Especificação, Modelação e Projecto de Sistemas Embutidos

Conceitos básicos de Tempo-Real

Paulo Pedreiras

pbrp@ua.pt

Departamento de Electrónica, Telecomunicações e Informática
Universidade de Aveiro

Parcialmente baseado nos materiais pedagógicos da disciplina de "Sistemas Tempo-Real", Luís Almeida, DETI, Universidade de Aveiro

V1.1 Outubro/2009

Definição

- **Computação** de Tempo-Real:
  - Os resultados das computações devem ser:
    - **Logicamente correctos** e **Produzidos a tempo**

Pontualidade  Correcção lógica
(Stankovic, 1988)

- **Funcionalidade ou serviço** de Tempo-Real
  - **Funcionalidade ou serviço** que tem de ser desempenhada ou prestado dentro de **intervalos de tempo finitos** impostos por um **processo físico**
- **Sistema** de Tempo-Real
  - Aquele que desempenha **pelo menos uma funcionalidade** de tempo-real ou que presta **pelo menos um serviço de tempo-real** (PDC, 1990)

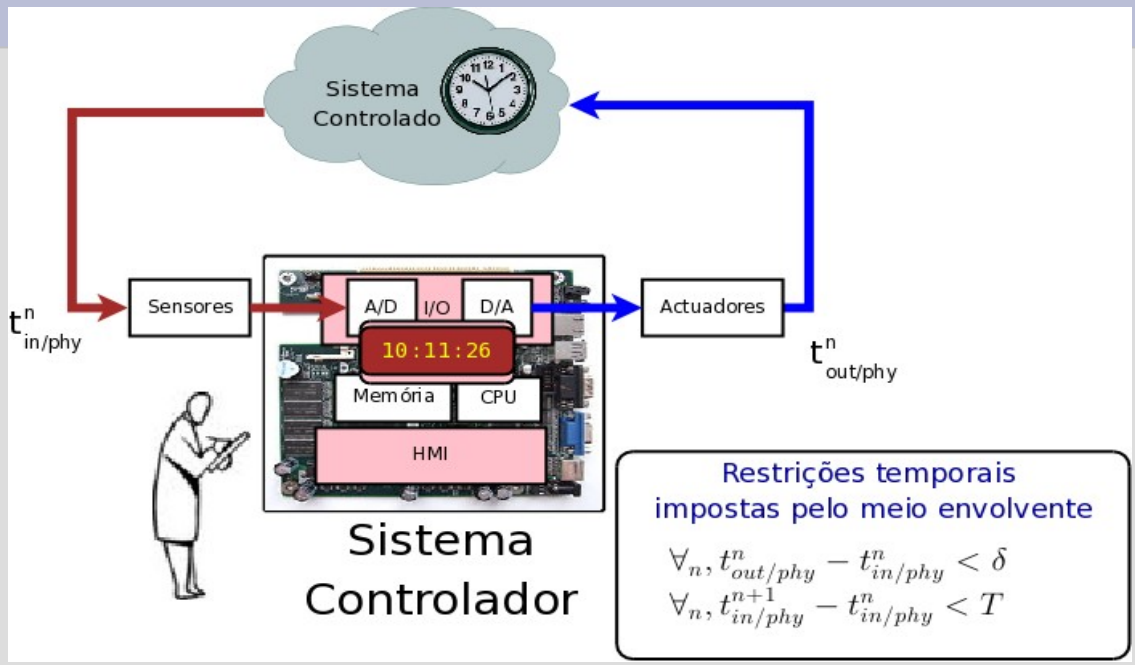
Requisitos temporais

Origem dos requisitos temporais:

- Normalmente advêm da **dinâmica** do **processo físico** que se pretende controlar
- Impõem **restrições** aos **instantes** em que as acções desempenhada num sistema são **executadas**.
  - No sistema de travagem de um automóvel não é suficiente dizer que após carregar-se no respectivo pedal se desencadeia a travagem! É necessário **garantir** que tal acontece no **tempo correcto** (e.g. ordem não pode demora mais de 50ms a ser executada).
- Estas restrições têm de ser cumpridas em **todas as instâncias** (incluindo o pior caso) e não apenas em termos médios

Exemplo de ES com requisitos temporais

- Exemplo de ES: controlo de um processo.



Algoritmo de controlo

Configurar um Timer para gerar uma **interrupção** em **intervalos** de **T**;
Em **cada interrupção** do Timer
do
Conversão A/D dos **parâmetros** relevantes do sistema
e.g. temperatura, pressão, humidade, ...
Cálculo do sinal de controlo
Escrita do sinal de controlo nos **actuadores** (e.g. válvula a 30%)
end do

Exemplo de ES com requisitos temporais

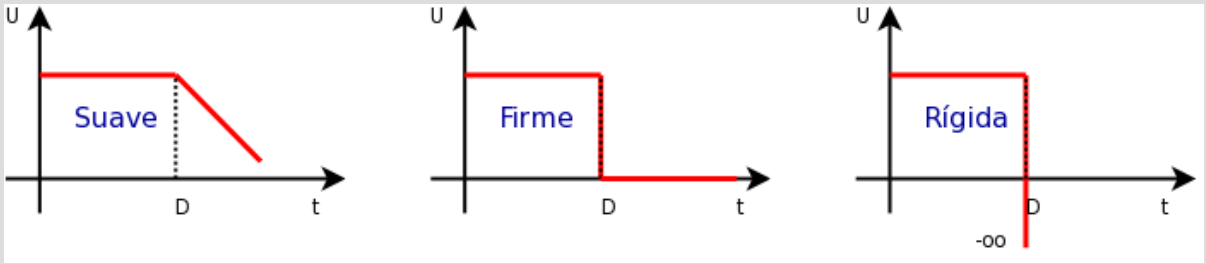
- Há muitos outros SE que possuem requisitos de tempo-real.
- E.g. **Sistemas multimédia**
  - A *Set-top box* recebe, para **cada *frame*** (imagem), vídeo e áudio comprimido em f**ormato digital** (e.g. MPEG-2)
  - A **taxa** de recepção de *frames* é e.g. **25 frames/sec**
    - i.e. são mostradas 25 imagens distintas em cada segundo
  - Cada *frame* tem de ser **processada** em 1/25=40ms
    - Em paralelo pode ter de ser efectuada desencriptação, recuperação de erros, ...
    - Quando tal não acontece pode acontecer uma degradação da qualidade

Muitas outras classes de SE possuem requisitos de tempo-real: sist. transporte, controlo de tráfego, equip. médico, equip./sistemas militares, automação industrial, equip. telecomunicações, ...

Classificação das restrições temporais

Classificação das restrições temporais

- De acordo com a utilidade do resultado para a aplicação podem classificar-se em:
  - ***Suave (Soft) -*** Restrição temporal em que o resultado que a ela está associado mantém **alguma utilidade** para a aplicação mesmo depois de um limite D embora haja uma degradação da qualidade de serviço.
  - ***Firme (Firm) -*** Restrição temporal em que o resultado que a ela está associado **perde qualquer utilidade** para a aplicação depois de um limite D.
  - ***Rígida (Hard) -*** Restrição temporal que, quando não cumprida, pode originar uma **falha catastrófica.**



Classificação dos sistemas tempo-real

Classificação do Sistemas de Tempo-Real:

- De acordo com o tipo das restrições temporais os STR podem classificar-se em:
  - **Soft Real-Time**
    - O sistema apenas apresenta restrições temporais do tipo ***firm* ou *soft*** (e.g., simuladores, sistemas multimédia)
  - **Hard Real-Time**
    - O sistema apresenta **pelo menos uma** restrição temporal do tipo ***hard***. São sistemas de segurança crítica (e.g. controlo de voo de aviões, de mísseis, de centrais nucleares, de fábricas de produtos perigosos)

Classificação das tarefas

Os ES são habitualmente estruturados como um **conjunto de tarefas concorrentes** que executam funções específicas.

Tipos de tarefas:

- **Periódicas**
  - *Time-driven*, activadas regularmente em intervalos fixos
  - Tipicamente especificadas por $\{C_i , T_i , D_i\}$
    - C_i = tempo de execução de pior caso
    - T_i = período da tarefa
    - D_i = *deadline* (relativa) da tarefa

Exemplo: amostragem periódica de um sensor

Classificação das tarefas

- **Tarefas esporádicas**
  - *Event-driven*, **activadas** por uma **entidade externa** ou **alteração** no ambiente
  - Tipicamente especificadas por $\{C_i, mit_i, D_i\}$
    - $\{C_i, D_i\}$ significado idêntico ao anterior
    - mit_i representa o tempo mínimo entre activações

      Exemplo: detecção de passagem de um objectos numa linha de montagem

- **Tarefas aperiódicas**
  - *Event-driven*, activadas por uma entidade externa ou alteração no ambiente
  - Chegada de **eventos** com **propriedades desconhecidas** (múltiplos eventos, eventualmente simultâneos ou muito próximos)

Normalmente não é possível garantir o cumprimento de restrições temporais deste tipo de tarefas! Porquê?

Conceitos básicos de escalonamento

O **tempo** que uma tarefa demora a **terminar** depende:

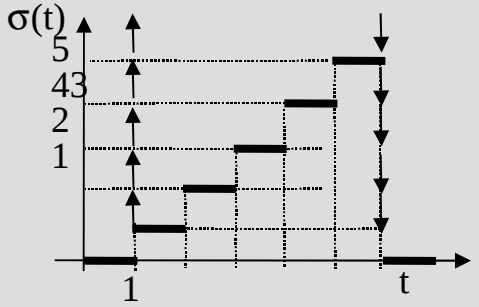
- **Tempo de execução** (próprio)
  - Tempo de execução das instruções que compõem a tarefa
  - Depende da complexidade da tarefa
  - É em geral difícil de estimar, sendo influenciado por factores como:
    - Estrutura do código (linguagem, condicionais, ciclos)
    - DMA, caches, pipeline
    - Sistema operativo ou *kernel* (*system calls*)
- **Interferência**
  - Tempo de espera motivado pela execução de tarefas com maior prioridade – **Depende do algoritmo de escalonamento**
- **Bloqueio**
  - Execução de tarefas de menor prioridade
  - E.g. devido a acesso a recursos partilhados (*buses*, discos, portos de comunicação,...)

Noção de escalonamento

Noção de **escalonamento** de tarefas:

- **Dados**:
  - um conjunto de tarefas
  - restrições que lhe estão associadas (ou função de custo)
- Encontrar uma **atribuição de tempo de processador às tarefas** que lhes permita :
  - executar as **tarefas completamente**
  - **cumprir as suas restrições** (ou minimizar a função de custo)

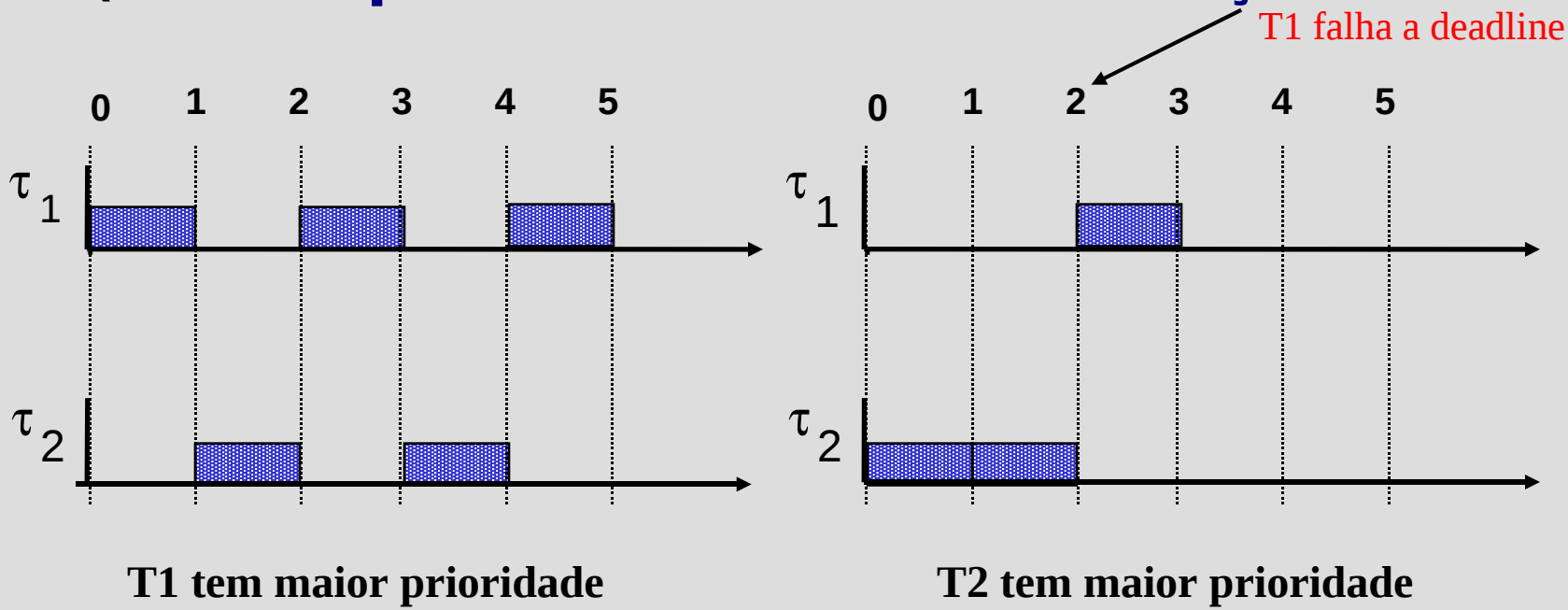
e.g. $J = \{J_i\ (C_i=1, a_i=1, D_i=5, i=1..5)\}$ ->



Noção de escalonamento

Ilustração do **problema** de escalonamento

- Considere as duas tarefas seguintes, **activadas sincronamente** no instante t=0:
  - T1 (1,2,2)
  - T2 (2,5,5)
- Qual o **impacto da ordem de execução**?

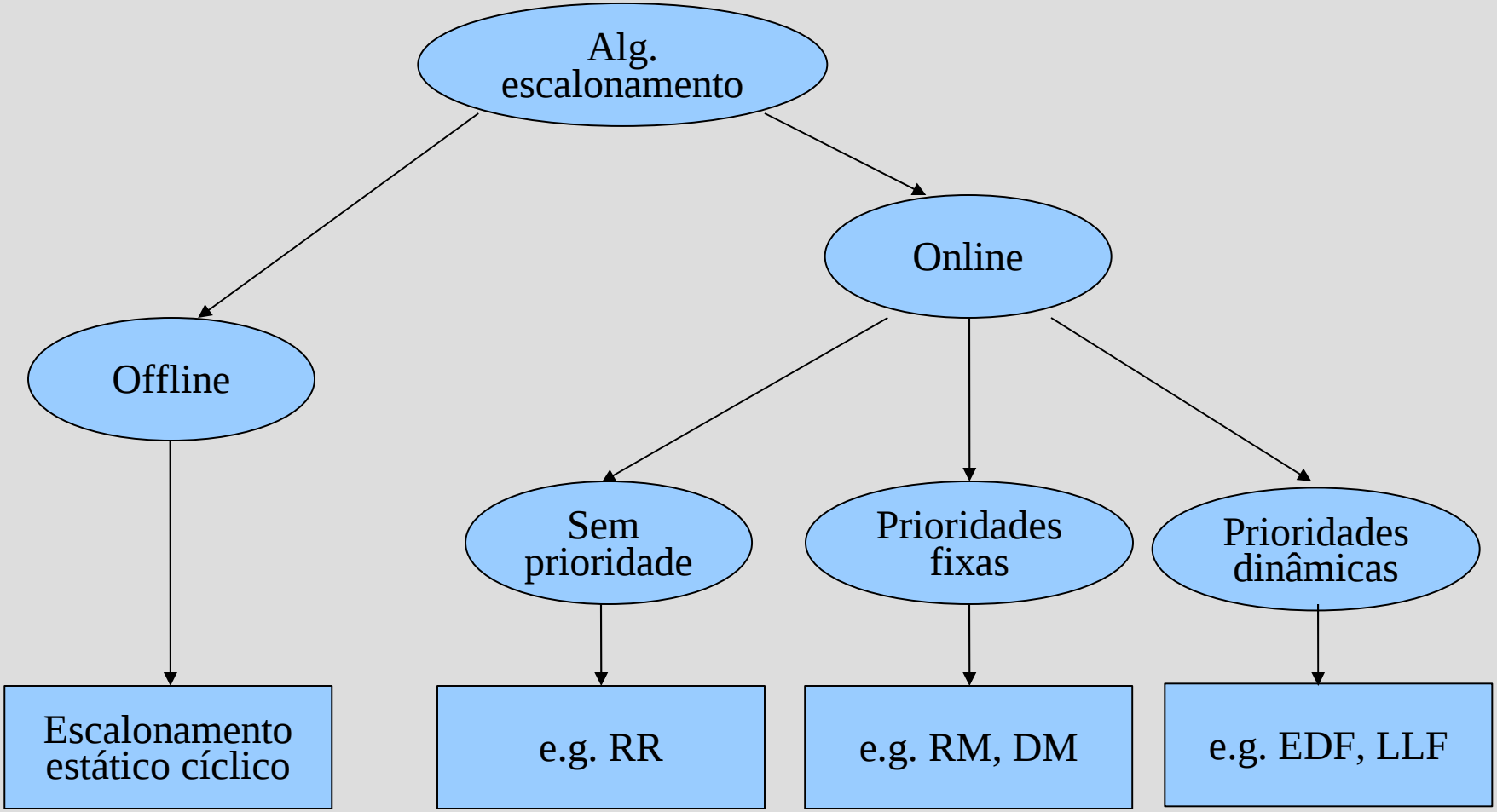


T1 tem maior prioridade

T2 tem maior prioridade

Taxonomia dos algoritmos de escalonamento

Taxonomia dos algoritmos de escalonamento



Algoritmos de escalonamento

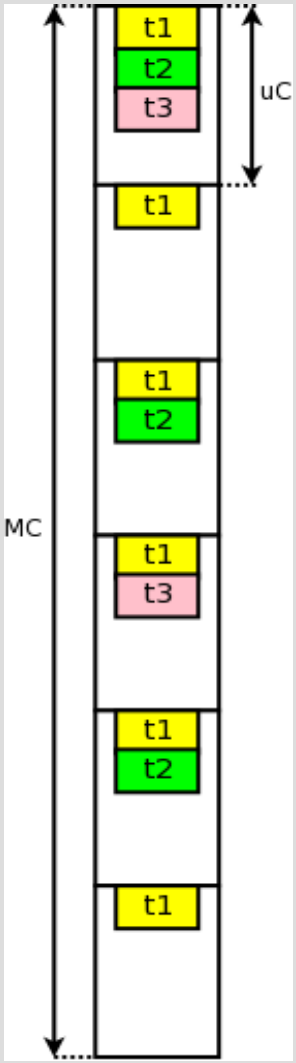
Escalonamento **estático cíclico**

- A tabela é organizada em **micro-ciclos** (uC) de **duração fixa** para que, quando varrida, se obtenha o carácter periódico das tarefas.
- Os micro-ciclos são **disparados** por um ***timer***.
- O varrimento contínuo da tabela resulta num **padrão cíclico global** chamado **macro-ciclo** (MC)

$$\Gamma = \{ \tau_i (C_i, \Phi_i, T_i, D_i, \ i=1..n)\}$$

uC = MDC(T_i) (*GCD*)
MC = MMC(T_i) (*LCM*)

E.g.
$\Phi_i = 0$,
$C_i = 1ms$,
$T_1 = 5ms$
$T_2 = 10ms$
$T_3 = 15ms$



Algoritmos de escalonamento

Algoritmos sem prioridade (exemplos):

- First-In-First-Out (**FIFO**) ou First-Come-First-Served (**FCFS**)
  - Tarefas prontas são inseridas numa **lista**
  - As tarefas **são despachadas** por **ordem de chegada**
  - Não há preempção
  - Não existe o conceito de prioridade

- **Round-Robin** (Preemptivo)
  - As tarefas prontas são despachadas "à vez"
  - Cada tarefa recebe periodicamente um **parte fixa equitativa** (*fair*) do tempo de CPU
  - A tarefa em execução é **suspensa** (*preempted*) no **final** de cada intervalo de execução
  - Não existe o conceito de prioridade

Algoritmos de escalonamento

Algoritmos baseados em **prioridades fixas**

- Cada tarefa recebe um nível de **prioridade específico**
- Algumas tarefas têm **maior importância** que outras
- O nível de prioridade **não** pode ser **alterado** em *run-time*

- **Prioridades arbitrárias**
  - Alocadas pelo projectista do sistema de acordo com um critério arbitrário (e.g. importância)

- **Rate Monotonic**
  - Período mais curto -> maior prioridade
  - Tarefas mais “rápidas” escalonadas em primeiro lugar

- **Deadline Monotonic**
  - *Deadline* (relativa) mais curta -> maior prioridade
  - As tarefas com prazos de execução mais “apertados” são escalonadas em primeiro lugar

Algoritmos de escalonamento

Algoritmos baseados em **prioridades dinâmicas**

- Algumas tarefas têm **maior importância** que outras
- O nível de prioridade **pode** ser **alterado** em *run-time*

- **Earliest Deadline First (EDF)**
  - As tarefas prontas recebem uma prioridade inversamente proporcional à distância à *deadline* absoluta respectiva.
- **Least Slack Time (LST)**
  - Maior prioridade às tarefas com menor tempo livre (*slack*, *laxity*)
- **Shortest Completion Time (SCT)**
  - Tarefas com menor tempo de computação restante são escalonadas em primeiro lugar

Análise de escalonabilidade

Análise de escalonabilidade:

- Em diversos sistemas é necessário saber à priori se é possível **garantir o cumprimento dos requisitos temporais** de um certo conjunto de tarefas, ou seja, se este é escalonável
- Metodologias para determinação da escalonabilidade
  - **Baseados em utilização**
    - Computacionalmente mais simples (complexidade O(n))
    - Menos rigoroso (e.g. para prioridades fixas é apenas suficiente)
  - **Baseados em tempo de respostas**
    - Computacionalmente mais complexo
    - Condição necessária e suficiente quer para prioridades fixas e dinâmicas (preempção, tarefas independentes e *release* síncrono). Fornece o tempo de resposta de cada tarefa.

Sumário

- **Definição** de tempo-real
- **Requisitos** temporais e **classificação** dos sistemas
- **Tipos** de tarefas e sua **caracterização**
- **Escalonamento**
  - **Noções** básicas, **Taxonomia**
  - Alguns **algoritmos** de escalonamento
    - Estático cíclico,
    - Sem prioridades (FIFO, RR)
    - Com prioridades fixas (RM, DM)
    - Prioridades dinâmicas (EDF, LST, SCF)