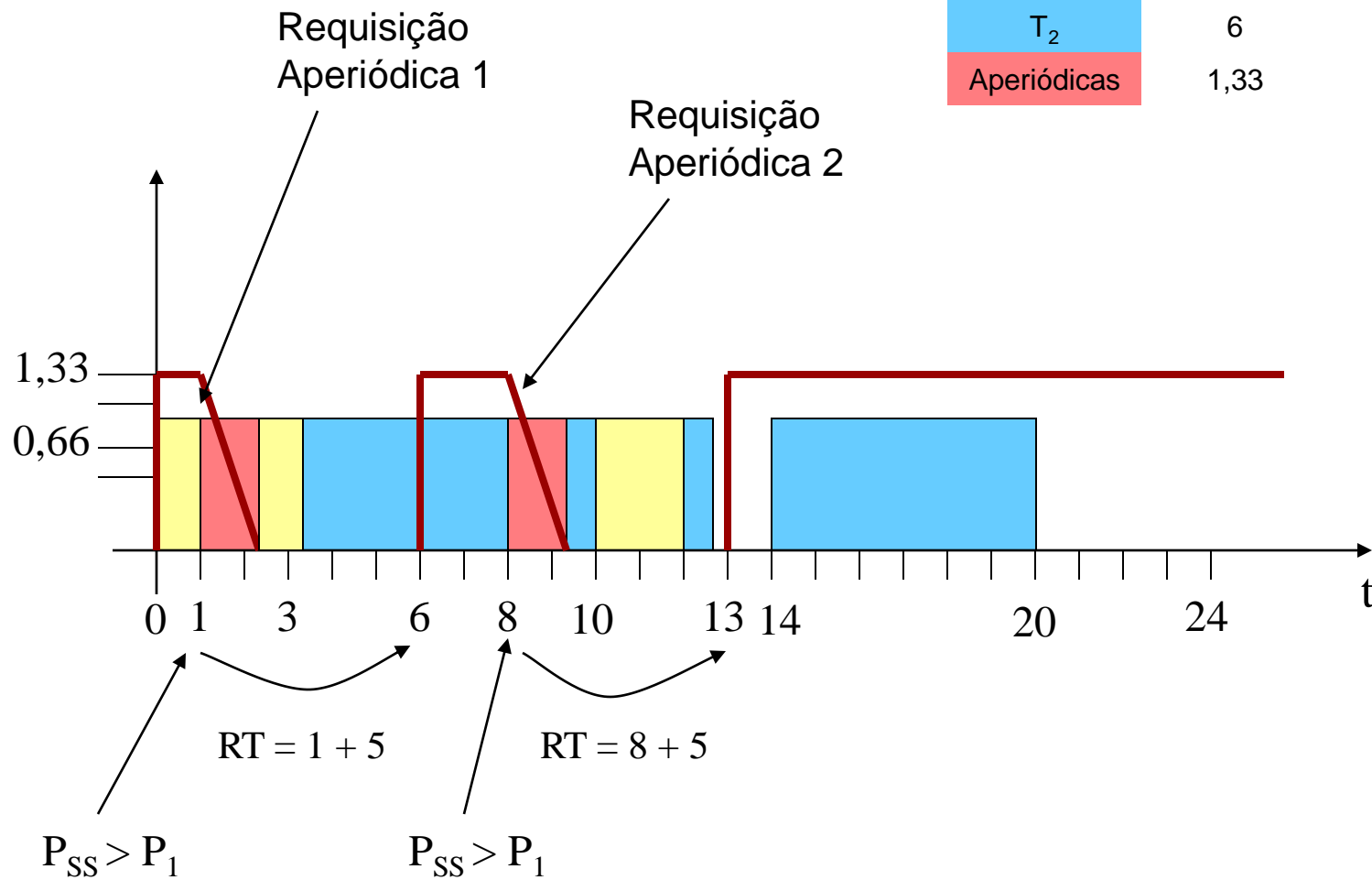
- Se o servidor tiver tempo de execução disponível, o  $RT_i$  é inicializado quando o nível de prioridade  $P_i$  se torna *Ativo*.  $RT_i = t_{atual} + P_i$ ;
- Se a capacidade do tempo de execução for esgotada, o próximo RT deve ser quando a capacidade do servidor for não-zero e  $P_i$  estiver *Ativo*;
- Regarregamento de qualquer tempo de execução é escalonado para que ocorra  $RT_i$ ;
- A quantidade a ser recarregada é igual a quantidade consumida pelo tempo de execução do servidor.

# Exemplo do algoritmo do SS

---

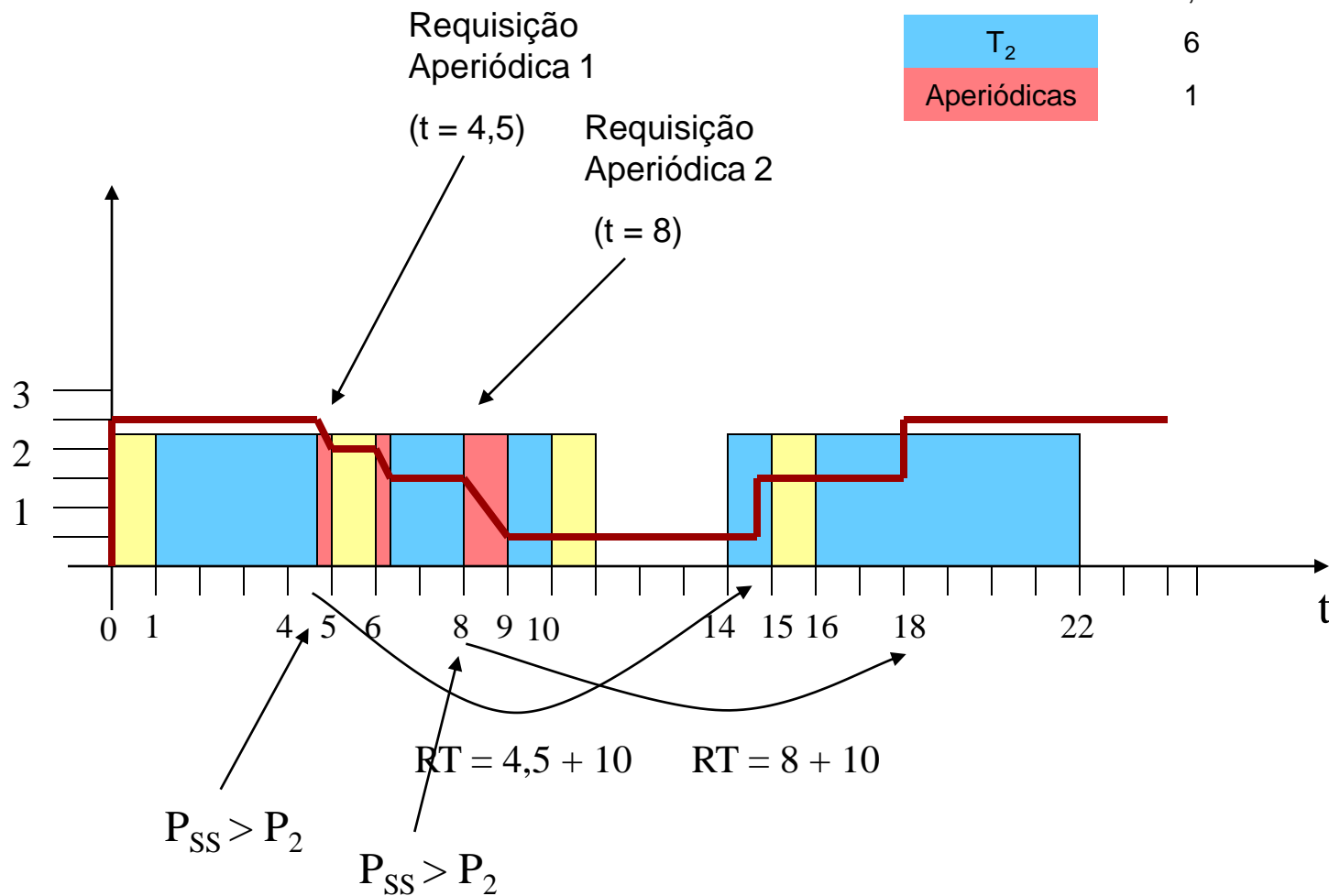
- Duas tarefas periódicas  $T_1$  e  $T_2$  e duas tarefas aperiódicas;
- Comparação com os exemplos do DS e PE;
- Comparação da carga do processador.

| Tarefa      | Tempo de Computação | Período | Utilização |
|-------------|---------------------|---------|------------|
| SS          | 1,33                | 5       | 20%        |
| $T_1$       | 2                   | 10      | 20%        |
| $T_2$       | 6                   | 14      | 42,9%      |
| Aperiódicas | 1,33                |         |            |



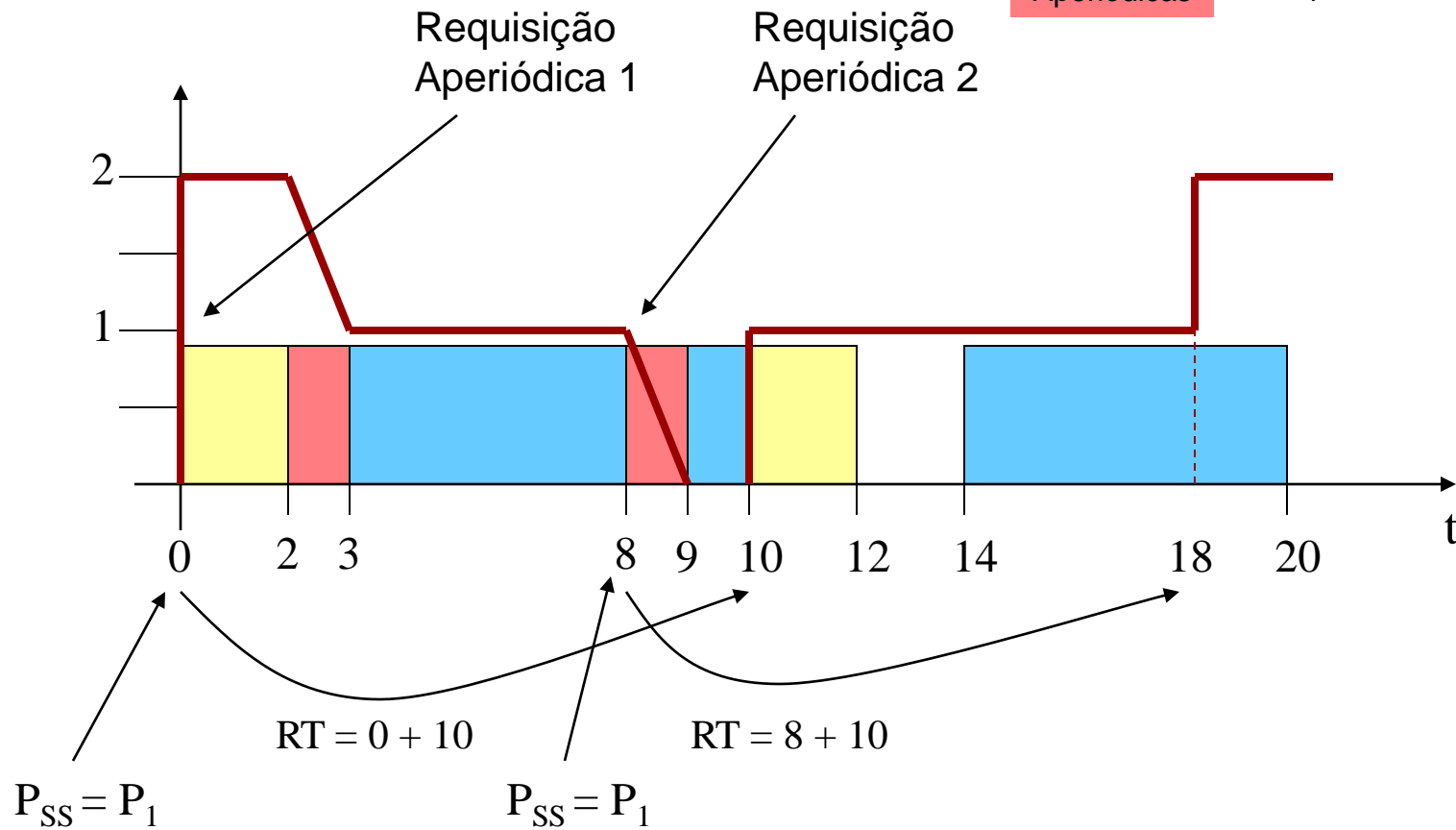
Servidor Esporádico de Alta prioridade

| Tarefa      | Tempo de Computação | Período | Utilização |
|-------------|---------------------|---------|------------|
| $T_1$       | 1                   | 5       | 20%        |
| SS          | 2,5                 | 10      | 25%        |
| $T_2$       | 6                   | 14      | 42,9%      |
| Aperiódicas | 1                   |         |            |



Servidor Esporádico de Média prioridade

| Tarefa      | Tempo de Computação | Período | Utilização |
|-------------|---------------------|---------|------------|
| SS          | 2                   | 10      | 20%        |
| $T_1$       | 2                   | 10      | 20%        |
| $T_2$       | 6                   | 14      | 42,9%      |
| Aperiódicas | 1                   |         |            |



Servidor Esporádico de Igual prioridade

# Servidor Esporádico Dinâmico

---

- Modelo proposto por Spuri e Buttazzo que estende a definição de Servidor Esporádico Clássico;
- Similar aos outros servidores possuindo um período  $P_S$  e um tempo de computação  $C_S$ ;
- O valor do tempo de carregamento não atinge o seu valor máximo a cada período do servidor.



# Servidor Esporádico Dinâmico

---

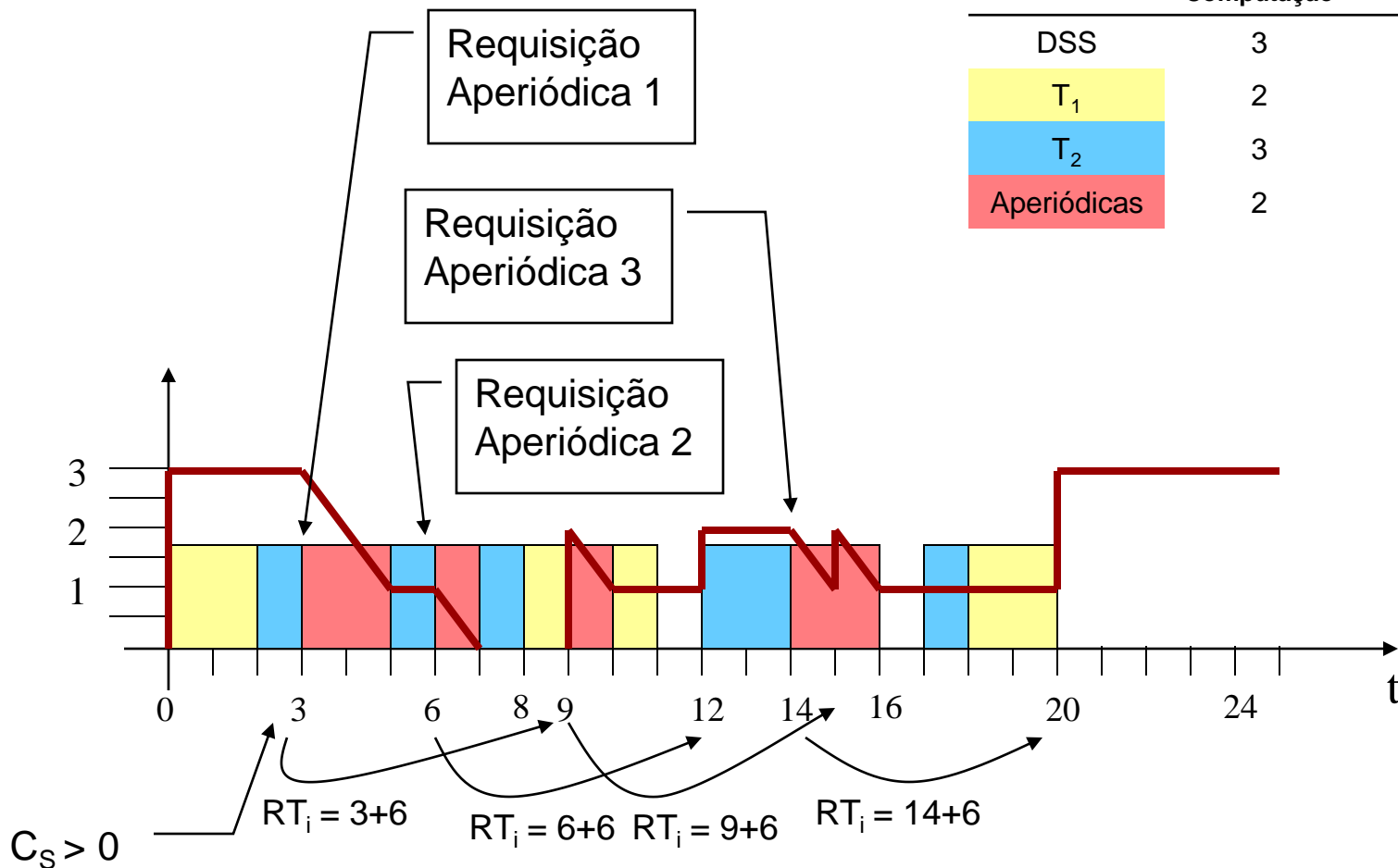
Diferenças entre o Servidor Esporádico Dinâmico (DSS) e o Servidor Esporádico Clássico (SS):

- SS tem sua prioridade baseada na ocorrência do RM enquanto que no DSS tem sua prioridade baseada no deadline.

Regras a seguir para o DSS:

- A capacidade  $C_S$  é inicializada ao máximo quando criado o servidor;
- Regra do  $Rt_i \rightarrow C_S > 0$  e ter uma requisição aperiódica pendente.  $Rt_i = d_S = t_A + T_S$ ;
- Capacidade regarregada é igual a capacidade consumida pela requisição aperiódica.

| Tarefa      | Tempo de Computação | Período | $\rho$ (EDF) |
|-------------|---------------------|---------|--------------|
| DSS         | 3                   | 6       | 1            |
| $T_1$       | 2                   | 8       | 2            |
| $T_2$       | 3                   | 12      | 3            |
| Aperiódicas | 2                   |         |              |



Servidor Esporádico Dinâmico (DSS)

# Escalonabilidade de Servidores Esporádicos

---

- Servidores Esporádicos não se comportam como tarefas periódicas convencionais:
  - Tarefa servidora, mesmo quando pronta, pode não executar devido à ausência de tarefas aperiódicas

# Escalonabilidade de Servidores Esporádicos

---

- Sprunt et al, provou que servidores esporádicos podem ser tratados como tarefas periódicas comuns.

# Escalonabilidade de Servidores Esporádicos

---

- **Lema 1**

- *Dado um conjunto de tarefas periódicas escalonáveis com uma tarefa periódica  $\tau_i$ , o conjunto de tarefas periódicas continua sendo escalonável mesmo que  $\tau_i$  seja dividido em  $k$  tarefas periódicas*

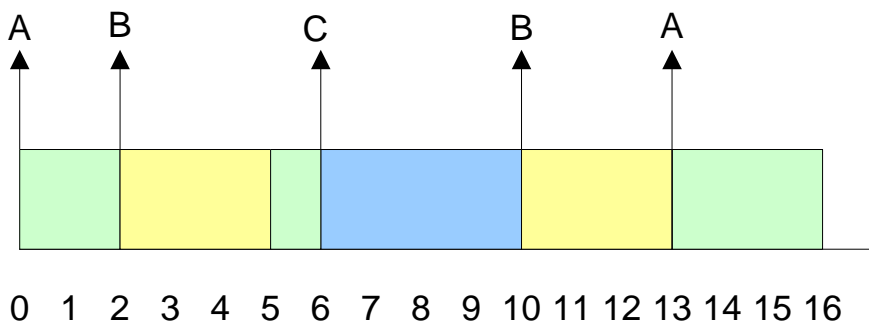
- $\sum_{j=1}^k C_{i,j} = C_i$

- $T_{i,j} = T_i$  para  $1 \leq j \leq k$ .

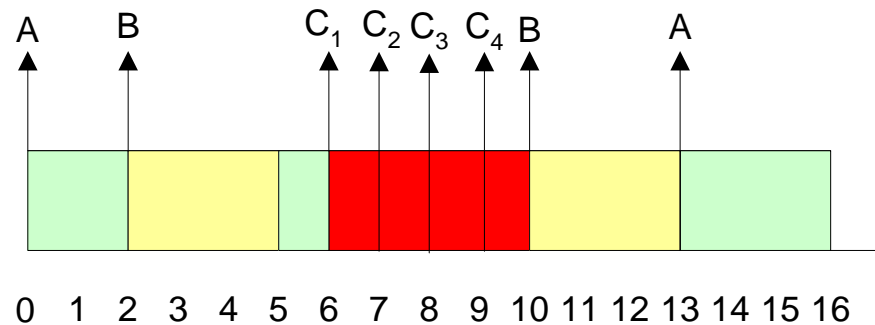
# Escalonabilidade de Servidores Esporádicos

## Demonstração → Lema 1

| Nome                  | Período | Carga Comp. | Prioridade |
|-----------------------|---------|-------------|------------|
| Tarefa A              | 10      | 3           | 2          |
| Tarefa B              | 8       | 3           | 1          |
| Tarefa C              | 15      | 4           | 3          |
| Tarefa C <sub>i</sub> | 15      | 1           | 3          |



(a)



(b)

# Escalonabilidade de Servidores Esporádicos

---

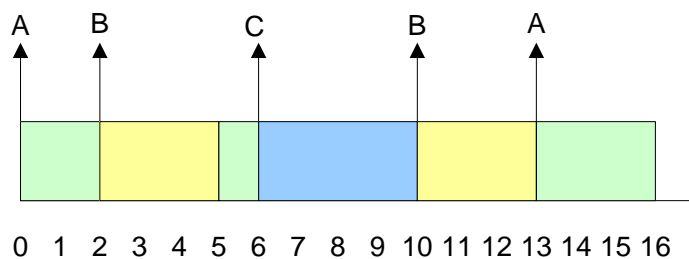
- **Lema 2**

- *Dado um conjunto de tarefas periódicas escalonáveis, substitua uma das tarefas periódicas por um servidor esporádico que possua período e tempo de execução igual ao da tarefa periódica substituída. Se as requisições pelo servidor esporádico ocorrem no mesmo instante de tempo que a tarefa periódica original, então a execução do servidor esporádico possui comportamento idêntico ao da tarefa periódica original.*

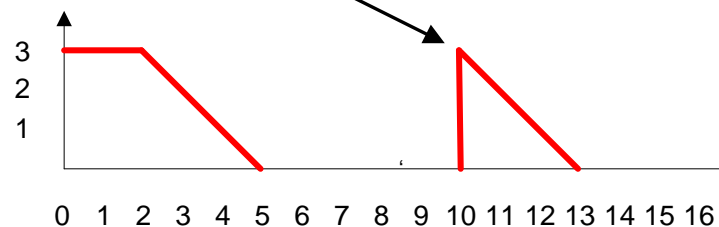
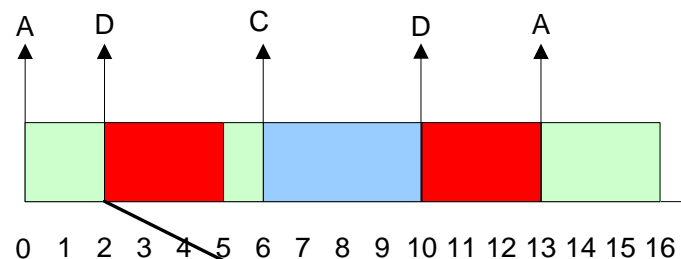
# Escalonabilidade de Servidores Esporádicos

## Demonstração → Lema 2

| Nome                | Período | Carga Comp. | Prioridade |
|---------------------|---------|-------------|------------|
| Tarefa A            | 10      | 3           | 2          |
| Tarefa B            | 8       | 3           | 1          |
| Tarefa C            | 15      | 4           | 3          |
| Tarefa Aperiódica D | -       | 3           | -          |
| Servidor Esporádico | 8       | 3           | 1          |



(a)



(b)



# Escalonabilidade de Servidores Esporádicos

---

- **Teorema**

- *Seja um conjunto de tarefas periódicas escalonáveis com uma tarefa  $\tau_i$ , logo este conjunto também é escalonável caso  $\tau_i$  seja substituído por um servidor esporádico com o mesmo período e tempo de execução.*

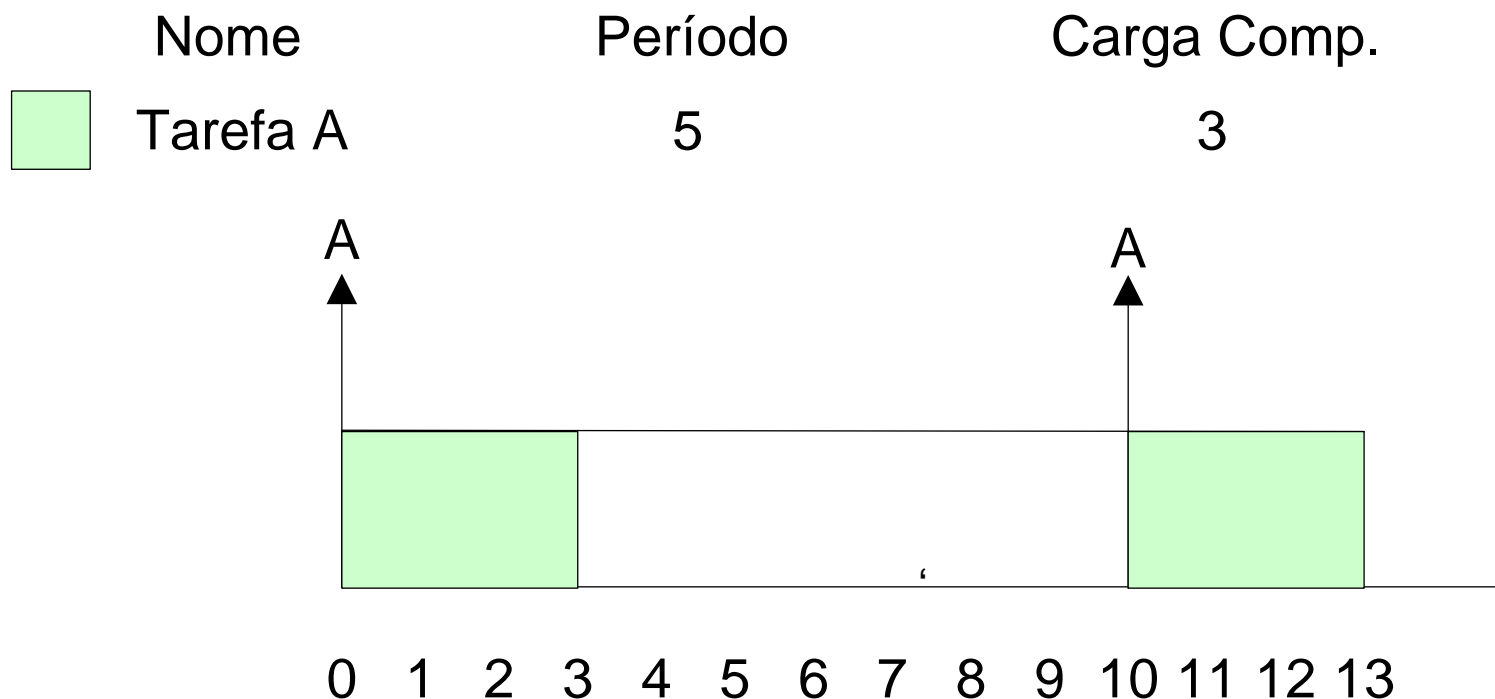
# Escalonabilidade de Servidores Esporádicos

---

- Para a prova do Teorema 2, temos três casos:
  - A carga de execução do servidor não é consumida
  - A carga de execução do servidor esporádico é totalmente consumida
  - A carga de execução do servidor é parcialmente consumida.

# Escalonabilidade de Servidores Esporádicos

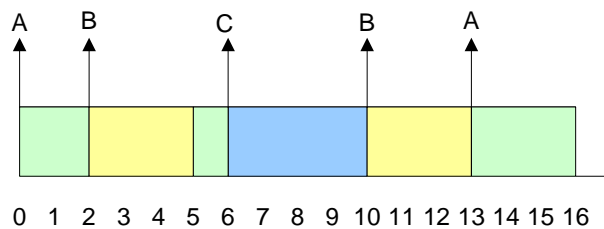
- **1º Caso:** A carga de execução do servidor não é consumida



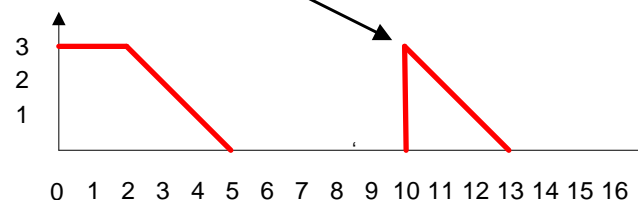
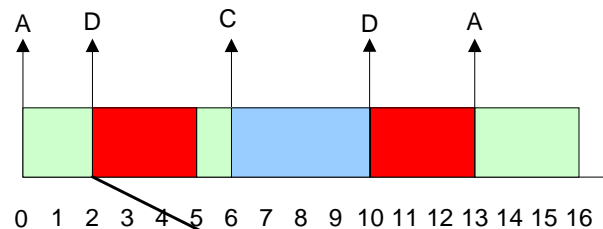
# Escalonabilidade de Servidores Esporádicos

- **2º Caso:** A carga de execução do servidor esporádico é totalmente consumida

| Nome                | Período | Carga Comp. | Prioridade |
|---------------------|---------|-------------|------------|
| Tarefa A            | 10      | 3           | 2          |
| Tarefa B            | 8       | 3           | 1          |
| Tarefa C            | 15      | 4           | 3          |
| Tarefa Aperiódica D | -       | 3           | -          |
| Servidor Esporádico | 8       | 3           | 1          |



(a)



(b)

# Escalonabilidade de Servidores Esporádicos

---

- **3º Caso:** A carga de execução do servidor é parcialmente consumida.
  - Baseado no Lema 1, Sprunt et al mesclaram os dois primeiros casos.

# Performance de Servidores Esporádicos

---

- Realizar o teste do tempo médio de resposta das tarefas aperiódicas.

# Performance de Servidores Esporádicos

---

- Dados relevantes
  - Conjunto de 10 tarefas periódicas com período variando de 54 a 1200
  - Três condições de carga periódica: 40%, 69% e 88%.
  - DS, PE e SS,  $T_{\text{servidor}} = 54$ .
  - Carga das tarefas aperiódicas = valor máximo permitido pelo algoritmo usado.
  - Intervalo de chegada das tarefas aperiódicas variando entre 16, 32 e 54.

# Performance de Servidores Esporádicos

---

- Gráficos



# Performance de Servidores Esporádicos

---

- Resultados relevantes
  - Algoritmos DS, PE e SS obtêm seus melhores resultados em situações onde as tarefas aperiódicas são menores e freqüentes.
  - O aumento gradual da carga aperiódica provoca uma piora acentuada no desempenho do algoritmo DS em relação aos demais algoritmos.
  - O aumento gradual da carga periódica provoca uma leve piora no desempenho do algoritmo SS em relação aos demais algoritmos.

# Escalonamento de tarefas esporádicas

---

- Os servidores esporádicos podem ser adaptados para a realização de escalonamento de tarefas esporádicas.
- Algumas condições devem ser obedecidas:
  - Intervalo mínimo de chegada de duas tarefas esporádicas deve ser maior ou igual ao período da tarefa servidora.
  - Prioridades de execução das tarefas periódicas são estabelecidas obedecendo ao algoritmo rate monotonic
  - A prioridade de execução das tarefas esporádicas é estabelecida obedecendo ao algoritmo deadline monotonic.

# Escalonamento de tarefas esporádicas

---

- Lehoczky, Sha e Ding criaram as seguintes fórmulas para o teste de escalonamento deste tipo de tarefas

$$\forall i \ 1 \leq i \leq n, \quad \frac{C_1}{T_1} + \frac{C_2}{T_2} + \dots + \frac{C_i}{T_i} + \frac{B_i}{T_i} \leq i (2^{1/i} - 1) \quad (1)$$

onde,

$C_i$  = tempo de execução da tarefa  $i$ .

$T_i$  = Período de execução da tarefa  $i$ .

$B_i$  = Pior caso de bloqueio para a tarefa  $\tau_i$ .

# Escalonamento de tarefas esporádicas

- Exemplo de aplicação

| Tarefas  | $C_i$ | $T_i$ | $B_i$ |
|----------|-------|-------|-------|
| $\tau_1$ | 3     | 8     | 4     |
| $\tau_2$ | 3     | 15    | 4     |
| $\tau_3$ | 4     | 20    | 0     |

para  $i = 1$

$$\forall i \ 1 \leq i \leq n, \quad \frac{C_1}{T_1} + \frac{C_2}{T_2} + \dots + \frac{C_i}{T_i} + \frac{B_i}{T_i} \leq i(2^{1/i} - 1)$$

$$\frac{C_1}{T_1} + \frac{B_1}{T_1} \leq 1$$

$$\frac{3}{8} + \frac{4}{8} \leq 1$$

$$\frac{7}{8} \leq 1$$

# Escalonamento de tarefas esporádicas

---

$$\forall i \ 1 \leq i \leq n, \quad \min \left[ \sum_{j=1}^{i-1} U_j \frac{T_j}{lT_k} \left\lceil \frac{lT_k}{T_j} \right\rceil + \frac{C_i}{lT_k} + \frac{B_i}{lT_k} \right] \leq 1 \quad (2)$$

onde,

$C_i$  = tempo de execução da tarefa  $i$ .

$T_i$  = Período de execução da tarefa  $i$ .

$B_i$  = Pior caso de bloqueio para a tarefa  $\tau_i$ .

$U_j$  = Tempo de utilização da tarefa  $\tau_j = C_j / T_j$

$1 \leq k \leq i,$

$l = 1, \dots, \lfloor T_i / T_k \rfloor$

# Escalonamento de tarefas esporádicas

- Exemplo de Aplicação

| Tarefas  | $C_i$ | $T_i$ | $B_i$ |
|----------|-------|-------|-------|
| $\tau_1$ | 3     | 8     | 4     |
| $\tau_2$ | 3     | 15    | 4     |
| $\tau_3$ | 4     | 20    | 0     |

Para  $i = 1$  temos que

$k = 1$ , pois  $1 \leq k \leq i$

$l = 1$ , pois  $l = 1, \dots, \lfloor T_i / T_k \rfloor$

$(k, l) = (1, 1)$ , logo

$$\forall i \ 1 \leq i \leq n, \quad \min \left[ \sum_{j=1}^{i-1} U_j \frac{T_j}{lT_k} \left[ \frac{lT_k}{T_j} \right] + \frac{C_i}{lT_k} + \frac{B_i}{lT_k} \right] \leq 1$$

$$\frac{C_1}{T_1} + \frac{B_1}{T_1} \leq 1$$

$$\frac{3}{8} + \frac{4}{8} \leq 1$$

$$\frac{7}{8} \leq 1$$

# Referências

---

1. Liu, C. L., and Layland, J. W. Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment. JACM 20 (January): 46-61 (1973).
2. Sha, L. Rajkumar, R., and Lehoczky, J.P. Priority Inheritance Protocols: An Approach to Real-Time Synchronization. Technical Report CMU-CS-87-181, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA 15213 (1987).
3. Sprunt, B. Aperiodic Task Scheduling for Hard Real-Time Systems, Ph.D. dissertation, Electrical and Computer Engineering, Carnegie Mellon University, 1990. This thesis is available online as: [http://www.eg.bucknell.edu/~bsprunt/publications/phd\\_thesis/aperiodic\\_task\\_scheduling\\_thesis.pdf](http://www.eg.bucknell.edu/~bsprunt/publications/phd_thesis/aperiodic_task_scheduling_thesis.pdf)
4. Sprunt, B., Sha, L., and Lehoczky, J. P. "Aperiodic task scheduling for hard real-time systems". The Journal of Real-Time Systems, vol. 1, pp. 27-60 (1989).