
Sistemas de Tempo Real: Introdução

Rômulo Silva de Oliveira
Departamento de Automação e Sistemas – DAS
UFSC
romulo@das.ufsc.br
<http://www.das.ufsc.br/~romulo>

Introdução

- Caracterização
- Mercados
- Concepções Erradas
- Conceitos Básicos
- Modelos de Tarefas
- Propriedades Temporais das Tarefas
- Área de Pesquisa

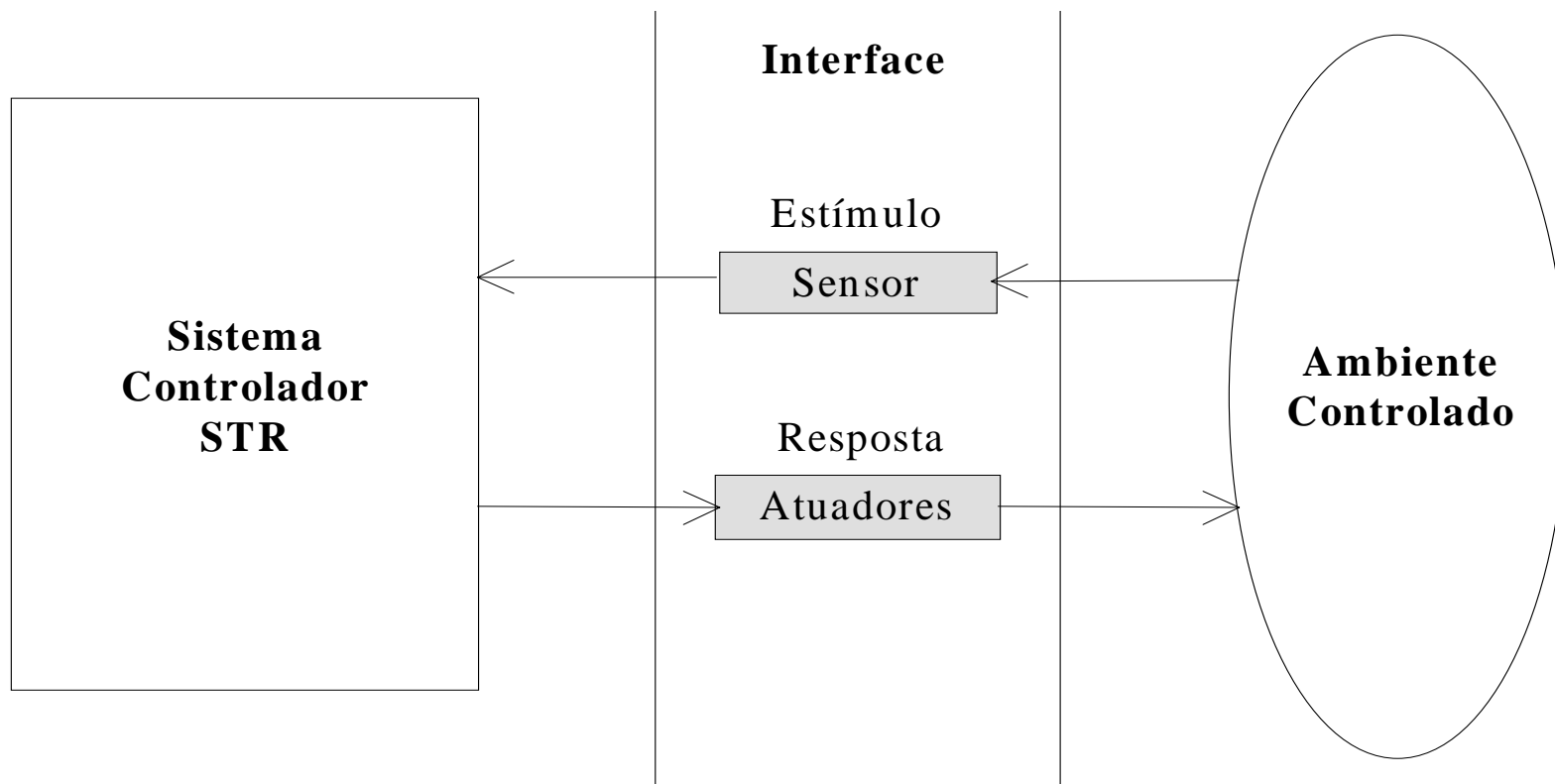
Caracterização

- Sistemas computacionais de tempo real:
 - Submetidos a requisitos de natureza temporal
 - Resultados devem estar corretos lógica e temporalmente
 - Requisitos definidos pelo ambiente físico
- Aspectos temporais
 - NÃO estão limitados a uma questão de maior ou menor desempenho
 - Estão diretamente associados com a funcionalidade
- Sistemas em geral:
 - “Fazer o trabalho usando o tempo necessário”
- Sistemas de tempo real:
 - **“Fazer o trabalho usando o tempo disponível”**

Caracterização

- Forte acoplamento de um STR com o seu Ambiente:
 - Processamento ativado por estímulos do ambiente
 - **Tempos de Resposta** delimitam **Estímulos/Respostas**
 - Processamentos devem terminar dentro de prazos (**deadlines**)
 - Se terminar fora de prazo sistema falha (**falha temporal**)
- Dados com prazos de validade:
 - Dados desatualizados ou não válidos (*outdated data*) podem conduzir a resultados ou respostas incorretas
- Fluxos de controle na execução são definidos pelo ambiente:
 - Impossibilidade em muitas aplicações do STR exercer um controle ou limitação nos estímulos provenientes do ambiente

Caracterização



- Telecomunicações
 - Centrais telefônicas, videoconferência, groupware
- Aeroespacial
 - Automação em aeronaves, sondas espaciais
- Defesa
 - Radar, sonar, sistema guia em mísseis
- Indústria
 - Controle de processos, robôs, aquisição de dados
- Financeiro
 - Transações em bolsa, negociação automática
- Entretenimento
 - Vídeo games, vídeo sob demanda, áudio

Concepções Erradas

- Tempo real significa execução rápida
- Computadores mais rápidos vão resolver todos os problemas
- Sistemas de tempo real são:
 - pequenos
 - escritos em linguagem de montagem (assembly)
 - formados apenas por tratadores de interrupção
 - device-drivers

Conceitos Básicos

- **Tarefa (task)**
 - Segmento de código cuja execução possui atributo temporal próprio (período, deadline, etc)
 - Exemplo: método em OO, sub-rotina, trecho de um programa
- **Deadline**
 - Instante máximo desejado para a conclusão de uma tarefa
- **Deadline relativo**
 - Em relação ao início da tarefa
- **Deadline absoluto**
 - Em relação a UTC (relógio da parede)

Conceitos Básicos

- Tempo real crítico (**hard real-time**)
 - Falha temporal pode resultar em conseqüências 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

Previsibilidade

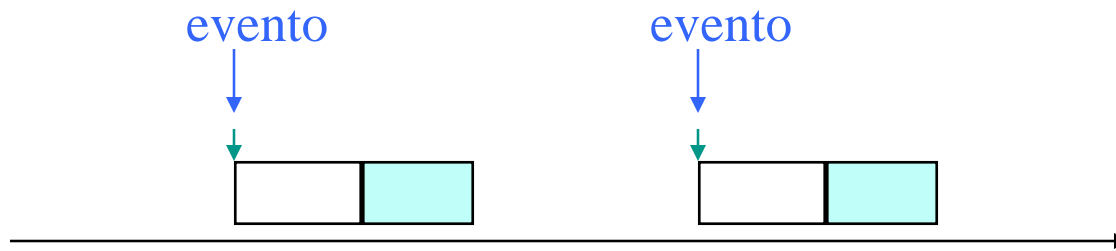
- **Previsibilidade** (“predictability”)
 - Está associada a capacidade de poder antecipar, em tempo de projeto, se os processamentos em um STR serão executados dentro de seus prazos especificados
- Associada a uma **previsão determinista**
 - todos os deadlines serão respeitados
- ou a uma **antecipação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

Conceitos Básicos

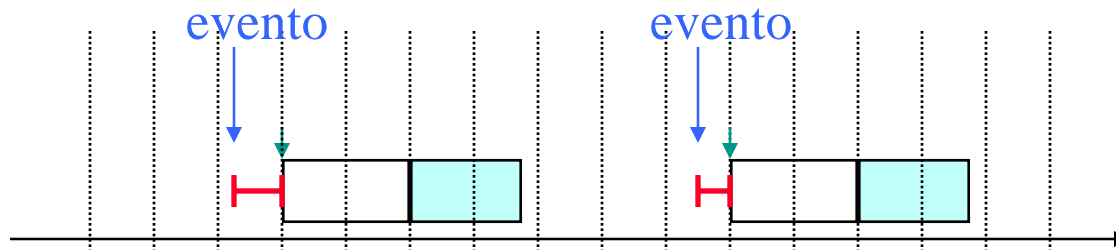
- **Escalonamento** (scheduling)
 - Identifica a forma como recursos são alocados às tarefas
- **Escalonador** (scheduler)
 - Componente do sistema responsável pela gerência dos recursos
- **Escala de Execução** (schedule)
 - Descreve quando cada tarefa ocupa cada recurso
- **Escalonamento Estático**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

Event-Triggered x Time-Triggered

- Event-Triggered



- Time-Triggered



Event-Triggered x Time-Triggered

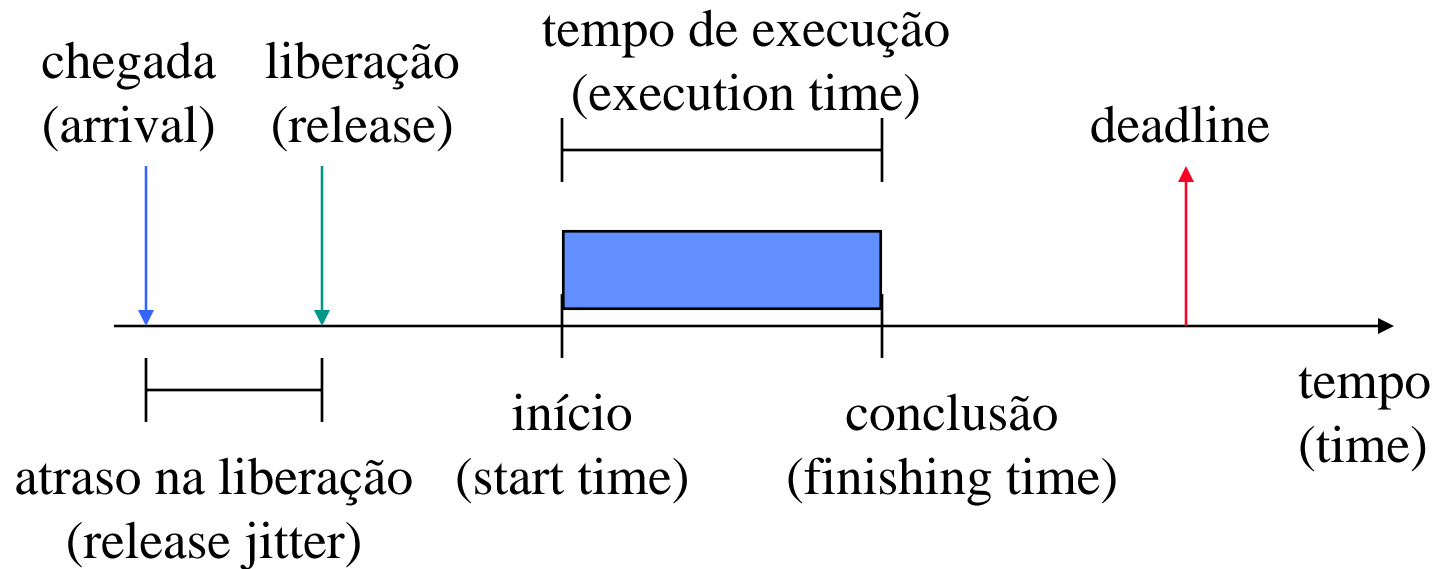
- **Dirigido por eventos (Event-Triggered)**
 - Sistema reage a eventos
 - Evento externo gera interrupção e dispara tarefa
 - Funciona bem com carga pequena
 - Pode falhar no caso de carga pesada
 - Ex: Cano estourado gera chuva de eventos

- **Dirigido pelo tempo (Time-Triggered)**
 - Interrupção de relógio a cada T milisegundos (tick)
 - A cada tick alguns sensores e atuadores são acessados
 - Não existem interrupções além das do relógio
 - Problema é selecionar T (carga x atraso)
 - Ex: Relação causa-efeito em trem e cancela

Modelos de Tarefas

- **Modelo de Tarefas** (*task system*)
 - Descrição das propriedades temporais das tarefas no sistema
 - Exemplo: periódicas ou não, com duração conhecida ou não, etc
- Varia muito de sistema para sistema
- Ponto de partida para:
 - Análise de escalonabilidade
 - Escolha do suporte
 - Linguagem de programação
 - Sistema operacional
 - Arquitetura do computador
- **Carga de Tarefas** (*task load*)
 - Descrição de quais tarefas deverão ser executadas
 - Estática: Limitada e conhecida em projeto
 - Dinâmica: Conhecida somente ao longo da execução

Propriedades Temporais das Tarefas



Folga = Deadline - Liberação - Tempo de execução

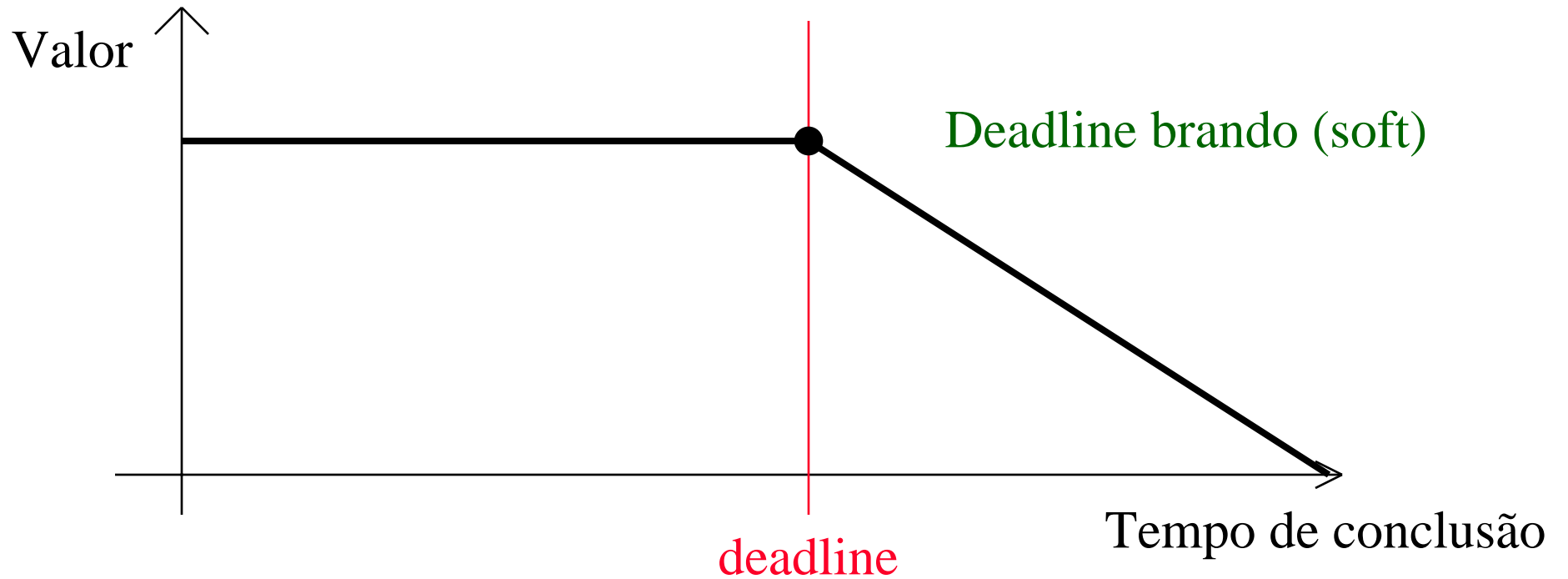
Atraso = MAX(0 , Conclusão - Deadline)

Tempo de resposta = Conclusão - Chegada

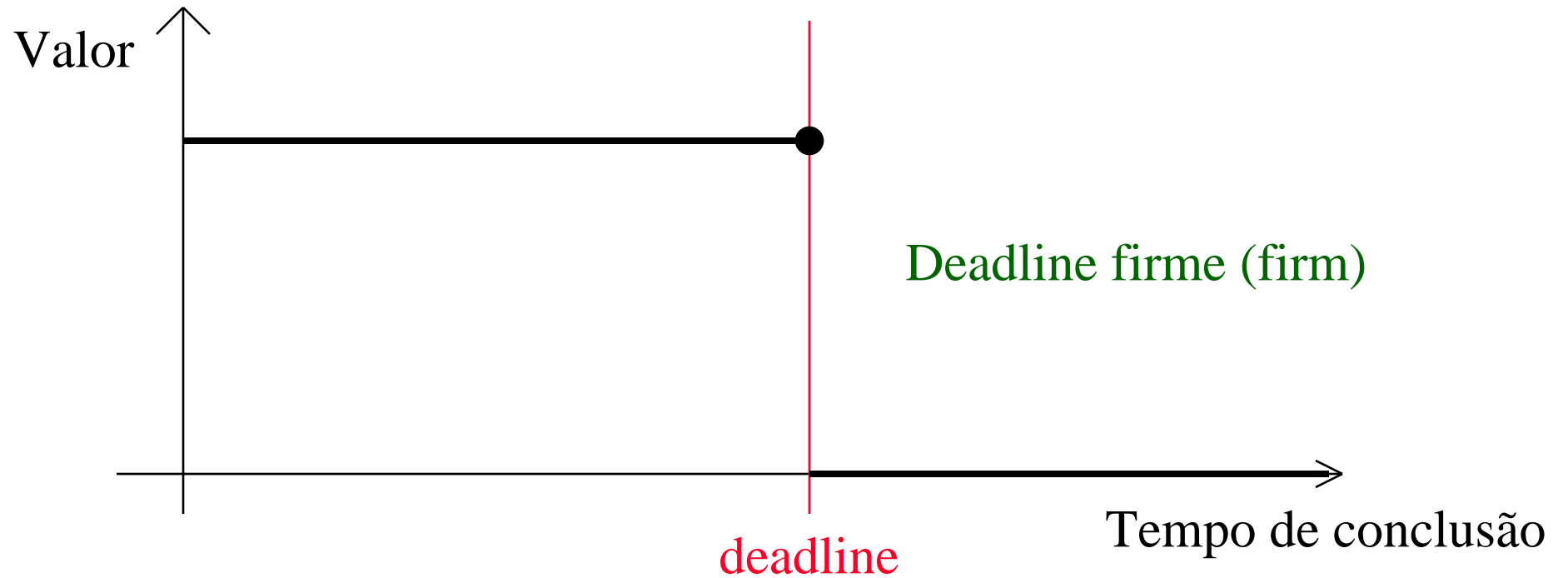
Tipos de Deadlines

- **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

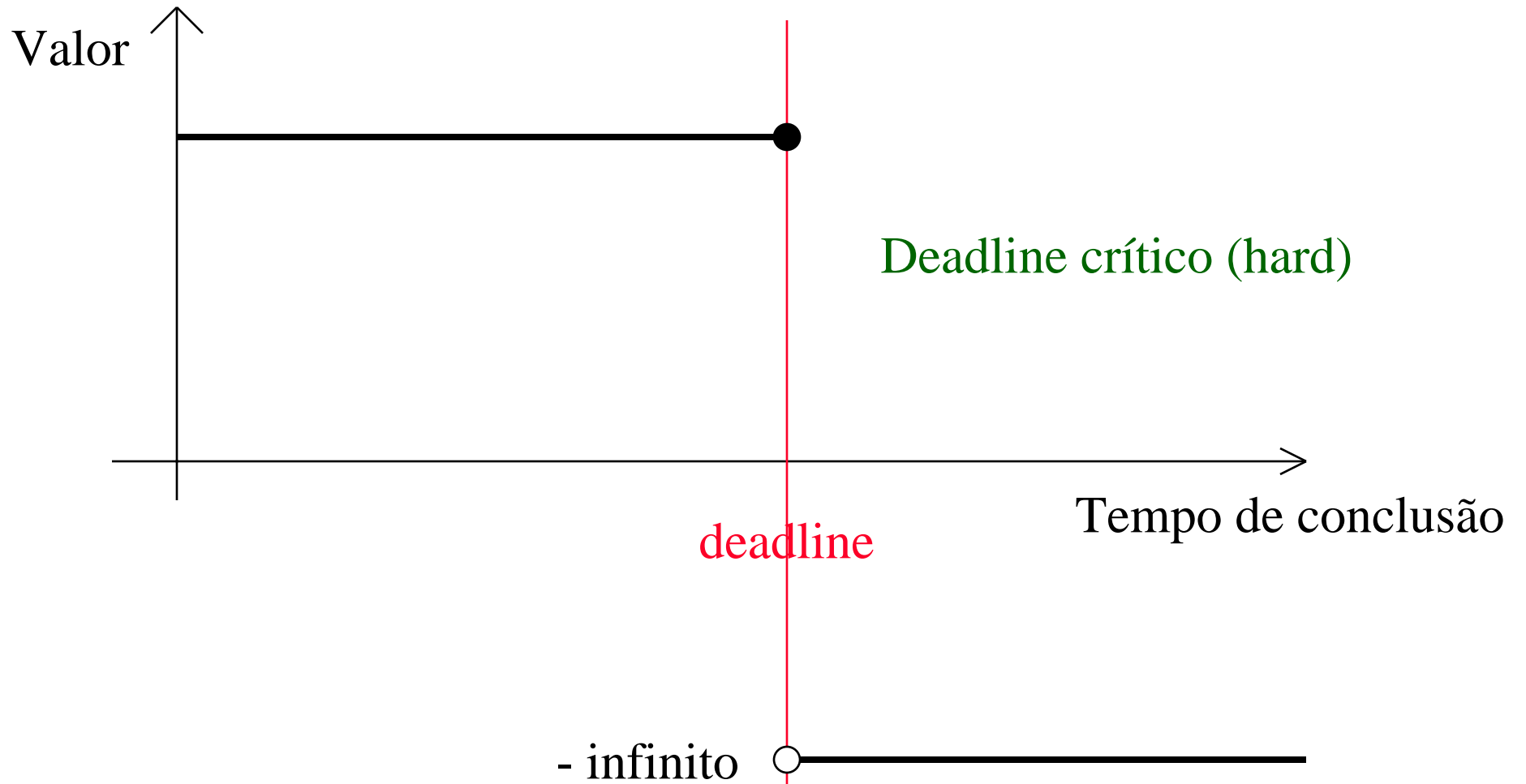
Tipos de Deadlines



Tipos de Deadlines



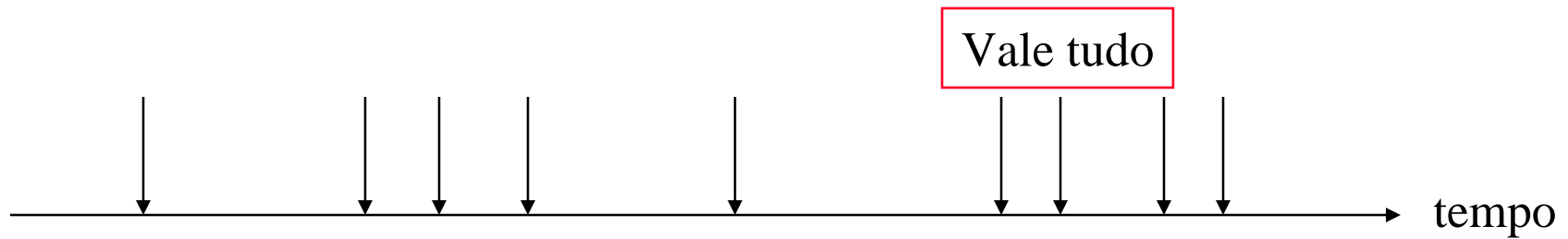
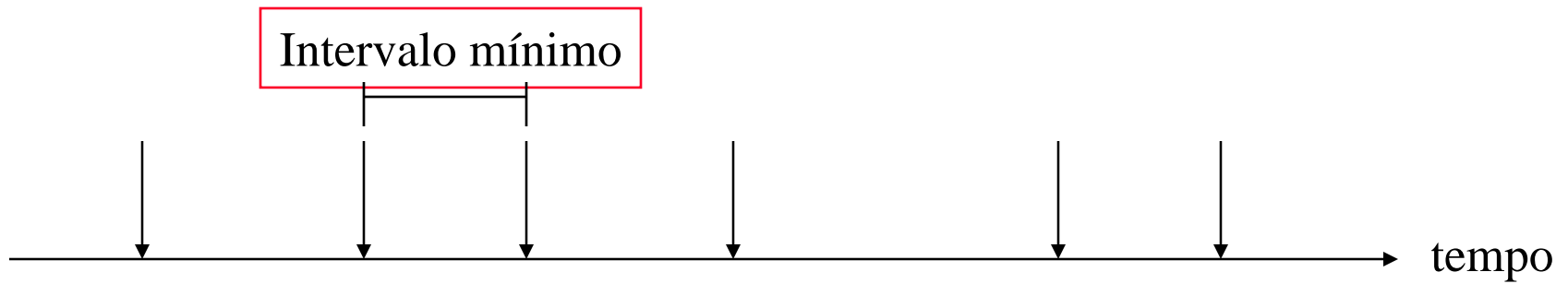
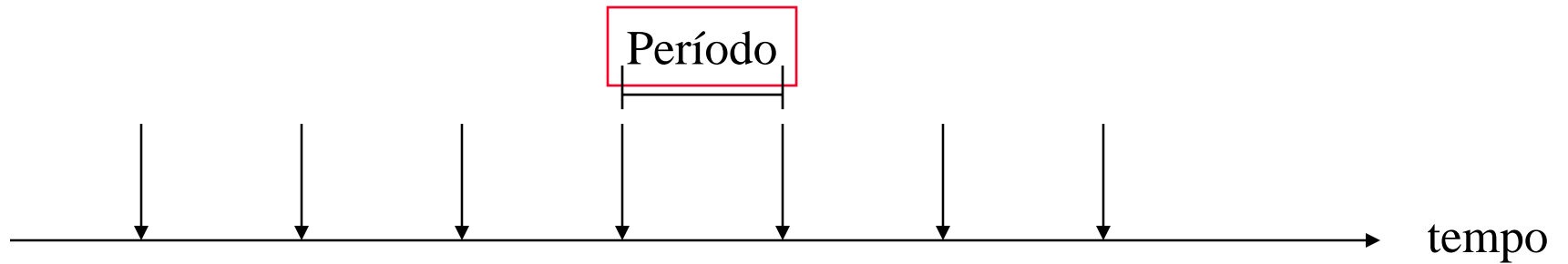
Tipos de Deadlines



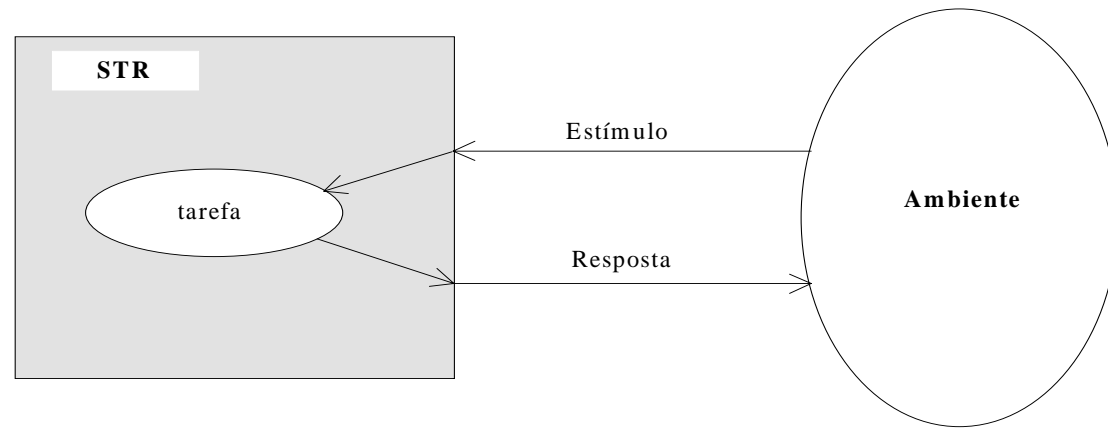
Tipos de Recorrência

- **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
- **Tarefas 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

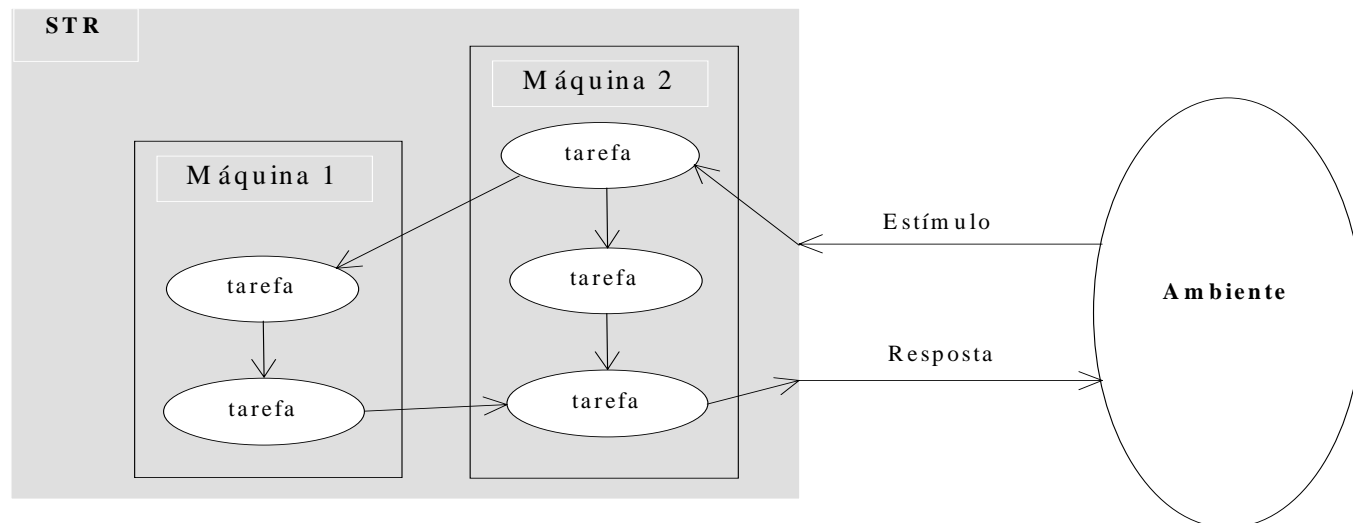
Tipos de Recorrência



Complexidade da Aplicação



Sistema Simples: Tarefa Única Responde ao Ambiente



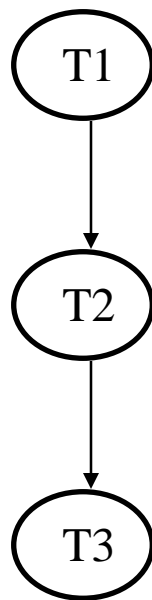
Sistema Complexo: Grafo de Tarefas Responde ao Ambiente

Relações de Precedência

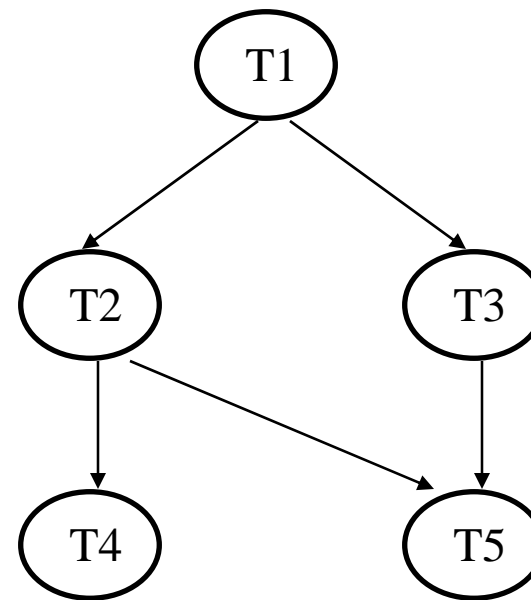
- Relações de **precedência** entre tarefas
 - Tarefa A é predecessora da tarefa B quando B somente pode iniciar depois que A estiver concluída
 - Neste caso, tarefa B é sucessora da tarefa A
 - Representado por $A \rightarrow B$
 - Exemplo: envio de mensagem de A para B
- **Atividade**
 - Conjunto de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

Relações de Precedência

Atividade com Relações de Precedência LINEARES



Atividade com Relações de Precedência ARBITRÁRIAS



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

- Sistemas de Tempo Real
 - Possuem requisitos temporais
 - Escalonamento recebe o maior destaque
 - Existem vários aspectos envolvidos
- Sistemas de Tempo Real possuem diferentes aspectos
 - Escalonamento
 - Modelos e linguagens de programação
 - Protocolos de comunicação
 - Arquitetura de computadores
 - Metodologias de desenvolvimento
 - etc

Evolução Rápida

- Área em rápida evolução
- Teoria de escalonamento desenvolvida nos anos 90
- Prática começa a mudar em função da teoria
- Existe muito a ser feito

- Evolução acontece em ondas
 - Pesquisa teórica (teoremas, modelos abstratos)
 - Desenvolvimento tecnológico (algoritmos, modelos realistas)
 - Empacotamento da tecnologia (compilador, sistema operacional)
 - Construção de aplicações