

Sistemas de Tempo-Real

Aula 8

Escalonamento de tarefas aperiódicas

Execução conjunta de tarefas periódicas e aperiódicas

Utilização de servidores de tarefas aperiódicas

Servidores de prioridades fixas

Servidores de prioridades dinâmicas

Adaptado dos slides desenvolvidos pelo Prof. Doutor Luís Almeida
para a disciplina “Sistemas de Tempo-Real”

Revisto em Nov/2011 por Paulo Pedreiras

Aula anterior (7)

- **Acesso exclusivo** a recursos partilhados: **bloqueio**
- A **inversão de prioridades**: necessidade de a limitar e analisar
- **Técnicas básicas** para acesso exclusivo a recursos partilhados
 - Inibição de interrupções e preempção, semáforos/mutex
- Protocolos avançados
 - **Herança** de prioridades (*Priority Inheritance Protocol – PIP*)
 - Protocolo de **teto de prioridades** (*Priority Ceiling Protocol – PCP*)
 - Protocolo de **pilha de recursos** (*Stack Resource Protocol- SRP*)

Conjugar tarefas periódicas e aperiódicas

Tarefas periódicas

Adequadas, por exemplo, a situações em que é necessário amostrar e processar uma dada grandeza física, normalmente contínua, ou produzir regularmente um dado valor ou atuação

Tarefas aperiódicas

Adequadas quando não se pode prever o respetivo instante de ativação, tal como em alarmes, interface com operadores, ou outros eventos assíncronos.

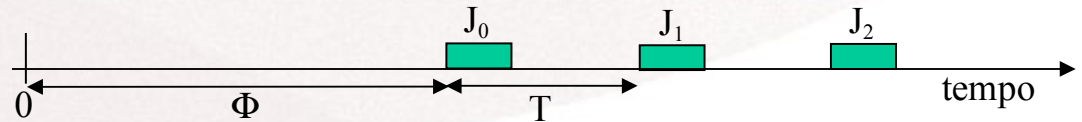
Sistemas híbridos

Aplicações constituídas por tarefas periódicas e aperiódicas.

Muitos sistemas incorporam, de uma forma natural, tarefas dos dois tipos, sendo porventura o caso mais frequente em aplicações reais.

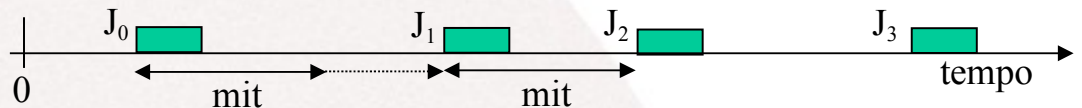
Conjugar tarefas periódicas e aperiódicas

Tarefas periódicas



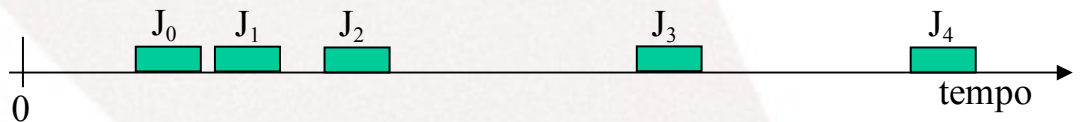
instância n activada em $a_n = n \cdot T + \Phi$ (pior caso bem conhecido)

Tarefas esporádicas



no pior caso é como uma tarefa periódica com período = *mit*

Tarefas aperiódicas



- Só se caracterizam de forma probabilística
- Como **limitar a interferência** sobre as tarefas periódicas?
- Como garantir o **melhor nível de serviço** possível?

Execução em plano de fundo (background)

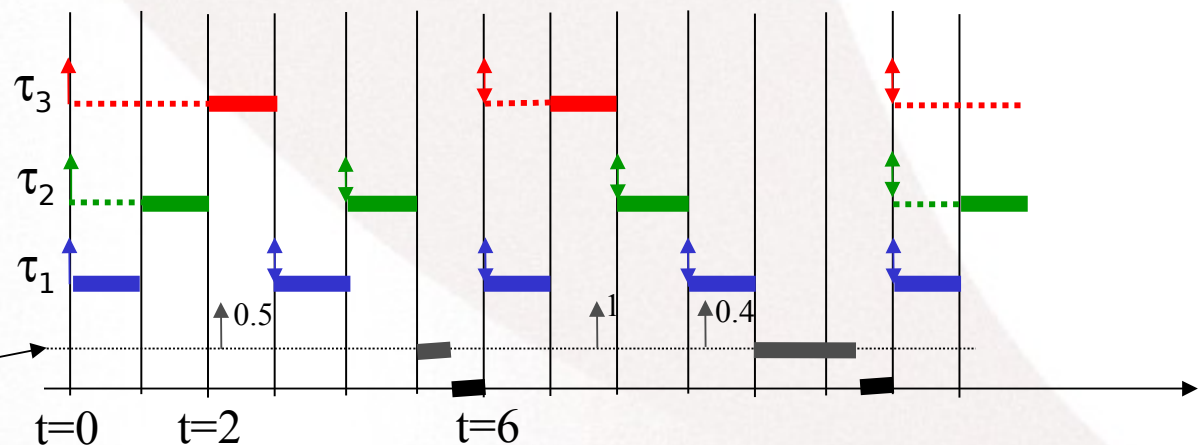
Uma forma comum e simples de combinar ambos os tipos de tarefas consiste em dar **prioridade** absoluta às **tarefas periódicas** e executar as **aperiódicas** apenas nos intervalos de tempo de CPU **deixados livres pelas primeiras**.

Diz-se que as tarefas aperiódicas são **executadas em plano de fundo** ou **background**.

| τ_i | T_i | C_i |
|----------|-------|-------|
| 1 | 3 | 1 |
| 2 | 4 | 1 |
| 3 | 6 | 1 |

Tarefas
periódicas

Plano de fundo
(background)



Execução em plano de fundo (background)

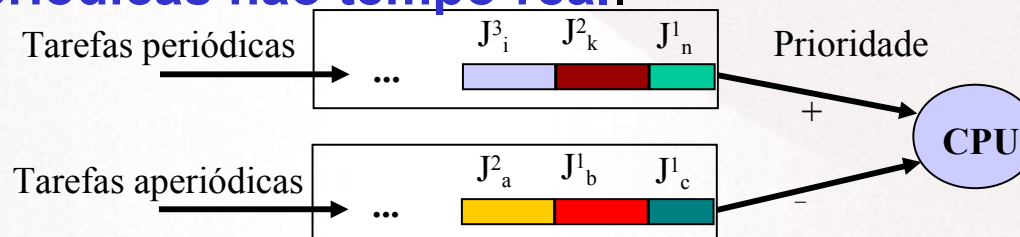
A execução em plano de fundo é muito **fácil de implementar** e **não perturba o subsistema periódico**

- Quando muito poderá haver alguma interferência por via de rotinas de interrupção

Por outro lado, as tarefas aperiódicas poderão sofrer **atrasos grandes** dependendo da carga periódica

- Pode ser calculado considerando as tarefas aperiódicas como de prioridade mais baixa, a seguir à periódica menos prioritária)

Desempenho **fraco para tarefas aperiódicas de tempo-real** mas **adequado para tarefas aperiódicas não tempo-real**.

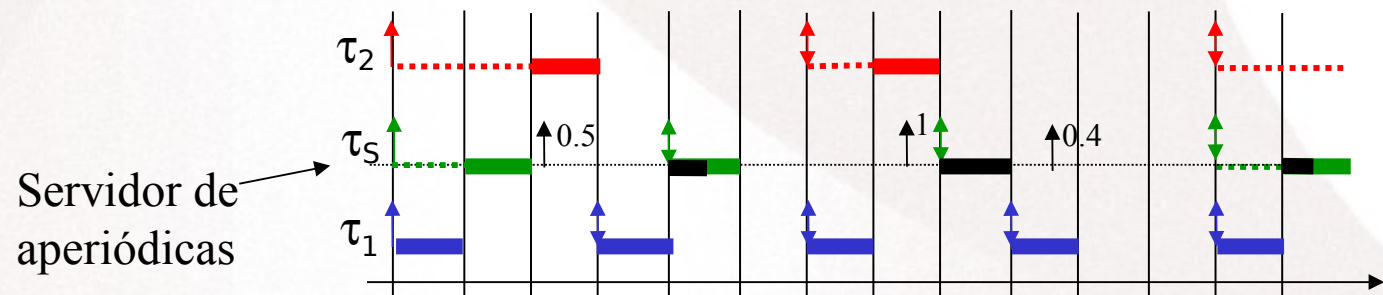


Utilização de servidores de aperiódicas

Quando o serviço em plano de fundo não permite cumprir as restrições de tempo real de tarefas aperiódicas, **o tempo de resposta destas pode ser melhorado** por meio da utilização de uma **tarefa periódica** cuja funcionalidade é apenas a de **executar as tarefas aperiódicas** que estejam ativas

Designa-se essa tarefa por **servidor de aperiódicas** e é caracterizado por um período T_s e uma capacidade C_s

É agora possível inserir o servidor no conjunto das tarefas periódicas e **atribuir-lhe a prioridade necessária** para o nível de serviço desejado.



Utilização de servidores de aperiódicas

- Existem **vários tipos** de servidores de aperiódicas, quer para **prioridades fixas** quer **dinâmicas**, que variam em termos de:
 - **impacto** sobre a escalonabilidade das tarefas periódicas
 - **tempo de resposta médio** aos pedidos de execução das tarefas aperiódicas
 - **custo** (*overhead*) computacional, de memória e de concretização.
- **Prioridades fixas:** Servidor de **sondagem** (*polling*), **adiável** (*deferrable*), de troca de prioridades, **esporádico**,...
- **Prioridades dinâmicas:** Servidores de prioridades fixas adaptados, de **largura de banda total** (TBS) e **constante** (CBS),...

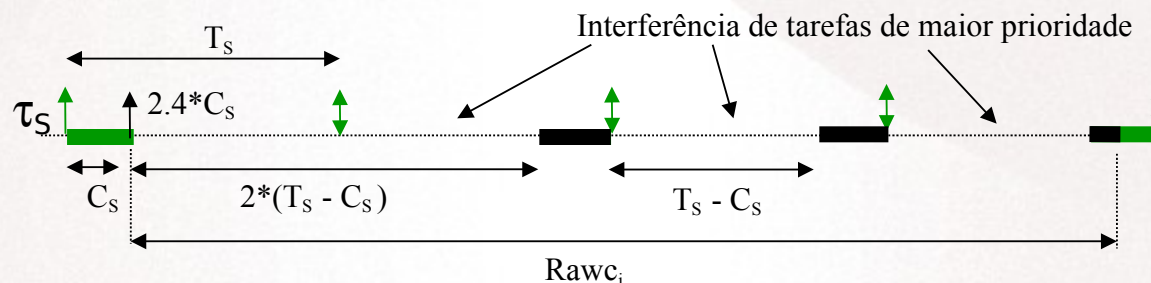
Resposta de pior caso a pedidos aperiódicos

Tempo de resposta de pior caso

- Igual para todos os tipos de servidores que se podem modelar com uma tarefa periódica
- Considera-se que:
 - O servidor é uma tarefa periódica $\tau_s (C_s, T_s)$
 - Sofre máximo *jitter* no instante do pedido aperiódico
 - Sofre máximo atraso em todas as instâncias subsequentes

$$Ra_{wc_i} = Ca_i + (T_s - C_s) * \left(1 + \left\lceil \frac{Ca_i}{C_s} \right\rceil\right)$$

Ca_i = tempo de exec. da
da tarefa aperiódica i



Resposta de pior caso a pedidos aperiódicos

Tempo de resposta de pior caso (cont.)

- Se para o mesmo servidor houver vários (N_a) pedidos aperiódicos em fila de espera (escalonados segundo um determinado critério – coerente com o índice k), o teste de escalonabilidade das tarefas aperiódicas é dado por:

$$\forall i=1..N_a, Ra_{wc_i} = \left(\sum_{k=1}^i Ca_k \right) + (T_s - C_s) * \left(1 + \left\lceil \frac{\sum_{k=1}^i Ca_k}{C_s} \right\rceil \right) \leq Da_i$$

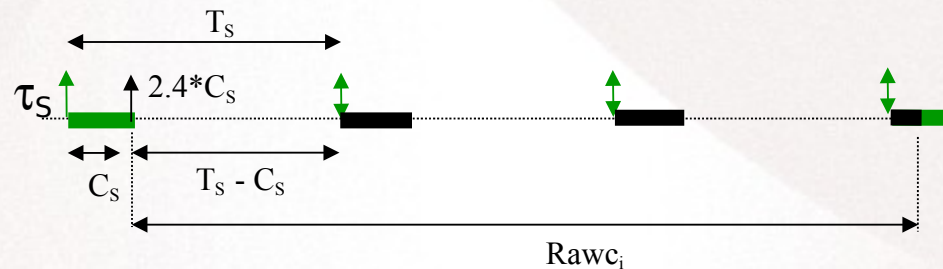
- Considera-se que todos os pedidos aperiódicos são efetuados no mesmo instante (situação de pior caso)

Resposta de pior caso a pedidos aperiódicos

Tempo de resposta de pior caso (cont.)

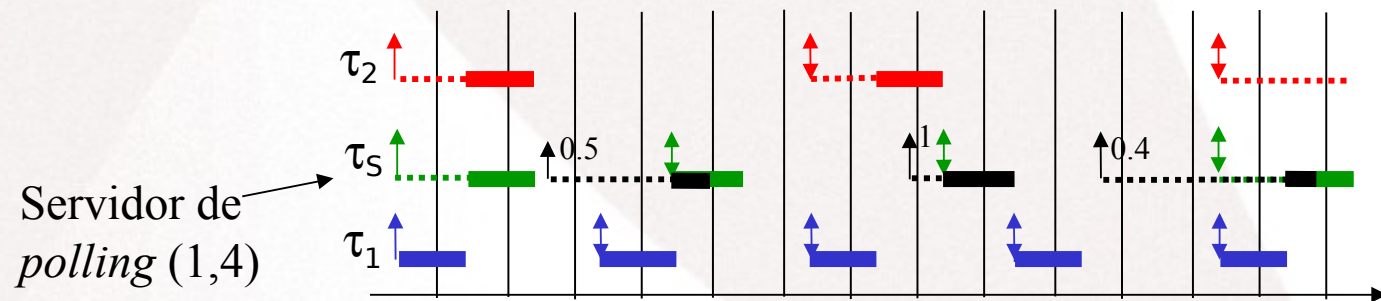
- No caso do servidor ter a máxima prioridade num sistema de prioridades fixas, o atraso das instâncias subsequentes desaparece e o tempo de resposta é:

$$Ra_{wc_i} = Ca_i + (T_s - C_s) * \left\lceil \frac{Ca_i}{C_s} \right\rceil$$



Servidor de sondagem (polling server - PS)

- Este servidor de **prioridades fixas** é completamente equivalente à execução de uma tarefa periódica. Os **pedidos aperiódicos** são servidos apenas **durante os instantes de execução atribuídos ao servidor** pelo escalonador das tarefas periódicas.



Servidor de sondagem (polling server - PS)

- A **realização** de um servidor de *polling* é **relativamente simples**, sendo necessária uma fila para os pedidos aperiódicos e um controlo de capacidade utilizada.
- O **tempo médio de resposta** aos pedidos aperiódicos é **melhorado** em relação à execução em *background* porque é possível executar as tarefas aperiódicas a uma prioridade mais elevada (mas continua a ter muitos períodos de indisponibilidade).
- O **impacto** sobre a escalonabilidade do conjunto das tarefas periódicas é apenas o da tarefa periódica virtual correspondente. Por exemplo, usando RM + PS

$$U_p + U_s \leq (n+1) \left(2^{\frac{1}{n+1}} - 1 \right) \quad (U_p: \text{utilização das } n \text{ tarefas periódicas})$$

Servidor de sondagem (polling server - PS)

- Nota: O menor majorante de Liu & Layland foi determinado de forma independente das taxas de utilização de cada tarefa. Contudo, **fixando a taxa de utilização** da tarefa de **maior prioridade** é possível melhorar (aumentar) o menor majorante.
- Assim, atribuindo a **maior prioridade ao servidor** e taxa de utilização $U_s = C_s/T_s$, o menor majorante de Liu&Layland pode-se expressar como:

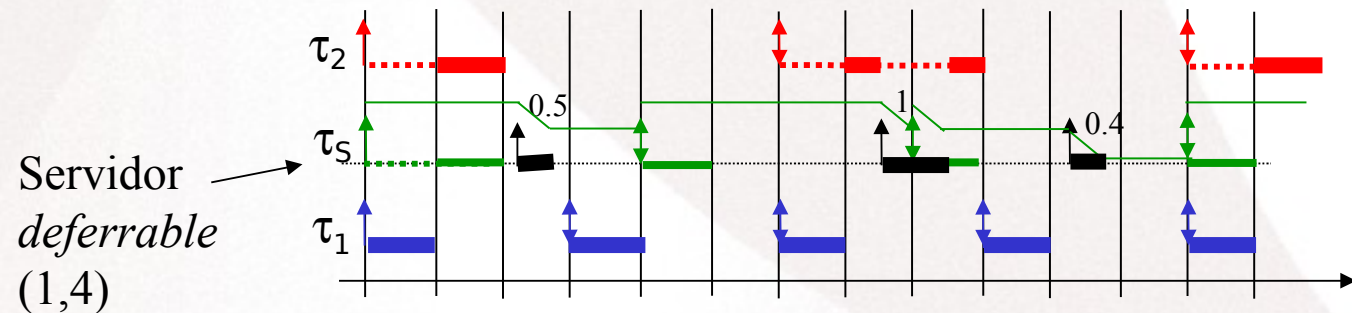
$$U_p + U_s \leq U_s + n \left(\left(\frac{2}{U_s + 1} \right)^{\frac{1}{n}} - 1 \right)$$

- E quando $n \rightarrow \infty$,

$$U_p + U_s \leq U_s + \ln \left(\frac{2}{U_s + 1} \right)$$

Servidor adiável (deferrable server - DS)

- A ideia subjacente a este servidor de **prioridades fixas** consiste em permitir atender pedidos aperiódicos **desde o início** da respetiva execução e até:
 - **final do respetivo período** (T_s) ou
 - que a sua **capacidade** (C_s) **se esgote**.
- A capacidade é reposta no início de cada período.



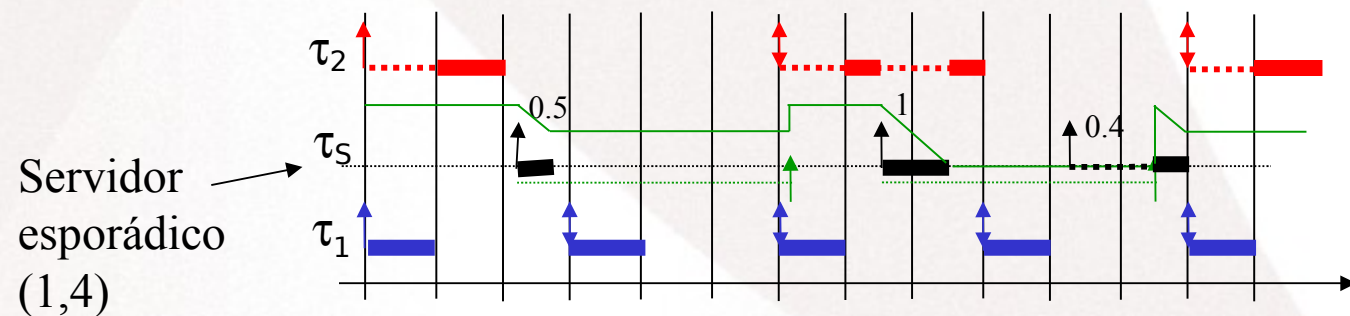
Servidor adiável (*deferrable server* - DS)

- A **complexidade** de realização de um servidor *deferrable* é semelhante à de um PS.
- O **tempo médio de resposta** aos pedidos aperiódicos é **melhorado** em relação ao PS porque é possível utilizar a capacidade do servidor fora dos instantes que lhe estavam inicialmente atribuídos, quando necessário.
- O **impacto** sobre a escalonabilidade do conjunto das tarefas periódicas pode ser negativo, devido ao atraso da sua execução.
 - Usando RM+DS e um servidor com a maior prioridade:

$$U_p + U_s \leq U_s + n \left(\left(\frac{U_s + 2}{2 \cdot U_s + 1} \right)^{\frac{1}{n}} - 1 \right)$$

Servidor esporádico (sporadic server - SS)

- A ideia subjacente a este servidor de **prioridades fixas** é também a de permitir **atender pedidos aperiódicos em qualquer instante** mas **sem penalizar a escalonabilidade** do subsistema periódico.
- Neste caso, a reposição da capacidade gasta não é efetuada periodicamente mas sim de acordo com os instantes em que é efetivamente consumida (instantes de consumo mais T_s)



Servidor esporádico (sporadic server - SS)

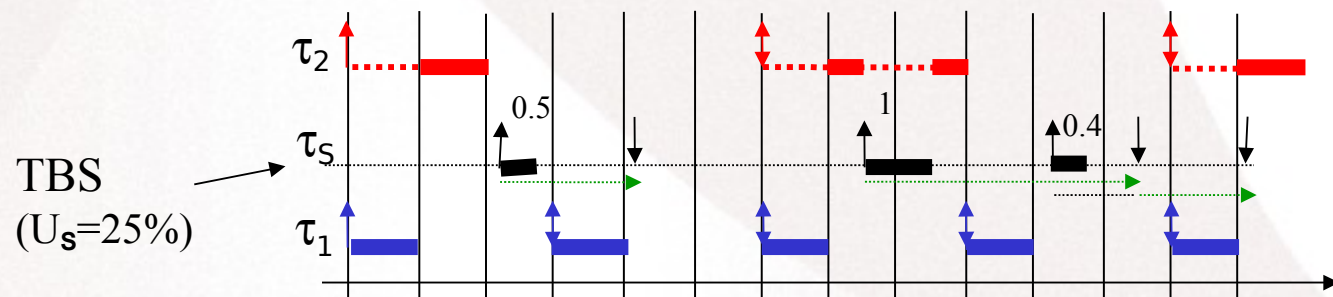
- A **complexidade** de realização de um servidor esporádico é um pouco superior à de um DS devido ao cálculo dos instantes e quantidades de reposição de capacidade.
- O **tempo médio de resposta** aos pedidos aperiódicos é **semelhante** em relação ao DS.
- O **impacto** sobre a escalonabilidade do conjunto das tarefas periódicas é semelhante ao de uma tarefa periódica (tal como o PS).
 - O servidor executa assim que seja ativado e as reativações (através das reposições de capacidade) preservam a largura de banda (ao contrário do DS).
- Usando RM+SS e atribuindo a maior prioridade ao servidor

$$U_p + U_s \leq U_s + n \left(\left(\frac{2}{U_s + 1} \right)^{\frac{1}{n}} - 1 \right)$$

***Servidor de largura de banda total
(total bandwidth server - TBS)***

- O Total Bandwidth Server (TBS) é um servidor de **prioridades dinâmicas** cujo objetivo é o de atender os pedidos aperiódicos num sistema EDF tão cedo quanto possível mas cumprindo a largura de banda que lhe foi atribuída para não perturbar as tarefas periódicas.
- Quando um pedido chega (r_k), é-lhe atribuída uma *deadline* d_k ,

$$d_k = \max(r_k, d_{k-1}) + \frac{C_k}{U_s}$$



Servidor de largura de banda total (total bandwidth server - TBS)

- A **complexidade** de realização de um TBS é pequena, sendo apenas necessário calcular as *deadlines* para cada pedido e depois a respetiva tarefa pode ser inserida na fila de tarefas prontas, juntamente com as tarefas periódicas.
- O **tempo médio de resposta** aos pedidos aperiódicos é **menor** em relação a versões dinâmicas dos servidores de prioridades fixas.
- O **impacto** sobre a escalonabilidade do conjunto das tarefas periódicas é semelhante ao de uma tarefa periódica com a taxa de utilização atribuída ao servidor. Usando EDF+TBS:

$$U_p + U_s \leq 1$$

- Requer **conhecimento prévio de C_k** e é **vulnerável a overruns**.
 - Após o início de execução, a tarefa aperiódica pode executar mais do que o declarado

Servidor de largura de banda constante (constant bandwidth server - CBS)

- O Constant Bandwidth Server (CBS) é um servidor de **prioridades dinâmicas** cujo objetivo é o de resolver o problema de robustez do TBS efetuando **isolamento de largura de banda**.
- Isso é conseguido através de uma gestão do tempo de execução usando um esquema de **gestão de capacidade** (Q_s, T_s).

- Quando chega um pedido r_k calcula-se a *deadline* do servidor d_s :

$$\text{se } r_k + \frac{c_s}{U_s} < d_s^{\text{actual}}, \text{ então } d_s^{\text{actual}} \text{ mantém-se}$$

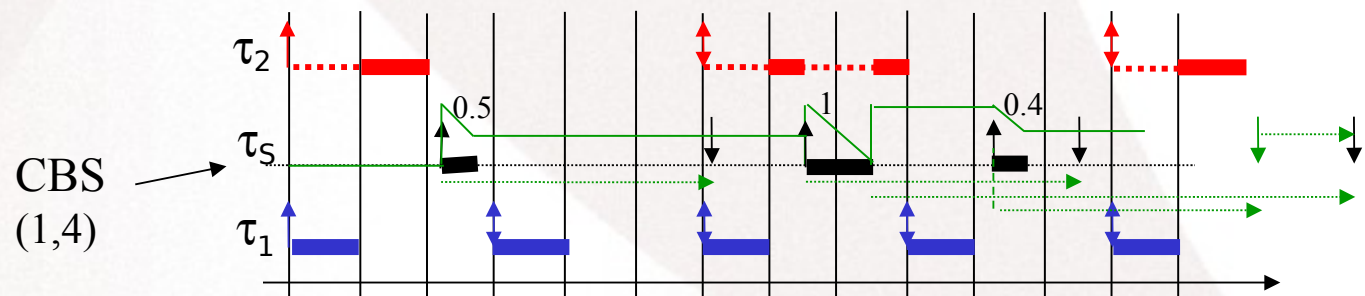
$$\text{senão } d_s = r_k + T_s, \text{ e } c_s = Q_s$$

- Quando se esgota a capacidade instantânea (c_s) repõe-se a capacidade e adia-se d_s :

$$d_s = d_s + T_s, c_s = Q_s$$

***Servidor de largura de banda constante
(constant bandwidth server - CBS)***

- O CBS atribui uma *deadline* que força a largura de banda atribuída ao servidor a nunca ultrapassar a que lhe foi atribuída.
- Se uma tarefa executar mais tempo do que esperado, a sua *deadline* é automaticamente adiada, baixando a prioridade da tarefa (é como se se aumentasse dinamicamente o período da tarefa)



Servidor de largura de banda constante (constant bandwidth server - CBS)

- A **complexidade** de realização de um CBS é um pouco superior à do TBS devido à gestão da capacidade. As tarefas aperiódicas são igualmente inseridas na fila de tarefas prontas.
- O **tempo médio de resposta** aos pedidos aperiódicos é **semelhante** ao do TBS.
- O **impacto** sobre a escalonabilidade do conjunto das tarefas periódicas é semelhante ao de uma tarefa periódica com a taxa de utilização atribuída ao servidor. Usando EDF+CBS

$$U_p + U_s \leq 1$$

Servidor de largura de banda constante (constant bandwidth server - CBS)

- O grande interesse na utilização do CBS reside no **isolamento de largura de banda**.
- Se uma tarefa é servida por um CBS com largura de banda U_s , em qualquer intervalo Δt essa tarefa nunca requererá mais de $\Delta t * U_s$ tempo de CPU.
- Qualquer tarefa $\tau_i (C_i, T_i)$ escalonável em EDF é também escalonável dentro de um CBS com $Q_s = C_i$ e $T_s = T_i$
- **Um CBS pode ser utilizado para:**
 - Proteger o sistema de possíveis **overruns** em quaisquer tarefas
 - Garantir um **serviço mínimo** a tarefas *soft real-time*
 - **Reservar largura de banda** para qualquer atividade

Resumo da Aula 8

- Execução conjunta de tarefas **periódicas e aperiódicas**
 - Execução de aperiódicas em *background*
- Utilização de **servidores de tarefas aperiódicas**
 - Servidores de **prioridades fixas**
 - Polling Server - PS
 - Deferrable Server - DS
 - Sporadic Server - SS
 - Servidores de **prioridades dinâmicas**
 - Total Bandwidth Server – TBS
 - Constant Bandwidth Server - CBS