

# Sistemas em Tempo Real

Jadsonlee da Silva Sá

[Jadsonlee.sa@univasf.edu.br](mailto:Jadsonlee.sa@univasf.edu.br)

[www.univasf.edu.br/~jadsonlee.sa](http://www.univasf.edu.br/~jadsonlee.sa)

# Introdução

## ✓ Sistemas em Tempo Real (STR).

- Sistemas onde o funcionamento correto depende não apenas dos resultados lógicos (**correção lógica - *correctness***), mas também do instante (**correção temporal - *timeliness***) em que estes resultados são produzidos.
- Sistemas computacionais que devem reagir a estímulos oriundos do seu ambiente em prazos específicos.

# Exemplos de Aplicação

- ✓ Controladores embarcados em eletrodomésticos.



# Exemplos de Aplicação

- ✓ Sistemas de controle de plantas industriais.



# Exemplos de Aplicação

✓ Controle de tráfego aéreo e ferroviário.



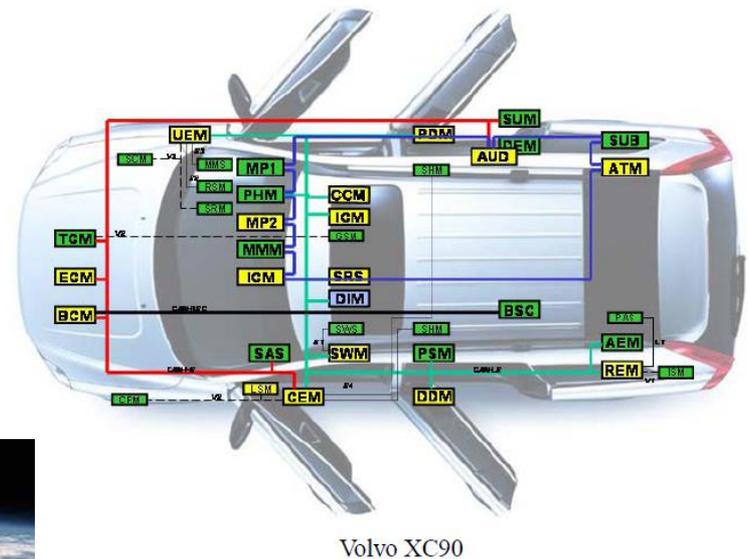
# Exemplos de Aplicação

- ✓ Monitoramento de pacientes em hospitais.



# Exemplos de Aplicação

- ✓ Sistemas embarcados em robôs e veículos.



# Exemplos de Aplicação

✓ Aplicações multimídia entre outras.



# Definições sobre Tempo

✓ Correção temporal - *Timeliness*.

- Tempo denso → Contínuo no tempo - conjunto dos reais.
- Tempo discreto → Conjunto dos naturais positivos.
- Tempo global → Tempo único em qualquer parte do sistema - Sistemas distribuídos.
- Tempo local → Observável localmente nos diferentes nós de um sistema distribuído.
- Tempo absoluto → Tempo com referência estabelecida a partir de um evento global.
- Tempo relativo → Referência dada por um evento local.

# Previsibilidade nos STRs

- ✓ Um STR é **previsível** quando o seu comportamento pode ser antecipado antes de sua execução, independentemente de variações ocorrendo à nível de *hardware (clock drifts)*, da carga e de falhas.
- ✓ Algumas hipóteses sobre o comportamento do ambiente externo são necessárias para poder prever a evolução de um STR e garantir dentro de certos limites as suas restrições temporais:
  - Qual a carga máxima computacional?
  - Quais os tipos e frequência de falhas que o sistema terá que conviver durante a execução?

# Previsibilidade nos STRs

- ✓ É necessário conhecer o comportamento do sistema, considerando o cenário de pior caso e as hipóteses de falhas.
- ✓ Essas hipóteses devem ser definidas de forma realista - Em muitos sistemas, esta não é uma tarefa simples.
- ✓ A garantia de previsibilidade depende também de outros fatores ligados ao sistema:
  - Arquitetura do *hardware*;
  - Sistema operacional (caso exista);
  - Linguagem de programação.

# Previsibilidade nos STRs

- ✓ Um sistema é previsível quando é possível antecipar que todos os prazos atribuídos a partir das interações com o seu ambiente serão cumpridos.
- ✓ Além da previsibilidade determinística, é possível ter também uma previsibilidade probabilística.
  - Indicam a probabilidade de um requisito de tempo ser satisfeito.
  - Alguns eventos ocorrem de modo aleatório.

# Classificação dos STRs

✓ Do ponto de vista de segurança:

- Sistemas de tempo real **não-crítico** (*soft real-time*):
  - As conseqüências de uma falha temporal são da mesma ordem de grandeza dos benefícios normais do sistema.
  - Ex.: sistema de comutação telefônico, alguns sistemas de processamento bancário entre outros.
- Sistemas de tempo real **crítico** (*hard real-time*).
  - As conseqüências de uma falha temporal excedem em muito os benefícios normais do sistema. A falha é catastrófica.
  - Ex.: sistema de controle de vôo e de plantas nucleares, entre outros.

# Tipos de STRs

- ✓ Do ponto de vista computacional, as atividades em um STR podem ser executadas:
  - Uma única Unidade de Processamento (UP) (sistema uniprocessador);
  - Em várias UPs acopladas por meio de uma memória compartilhada (sistema multiprocessador) ou via rede de comunicações (sistema distribuído em tempo real).

# Tarefas de Tempo Real

- ✓ STRs são normalmente modelados por um conjunto de tarefas de tempo real.
- ✓ O comportamento das tarefas são caracterizados pelas suas restrições temporais (atributos), relações de precedência e de exclusão.
- ✓ Modelo de tarefas.

# Tarefas de Tempo Real

- ✓ **Definição** → É um *thread* independente de execução que pode competir pelo uso do processador e outros recursos e que possui restrições de tempo.
  - Ex.: trecho de código que periodicamente realiza a conversão analógico digital de um sinal proveniente de um sensor e armazena-o em uma variável.
- ✓ Uma **tarefa** consiste de um número "infinito" de requisições (**instâncias**).

# Tarefas de Tempo Real

## ✓ Restrições Temporais.

- Toda tarefa possui um prazo para ser finalizada - Deadline.
- As conseqüências de uma tarefa ser finalizada após o seu *deadline* definem dois tipos de tarefas.
  - **Tarefas críticas (*hard*)** - Falhas catastróficas ocorrerão se a tarefa for finalizada após o seu *deadline*.
  - **Tarefas não-críticas (*soft*)** - Haverá no máximo uma perda de desempenho do sistema.

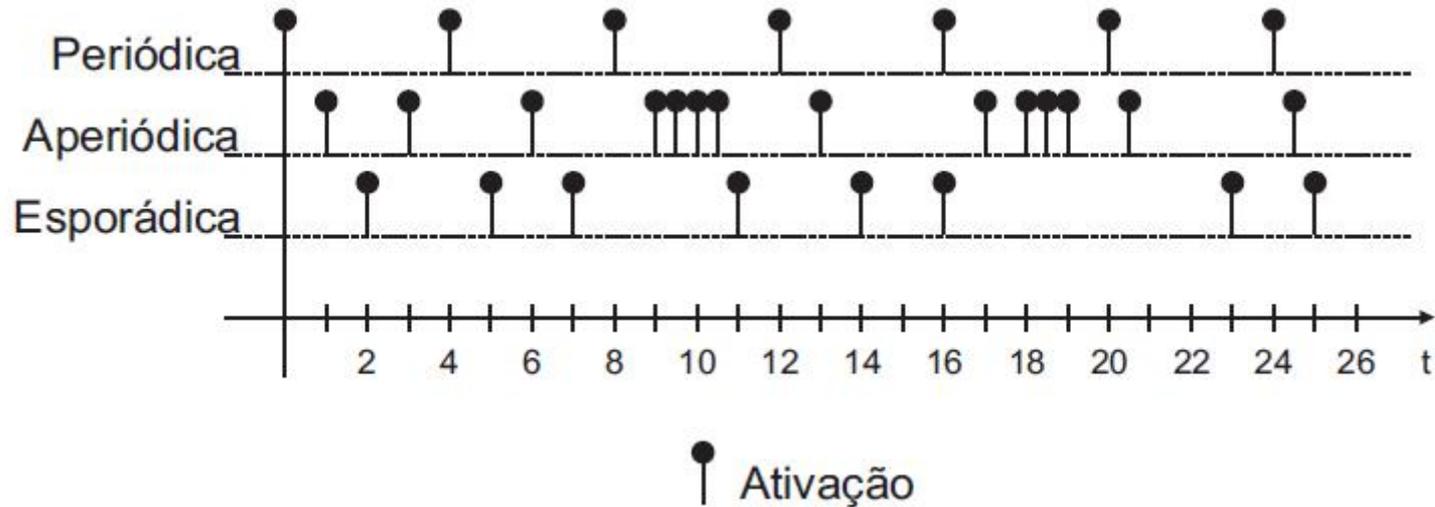
# Tarefas de Tempo Real

## ✓ Restrições Temporais.

- As tarefas podem ser classificadas de acordo com a sua frequência de **ativação**:
  - **Periódicas** → Possuem um intervalo de tempo constante entre sucessivas ativações - Período.
  - **Aperiódicas** → Possuem um intervalo de tempo desconhecido entre sucessivas ativações.
  - **Esporádicas** → São um tipo de tarefas aperiódicas, mas são caracterizadas pelo conhecimento de um intervalo mínimo entre sucessivas ativações.

# Tarefas de Tempo Real

## ✓ Restrições Temporais.



# Tarefas de Tempo Real

## ✓ Restrições Temporais.

- Uma tarefa é ativada e depois liberada (pronta para execução).
- Instante de ativação e instante de liberação.
- A primeira instância de uma tarefa é ativada após um tempo chamado de offset.
  - Offset estático → O instante de ativação é fixado.
  - Offset dinâmico → O instante de ativação varia entre 0 e um limite superior conhecido.
- Após o *offset*, as sucessivas ativações ocorrem de acordo com a frequência de ativação da tarefa.

# Tarefas de Tempo Real

## ✓ Restrições Temporais.

- **Instante de liberação** → Pode coincidir ou não com o instante de ativação.
- A liberação de uma tarefa pode ser atrasada por vários motivos:
  - *Polling* de um escalonador ativado por tempo;
  - Espera de um evento externo: chegada de uma mensagem via rede; finalização de uma outra tarefa entre outros.
- **Jitter de liberação** → Diferença de tempo entre os instantes de liberação e ativação da instância de uma tarefa.

# Tarefas de Tempo Real

## ✓ Restrições Temporais.

- Tempo de computação → Tempo necessário para executar a instância de uma tarefa.
  - Este tempo é normalmente constante ou variante de forma limitada.

# Tarefas de Tempo Real

## ✓ Comportamento Temporal de uma Tarefa Periódica.

- Uma tarefa periódica  $\tau_i$  pode ser descrita pela quintupla  $(T_i, O_i, C_i, J_i, D_i)$ .

