

Sistemas de Tempo Real: Caracterização

Rômulo Silva de Oliveira
Departamento de Automação e Sistemas – DAS – UFSC

romulo@das.ufsc.br
<http://www.das.ufsc.br/~romulo>
Maio/2010

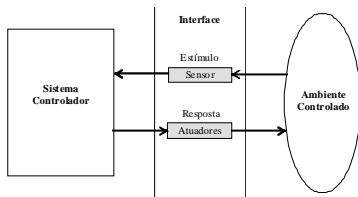
Rômulo Silva de Oliveira, DAS-UFSC, maio/2010 1

Caracterização

- Sistemas computacionais com requisitos de tempo real:
 - Submetidos a requisitos de natureza temporal não triviais
 - Resultados devem estar corretos lógica e temporalmente
 - Requisitos definidos pelo ambiente físico
 - O problema é de tempo real
- 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”

Rômulo Silva de Oliveira, DAS-UFSC, maio/2010 2

Caracterização



Rômulo Silva de Oliveira, DAS-UFSC, maio/2010 3

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 - Dados desatualizados ou não válidos (*outdated data*) podem conduzir a resultados ou respostas incorretas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 estímulos na execução são definidos pelo ambiente:
 - Quase sempre impossível controlar os estímulos provenientes do ambiente

Rômulo Silva de Oliveira, DAS-UFSC, maio/2010 4

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 necessidade de previsibilidade gera implicações em todos os níveis:
 - linguagens
 - sistemas operacionais
 - comunicação
 - arquitetura do computador
 - etc

Rômulo Silva de Oliveira, DAS-UFSC, maio/2010 5

Mercados

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

Rômulo Silva de Oliveira, DAS-UFSC, maio/2010 6

Concepções Erradas

- Tempo real significa execução rápida
 - Controlador semafórico
 - Amostragem de dados
- Computadores mais rápidos vão resolver todos os problemas
 - Custo
 - Multicore
 - Design do software
- Sistemas de tempo real são:
 - Pequenos
 - Escritos em linguagem de montagem (assembly)
 - Formados apenas por tratadores de interrupção
 - Compostos por device-drivers

Rômulo Silva de Oliveira, DAS-UFSC, maio/2010 7

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)

Rômulo Silva de Oliveira, DAS-UFSC, maio/2010 8

Criticalidade

- 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

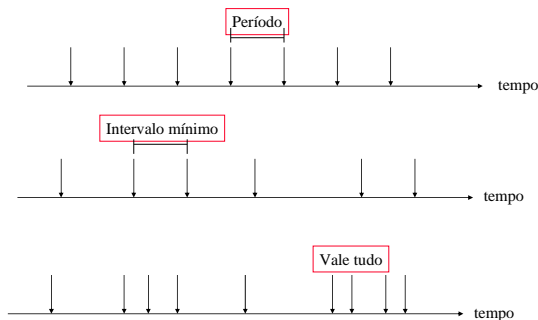
Rômulo Silva de Oliveira, DAS-UFSC, maio/2010 9

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

Rômulo Silva de Oliveira, DAS-UFSC, maio/2010 10

Tipos de Recorrência



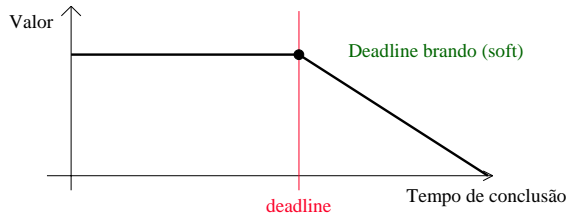
Rômulo Silva de Oliveira, DAS-UFSC, maio/2010 11

Tipos de Deadlines

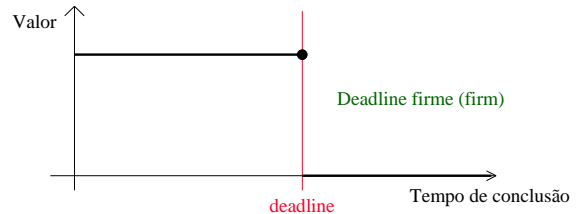
- **Deadline Hard**
 - Perda do deadline pode ter consequências catastróficas
 - Exemplo: abrir válvula em duto de alta pressão
- **Deadline Firm**
 - Perda do deadline NÃO tem consequências catastróficas
 - Não existe valor em terminar a tarefa após o deadline
 - Exemplo: amostrar periodicamente valor 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

Rômulo Silva de Oliveira, DAS-UFSC, maio/2010 12

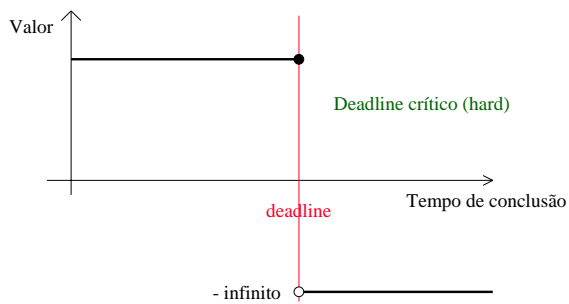
Tipos de Deadlines



Tipos de Deadlines



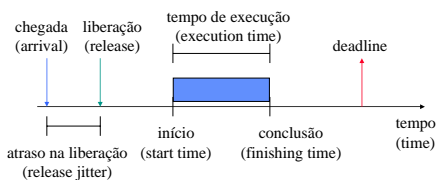
Tipos de Deadlines



Tempo de Execução

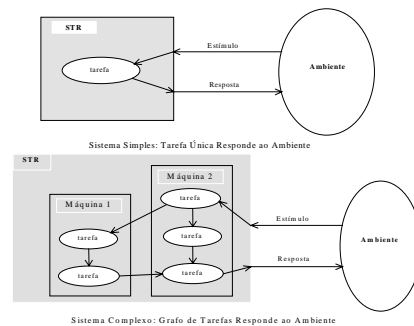
- Tempo de processador necessário para a tarefa
- Não inclui atrapalhamentos causadas pelas outras tarefas
- Depende de
 - Fluxo de controle
 - Dados de entrada
 - Comportamento da memória cache
 - Comportamento dos barramentos
 - Arquitetura do processador
- WCET – Worst-case execution time
 - Difícil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

Propriedades Temporais das Tarefas



Folga = Deadline - Liberação - Tempo de execução
 Atraso = MAX(0 , Conclusão - Deadline)
 Tempo de resposta = Conclusão - Chegada

Complexidade da Aplicação



Relações de Exclusão Mútua

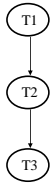
- Relações de **exclusão mútua** entre tarefas
 - Tarefas A e B apresentam exclusão mútua quando NÃO podem executar simultaneamente
- Exemplos:
 - Estrutura de dados compartilhada
 - Arquivo
 - Controlador de periférico

Relações de Precedência

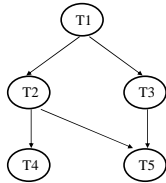
- Relações de **precedência** entre tarefas
 - Tarefa A é predecessora da tarefa B quando B somente pode iniciar depois que A estiver concluída
 - Neste caso, tarefa B é sucessora da tarefa A
 - Representado por $A \rightarrow B$
 - Exemplo: envio de mensagem de A para B
- **Atividade**
 - Conjunto de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

Relações de Precedência

Atividade com Relações de Precedência LINEARES



Atividade com Relações de Precedência ARBITRÁRIAS



Modelo de Tarefas

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

Carga de Tarefas

- **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

Tipos de Escalonadores

- **Escalonamento (scheduling)**
 - Identifica a forma como recursos são alocados às tarefas
- **Escalonador (scheduler)**
 - Componente do sistema responsável pela gestão 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

Resumo

- **Sistemas de Tempo Real**
 - Possuem requisitos temporais
 - Escalonamento recebe o maior destaque
 - Existem diversas questões a serem consideradas para o escalonamento

- **Sistemas de Tempo Real possuem diferentes aspectos**
 - Escalonamento
 - Modelos e linguagens de programação
 - Protocolos de comunicação
 - Arquitetura de computadores
 - Metodologias de desenvolvimento
 - etc