

---

# Conceitos Básicos dos Sistemas de Tempo Real

**Fundamentos dos Sistemas de Tempo Real**

Rômulo Silva de Oliveira

Edição do Autor, 2018

[www.romulosilvadeoliveira.eng.br/livrotemporeal](http://www.romulosilvadeoliveira.eng.br/livrotemporeal)

Outubro/2018



- Sistemas computacionais com requisitos de tempo real:
  - Submetidos a requisitos de natureza temporal não triviais
  - Resultados devem estar corretos lógica e temporalmente
  - Requisitos temporais são definidos pelo ambiente físico
- Sistema deve ser **construído para atender os requisitos temporais**
- Podem existir vários requisitos temporais simultâneos
  - Várias malhas de controle realimentado, teclado, rede, etc
  - Junto com outras atividades que não são de tempo real mas precisam ser realizadas com desempenho razoável
- É necessário entender o sistema para ter segurança de que os requisitos temporais serão cumpridos
- Para isto são usados **modelos** que descrevem como a execução do software do sistema acontece

- **Tarefa ( task )**
  - Segmento de código cuja execução possui atributo temporal próprio (período, deadline, etc)
  - Exemplo: método em OO, função C, trecho de um programa
  
- **Deadline**
  - Instante máximo desejado para a conclusão de uma tarefa
  - É o requisito temporal mais relevante na análise de um sistema
  
- **Deadline relativo**
  - Em relação ao início da tarefa (fazer em 10ms)
  
- **Deadline absoluto**
  - Em relação a UTC (fazer até as 10:00:00)

- Tempo real crítico ( **hard real-time** )
  - Falha temporal pode resultar em consequências catastróficas
  - Necessário garantir requisitos temporais em projeto
  - Exemplo: usina nuclear, indústria petroquímica, mísseis
  
- Tempo real não crítico ( **soft real-time** )
  - Requisito temporal descreve apenas comportamento desejado
  - Exemplo: multimídia, videogame

- **Deadline Hard**

- Perda do deadline pode ter consequências catastróficas
- Exemplo: abrir válvula em duto de alta pressão

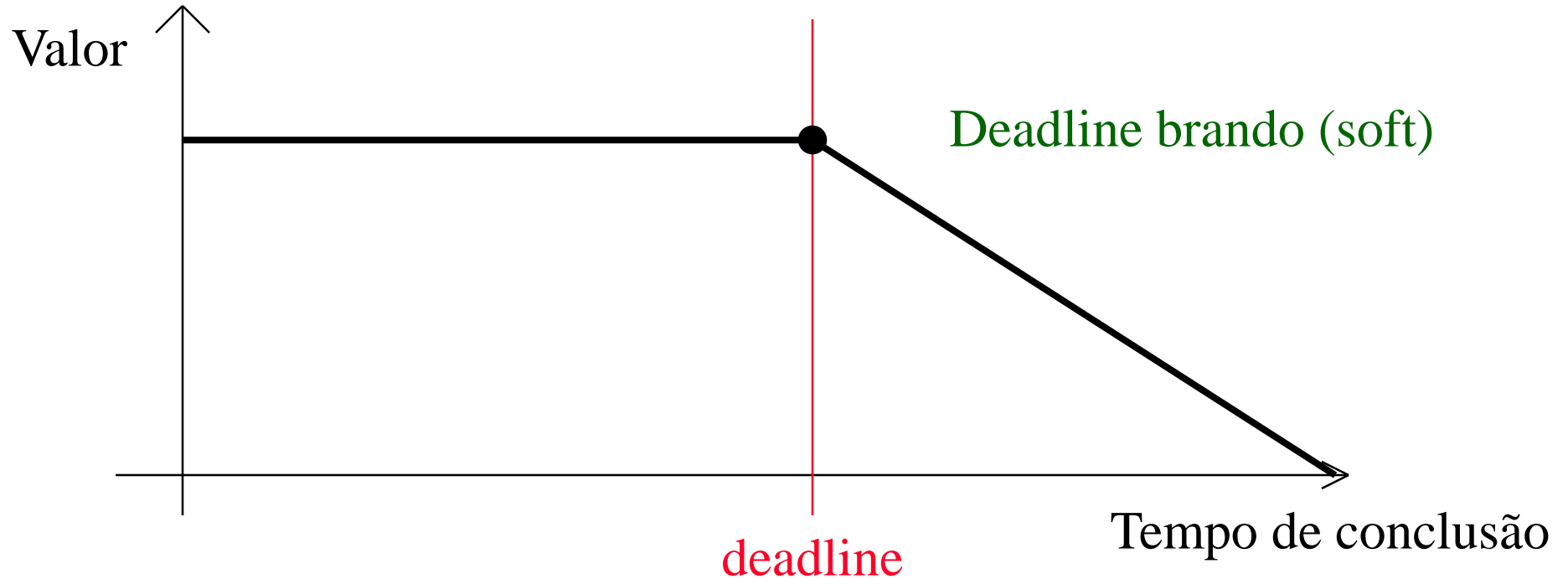
- **Deadline Firm**

- Perda do deadline NÃO tem consequências catastróficas
- Não existe valor em terminar a tarefa após o deadline
- Exemplo: amostrar periodicamente valor físico

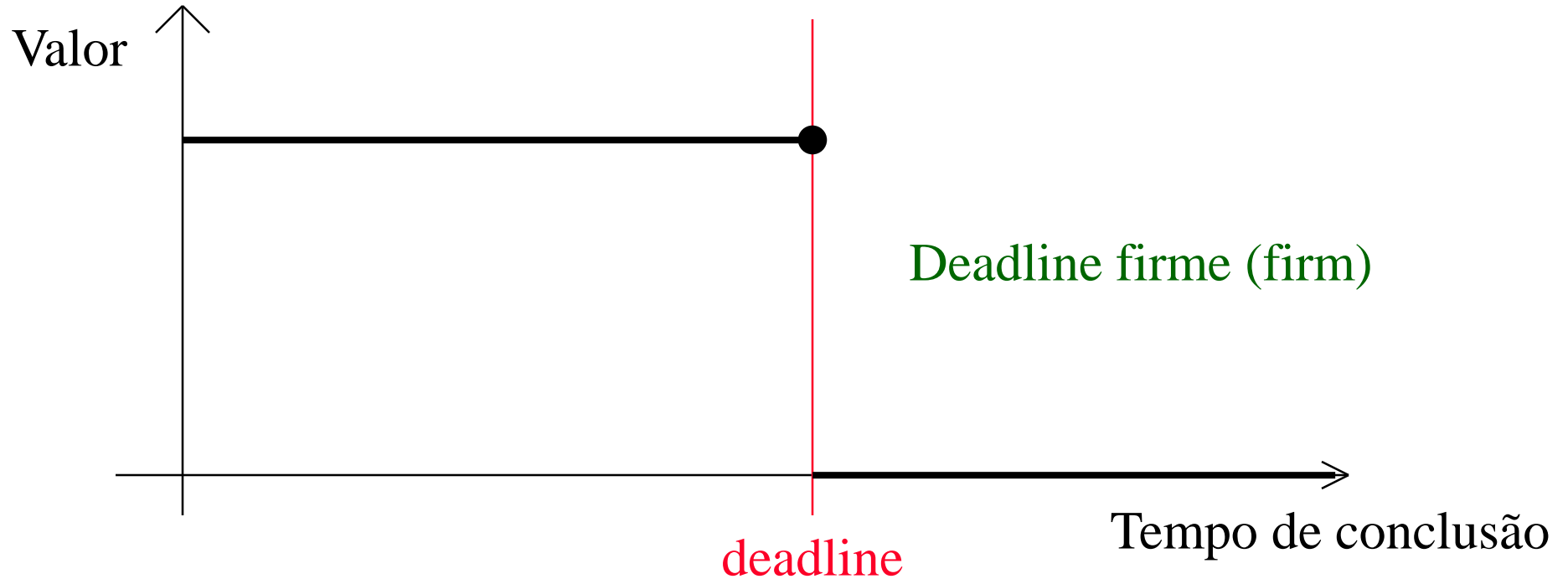
- **Deadline Soft**

- Perda do deadline NÃO tem consequências catastróficas
- Existe valor em terminar a tarefa com atraso
- Exemplo: movimento de objeto em vídeo game

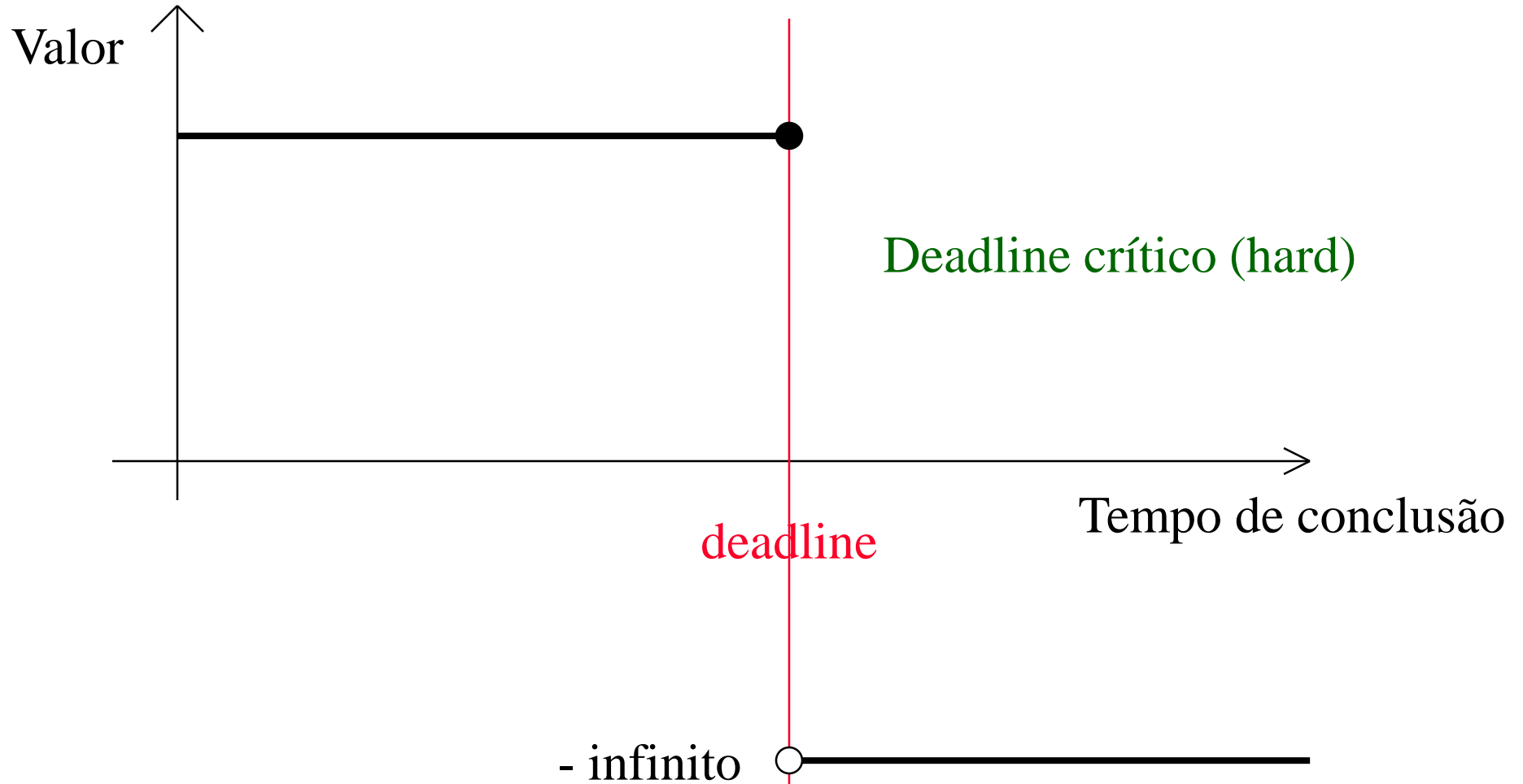
## Tipos de Deadlines 2/4



## Tipos de Deadlines 3/4



## Tipos de Deadlines 4/4





- **Previsibilidade** (*predictability*)
  - Está associada a capacidade de poder antecipar, em tempo de projeto, se os processamentos em um sistema de tempo real serão executados dentro de seus prazos especificados
- Associada a uma **previsão determinista**
  - todos os deadlines serão respeitados
- ou a uma **antecipação probabilista**
  - baseadas em estimativas, probabilidades são associadas a deadlines definindo as possibilidades dos mesmos serem respeitados
- A necessidade de previsibilidade gera implicações em todos os níveis:
  - linguagens
  - sistemas operacionais
  - comunicação
  - arquitetura do computador
  - etc

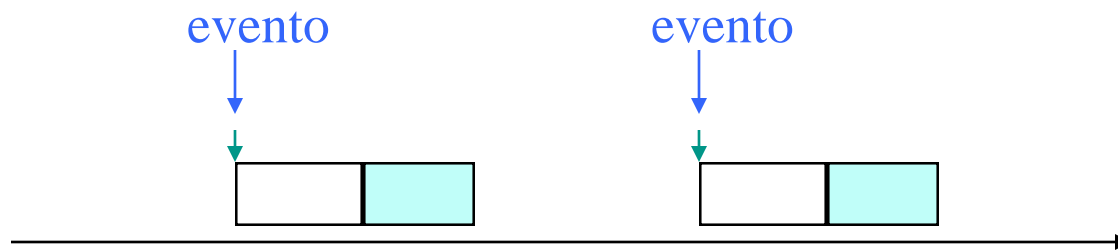
# Terminologia do Escalonamento

---

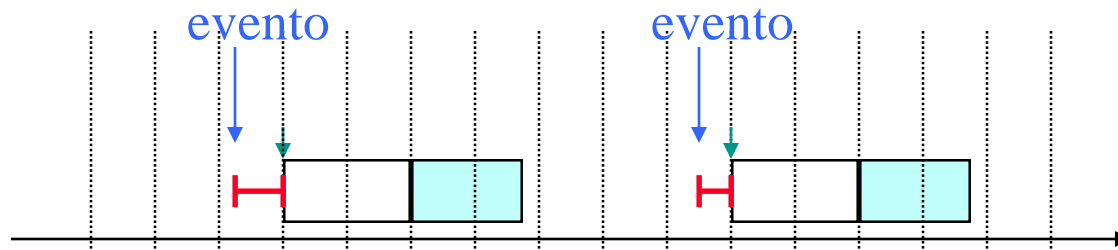
- **Escalonamento** ( scheduling )
  - Identifica a forma como recursos são alocados às tarefas
- **Escalonador** ( scheduler )
  - Componente do sistema responsável pela gerência dos recursos
- **Escala de Execução** ( schedule )
  - Descreve quando cada tarefa ocupa cada recurso
  
- **Escalonamento Estático** ( static scheduling )
  - Utiliza apenas informações disponíveis antes da execução
- **Escalonamento Dinâmico** ( dynamic scheduling )
  - Utiliza informações disponíveis apenas durante a execução

# Event-Triggered x Time-Triggered 1/2

- Sistema dirigido por eventos (*Event-Triggered*)



- Sistema dirigido por tempo (*Time-Triggered*)



## Event-Triggered x Time-Triggered 2/2

---

- **Dirigido por eventos** (*Event-Triggered*)
  - Sistema reage a eventos
  - Evento externo gera interrupção e dispara tarefa
  - Pior caso: todos os eventos acontecem ao mesmo tempo
  - Sistema flexível
  - Determinismo de deadline
  
- **Dirigido por tempo** (*Time-Triggered*)
  - Interrupção de relógio a cada T milisegundos (tick)
  - A cada tick algumas tarefas são executadas
  - Sensores e atuadores são acessados em momentos pré-definidos
  - Não existem interrupções além das do relógio
  - Sistema rígido
  - Determinismo de execução

- **Tarefa Periódica**

- Tarefa é ativada a cada  $P$  unidades de tempo
- Instantes de chegada podem ser calculados a partir do inicial
- Exemplo: controle de processo via laço de realimentação

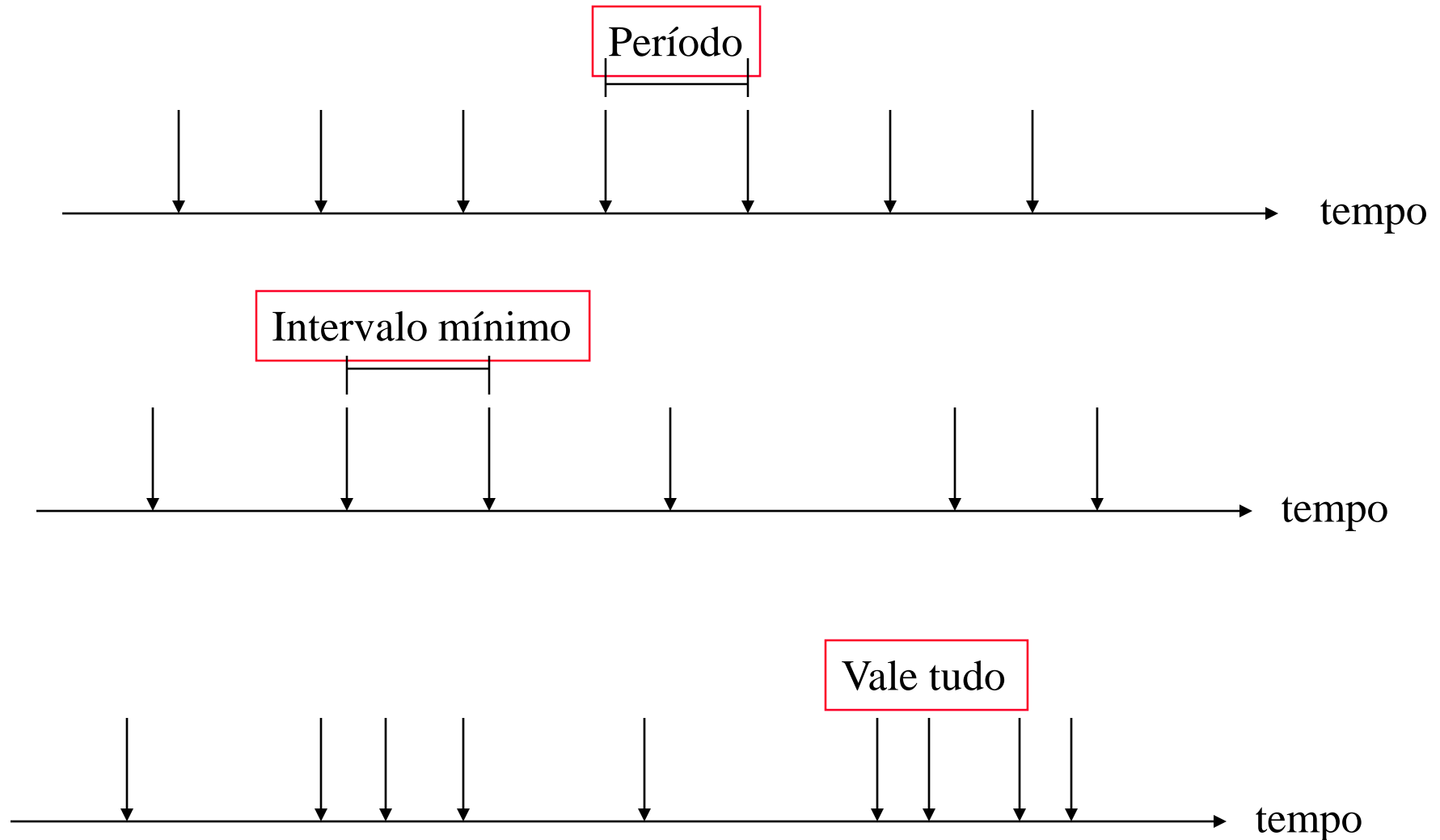
- **Tarefa Esporádica**

- Instantes de chegada não são conhecidos
- Existe um intervalo mínimo de tempo entre chegadas
- Exemplo: atendimento a botão de alarme

- **Tarefa Aperiódica**

- Nada é sabido quanto as ativações da tarefa
- Exemplo: aparecimento de objeto em tela de radar

## Tipos de Recorrência 2/2



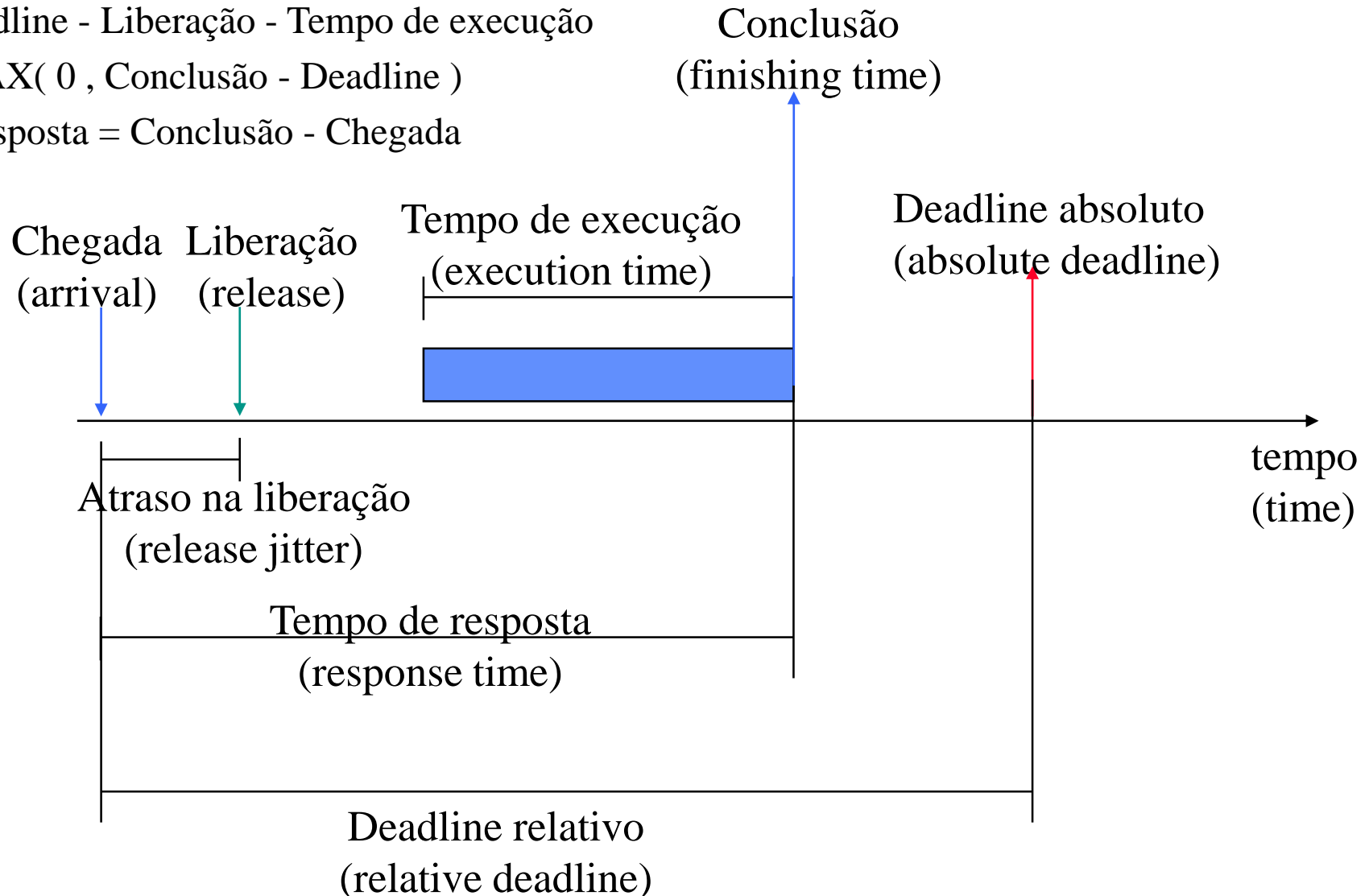
- Tempo de processador necessário para a tarefa
- Não inclui atrapalhões causadas pelas outras tarefas
- Depende de
  - Fluxo de controle
  - Dados de entrada
  - Comportamento da memória cache
  - Comportamento dos barramentos
  - Arquitetura do processador
- **WCET – Worst-case execution time**
  - Difícil de obter
  - Medições são incompletas
  - Modelos analíticos são complexos, pessimistas

# Propriedades Temporais das Tarefas 1/2

Folga = Deadline - Liberação - Tempo de execução

Atraso =  $\text{MAX}(0, \text{Conclusão} - \text{Deadline})$

Tempo de resposta = Conclusão - Chegada





## Propriedades Temporais das Tarefas 2/2

---

- Uma tarefa pode ser atrapalhada por tarefas de mais alta prioridade
- Este tempo é chamado de **interferência**
- $I_k$  é a máxima interferência recebida pela tarefa  $k$
- Se durante um intervalo de tempo executam sempre tarefas com prioridade igual ou superior à uma prioridade  $k$ , temos um período continuamente ocupado por tarefas (**busy period**)
- A duração deste tempo ocupado é denotada por  $W_k$
- **Tempo de resposta** vai da chegada da tarefa até a sua conclusão
- O máximo tempo de resposta da tarefa  $k$  é denotado por  $R_k$

## Relações de Exclusão Mútua

---

- Relações de **exclusão mútua** entre tarefas
  - Tarefas A e B apresentam exclusão mútua quando **NÃO** podem executar simultaneamente
- Exemplos:
  - Estrutura de dados compartilhada
  - Arquivo
  - Controlador de periférico
- $B_k$  representa o pior caso de tempo de **bloqueio** da tarefa k

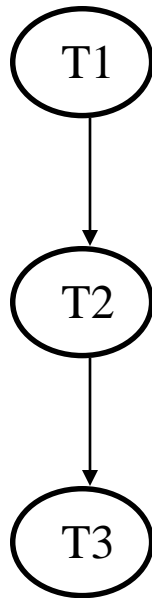
## Relações de Precedência 1/2

---

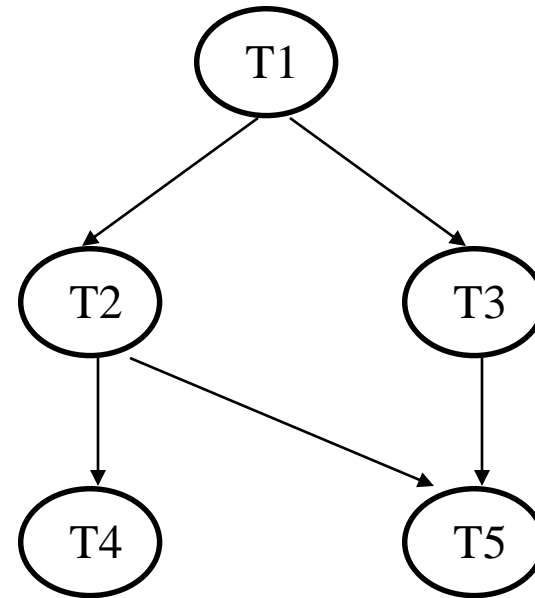
- Relações de **precedência** entre tarefas
  - Tarefa A é predecessora da tarefa B quando B somente pode iniciar depois que A estiver concluída
  - Neste caso, tarefa B é sucessora da tarefa A
  - Representado por  $A \rightarrow B$
  - Exemplo: envio de mensagem de A para B
- **Atividade (activity)**
  - Conjunto de tarefas interligadas por relações de precedência
  - Representada por um grafo onde
    - Nodos são as tarefas
    - Flexas são as relações de precedência

## Relações de Precedência 2/2

Atividade com Relações de Precedência LINEARES



Atividade com Relações de Precedência ARBITRÁRIAS



- **Modelo de Tarefas** ( *task model* )
  - Descrição das propriedades temporais das tarefas no sistema
  - Exemplo:
    - Tarefas são periódicas ou não
    - Tarefas com duração conhecida ou não
    - Tem ou não tem exclusão mútua
    - Etc
- Varia muito de sistema para sistema
- Ponto de partida para a análise de escalonabilidade

## Modelo de Tarefas 2/2

---

- O modelo de tarefas inclui vários parâmetros para caracterizar as propriedades temporais das tarefas
- C: tempo de execução no pior caso
- P: período ou intervalo mínimo entre ativações
- D: deadline relativo
- J: atraso de liberação (*release jitter*) máximo
- B: tempo de bloqueio no pior caso
- I: interferência
- W: tempo ocupado (*busy period*) no nível de prioridade
- R: tempo de resposta máximo

- **Carga de Tarefas** ( *task load* )
  - Parte de um dado modelos de tarefas
  - Descreve quais tarefas serão executadas
  - Quantifica as propriedades do modelo de tarefas
  
  - Carga Estática: Limitada e conhecida em projeto
  - Carga Dinâmica: Conhecida somente ao longo da execução

- Conceitos básicos
- Criticalidade
- Tipos de deadlines
- Previsibilidade
- Terminologia do escalonamento
- Event-Triggered x Time-Triggered
- Tipos de recorrência
- Tempo de execução
- Propriedades temporais das tarefas
- Relações de exclusão mútua e de precedência
- Modelo de tarefas e Carga de tarefas

