

Introdução aos Sistemas de Tempo Real

Sergio Cavalcante

Centro de Informática - UFPE

str-1@cin.ufpe.br

svc@cin.ufpe.br

Assunto: [str]

88350950

34254714

Sistemas de Tempo Real

- Sistemas computacionais de tempo real:
 - Tem aspectos temporais em sua especificação
 - Submetidos a requisitos de natureza temporal
 - Resultados devem estar corretos lógica e temporalmente
“Fazer o trabalho usando o tempo disponível”
 - Requisitos definidos pelo ambiente físico
- Aspectos temporais
 - Não estão limitados a uma questão de desempenho
 - Estão diretamente associados com a funcionalidade

Sistema de Tempo Real

- Sistema no qual o comportamento está sujeito a restrições temporais

Tipos de uso

- Controle
 - Baixo processamento
 - Requer pouca memória
 - Altamente reativo (muito interativo)
- Processamentos de dados
 - Processamento digital de sinais
 - Alto processamento
 - Pouco interativo

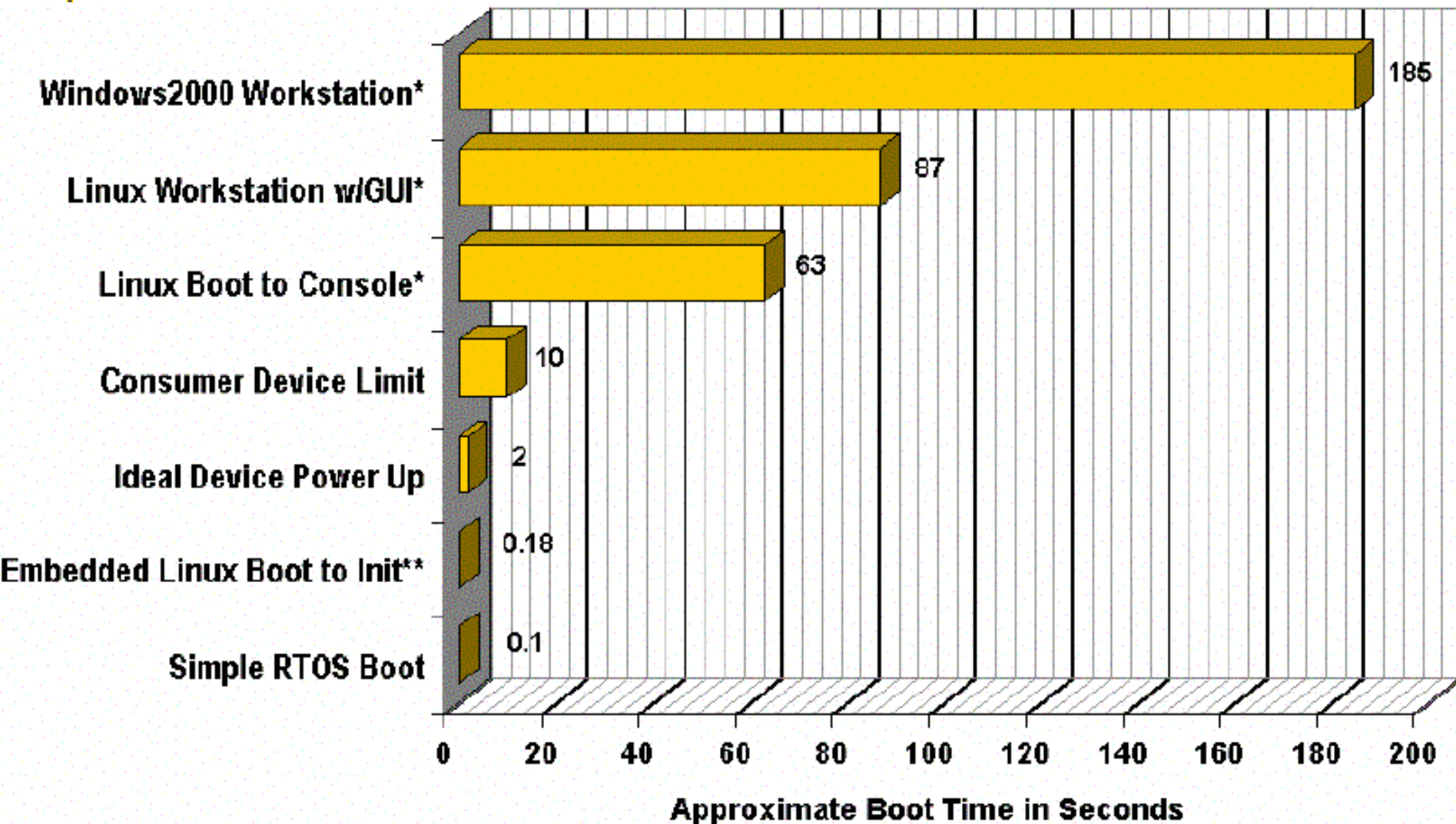
Sistema de Controle de Tempo Real

- Monitoram e modificam seu ambiente
- Estão associados a dispositivos de hardware
 - Sensores: coletam dados do ambiente
 - Atuadores: Mudam o ambiente do sistema
- Tempo é crítico

Áreas de Pesquisa

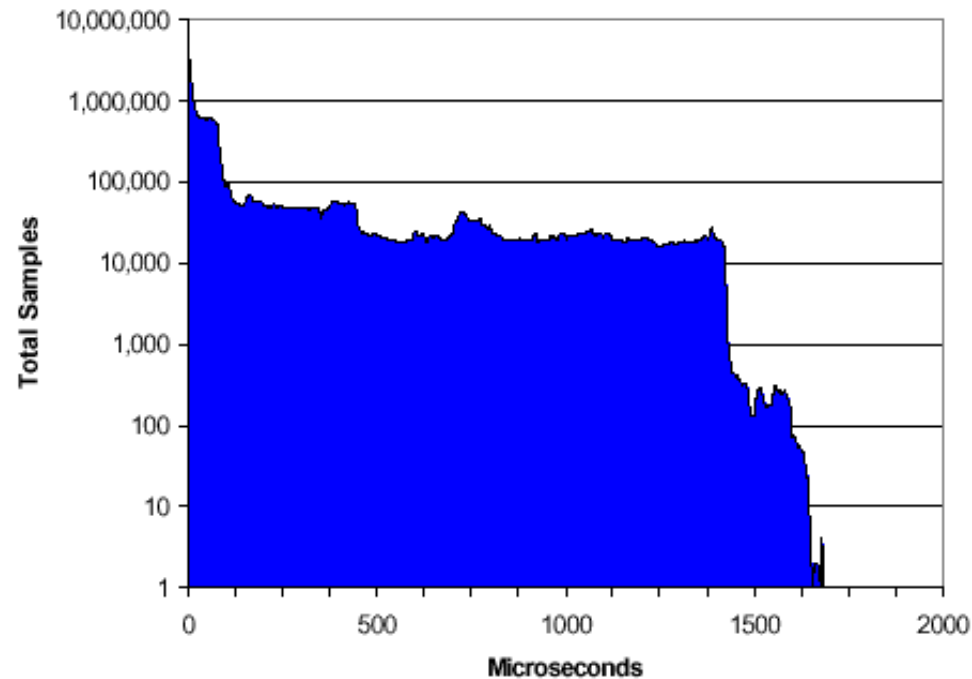
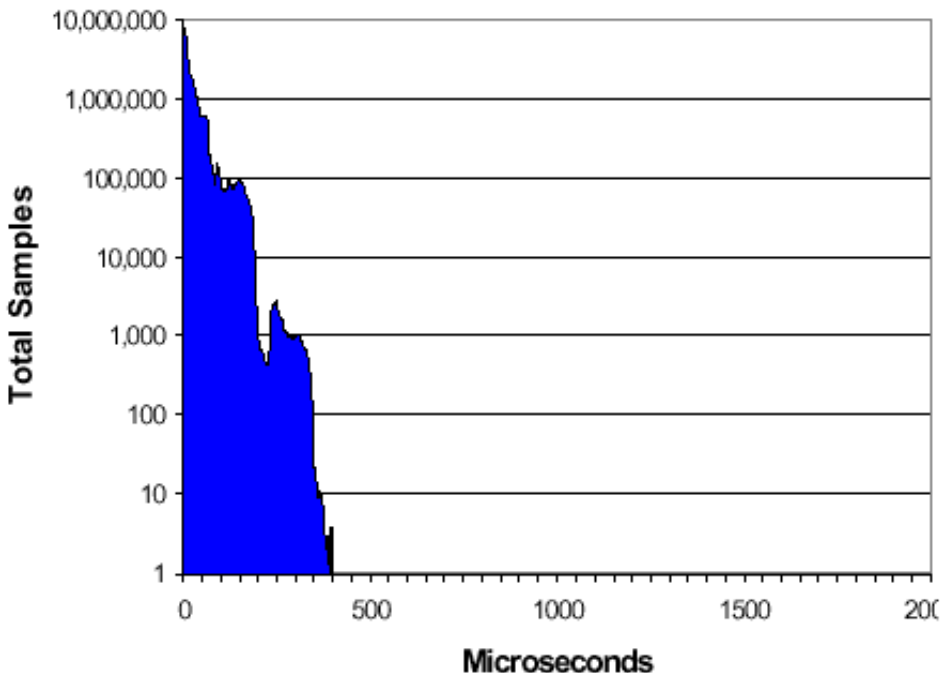
- Escalonamento (mais estudado)
- Modelos e linguagens de programação
- Protocolos de comunicação
- Arquitetura de computadores
- Metodologias de desenvolvimento
- ...

Exemplo: Tempo de Boot



Exemplo: Mudança de Contexto

- MontaVista Linux Kernel 2.4.17
- Pior caso 436us (vs. 1743us)
- 99.9% < 195us (vs. 1420us)

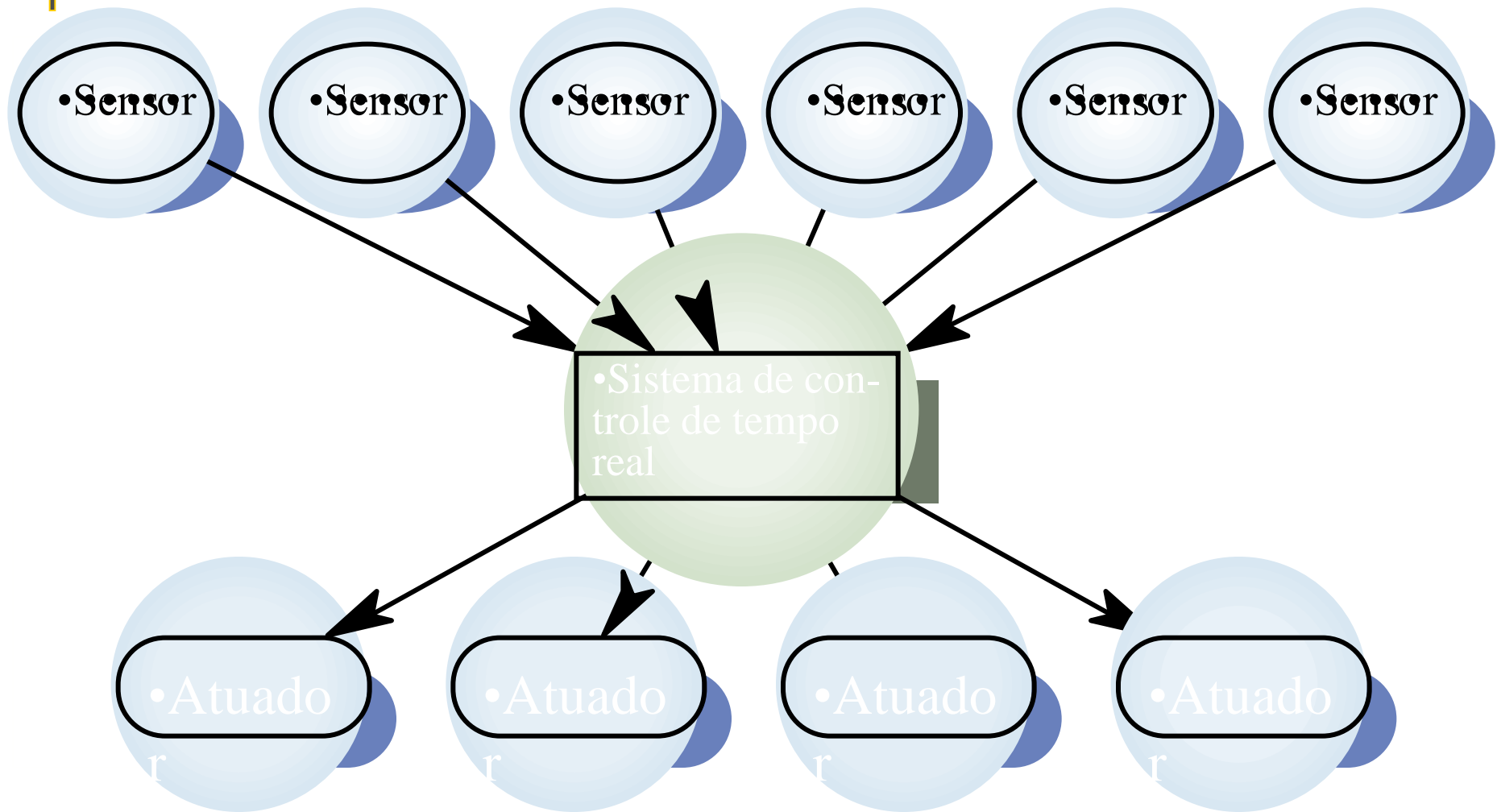


Fonte: Bill Weinberg, "Linux and XScale for Wireless Consumer Electronics Devices", IDF'Fall03

STR e o Ambiente



Modelo de sistema de tempo real



STR e o Ambiente

- Fluxo de controle definido pelo ambiente
 - Tarefas ativadas por estímulos do ambiente
 - STR geralmente não tem como limitar estes estímulos
- Tempos de resposta ao ambiente definem deadlines
- Dados com prazos de validade
 - Dados desatualizados podem conduzir a respostas incorretas

Conceitos Básicos

Tarefa (task)

- Segmento de código cuja execução possui atributo temporal próprio
- Exemplo: método em OO, subrotina, trecho de um programa

Deadline

- Instante máximo desejado para a conclusão de uma tarefa

Criticidade

- Soft real-time system: sua operação é **degradada** se as restrições de tempo não são obedecidas
- Hard real-time system: sua operação é **incorreta** se as restrições de tempo não são obedecidas

Criticidade

Sistema de tempo real crítico (Hard real-time system)

- Todas as tarefas têm Hard Deadline
 - Perda do deadline pode ter conseqüências catastróficas
- É necessário garantir requisitos temporais ainda durante o projeto
- Exemplo: usina nuclear, indústria petroquímica, mísseis

Sistema de tempo real não crítico (Soft real-time system)

- O requisito temporal descreve apenas o comportamento desejado
- Perda do deadline não tem conseqüências catastróficas
- Existe interesse em terminar a tarefa mesmo com atraso
- Exemplo: início de gravação de vídeo-cassete

Criticidade

Deadline Firm

- Perda do deadline não tem consequências catastróficas
- Não existe valor em terminar a tarefa após o deadline
- Exemplo: ler o valor da temperatura

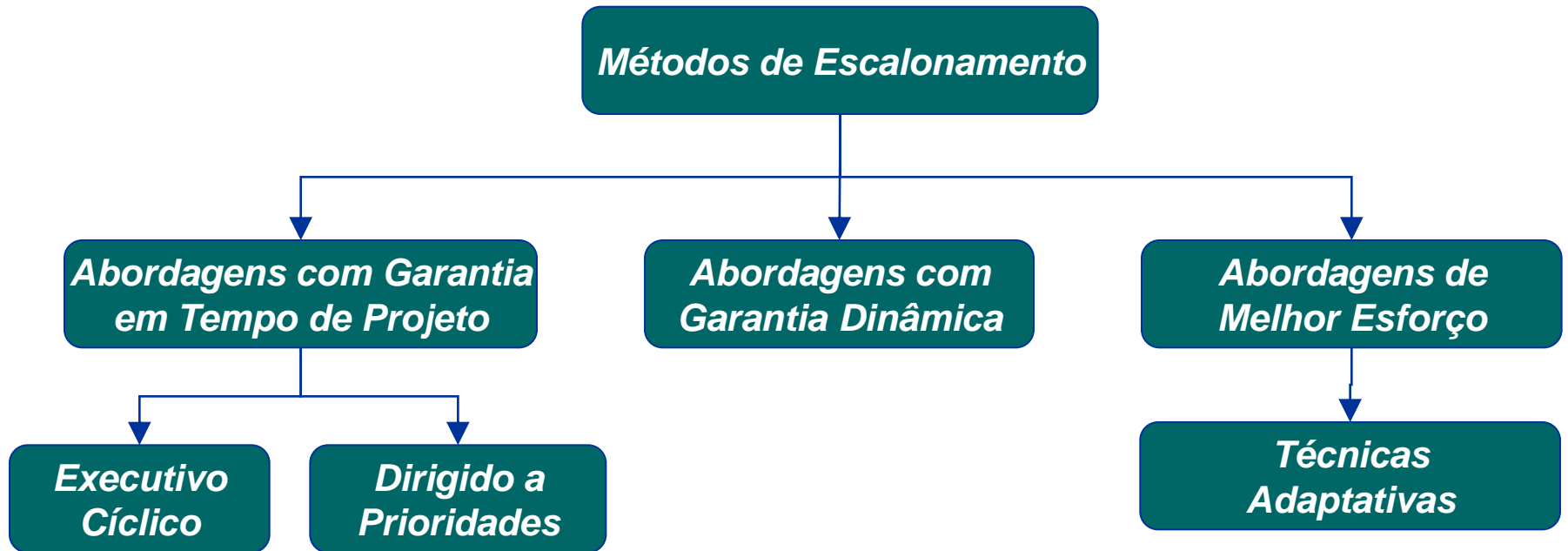
Modelagem das Tarefas

- Deadline – Tempo máximo para término de uma tarefa
- Release-time – Tempo mínimo para início de uma tarefa
- Tempo de execução: considera o pior caso
 - WCET – Worst-Case Execution Time

Sistemas reativos

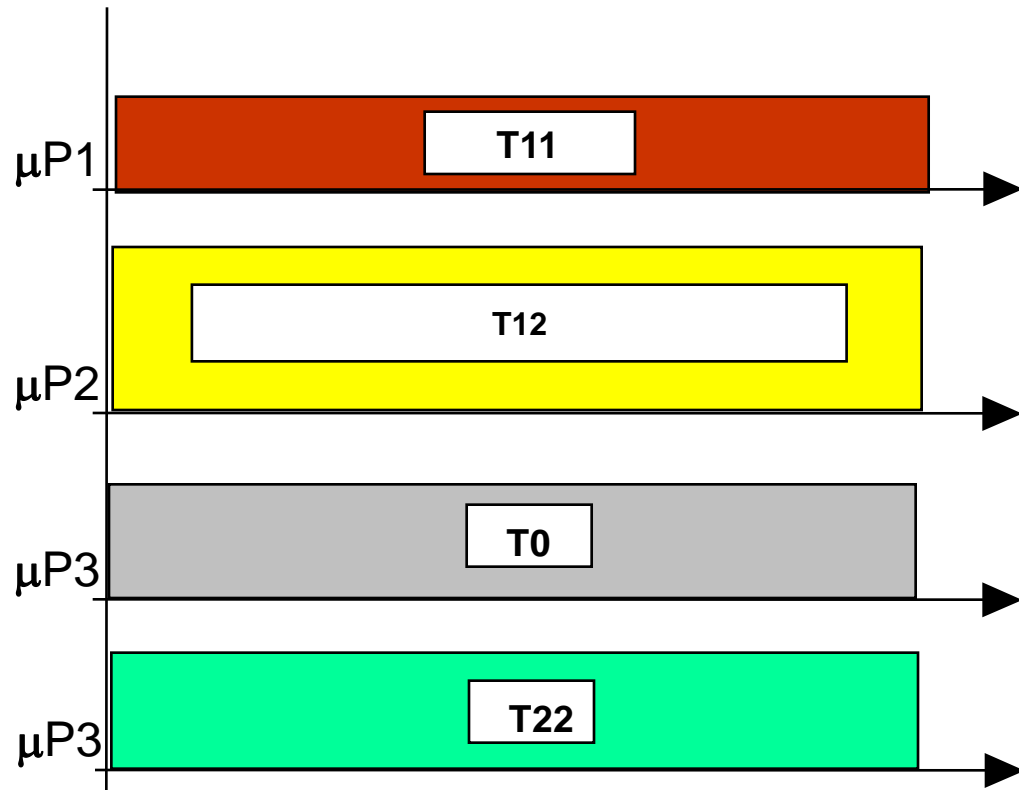
- Dado um estímulo o sistema deve produzir uma resposta dentro do tempo estipulado.
- Estímulo periódico: ocorre em intervalos de tempo previsíveis
- Estímulo aperiódico: ocorre em intervalos imprevisíveis
- Estímulo esporádico: não é periódico mas sabemos o intervalo mínimo entre ocorrências

Escalonamento



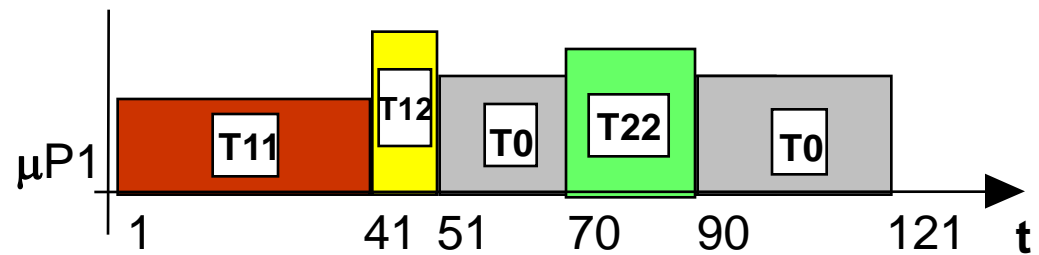
Abstração

- Uma máquina para cada processo
- Paralelismo real



Realidade

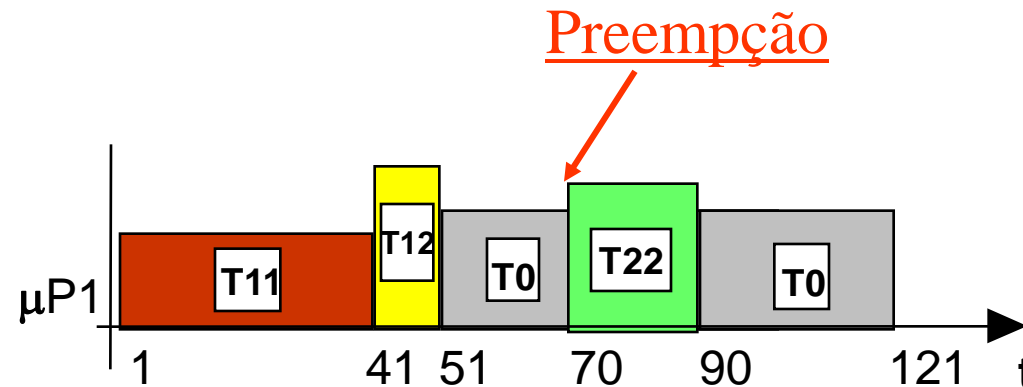
- Compartilhamento do tempo
- Pseudoparalelismo



Tipos de Escalonamento

Preemptivo

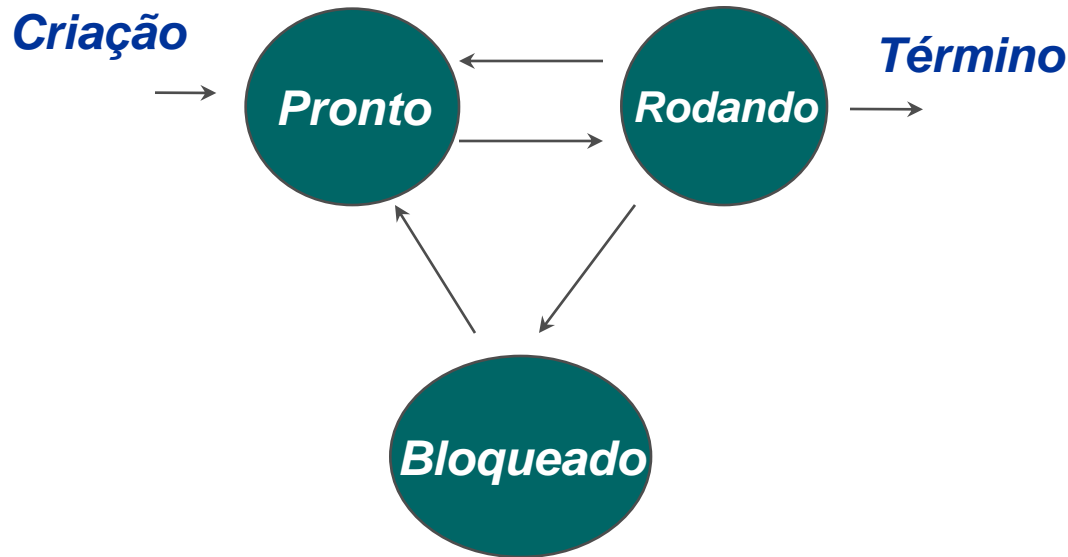
- Permite a suspensão temporária de processos
- Quantum ou time-slice: período de tempo durante o qual um processo usa o processador a cada vez



- Quantum grande:
 - Diminui núm. de mudanças de contexto e overhead do S.O.
 - Ruim para processos interativos

Processo

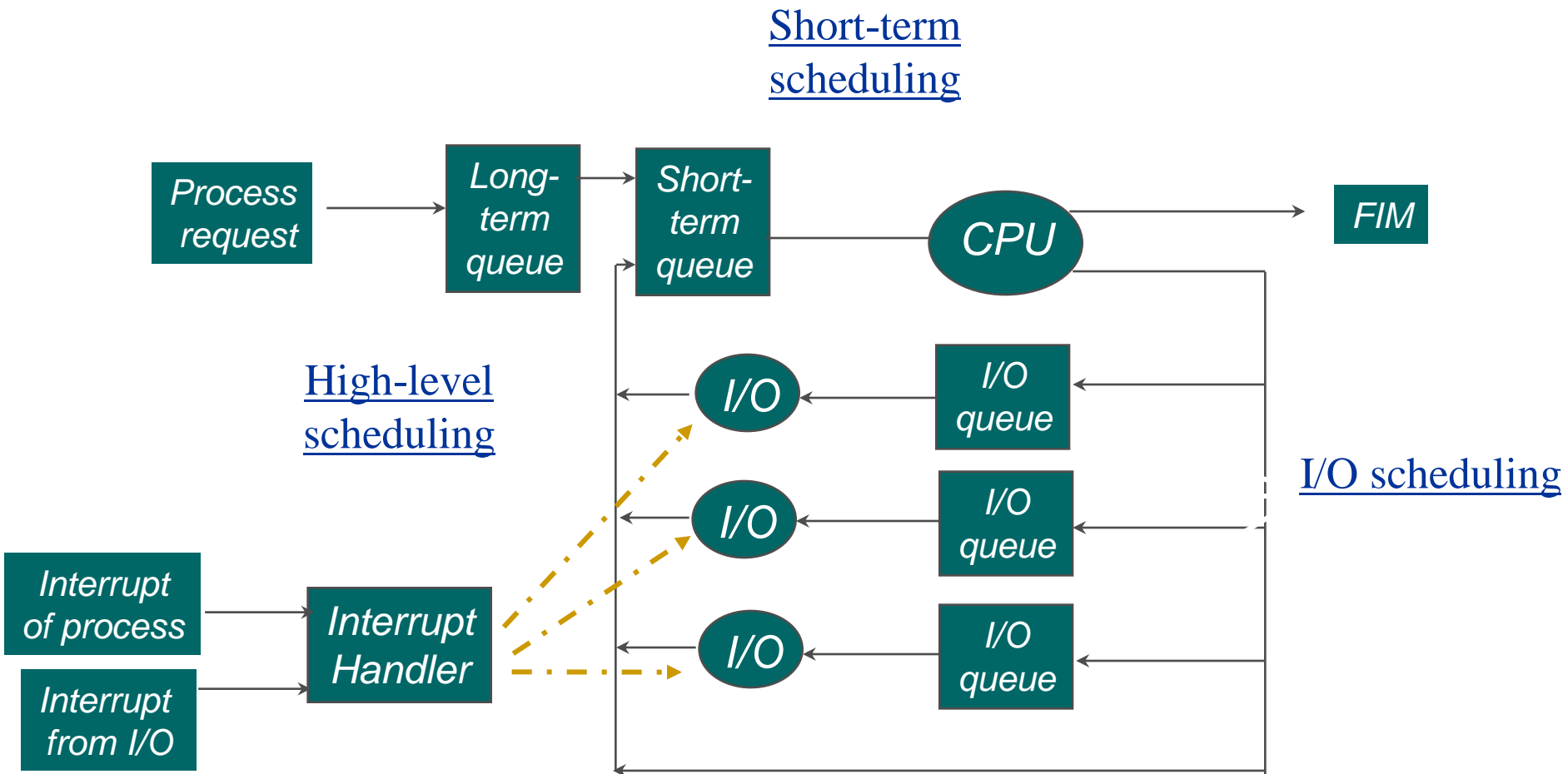
- Estados de um processo



Process Control Block - PCB

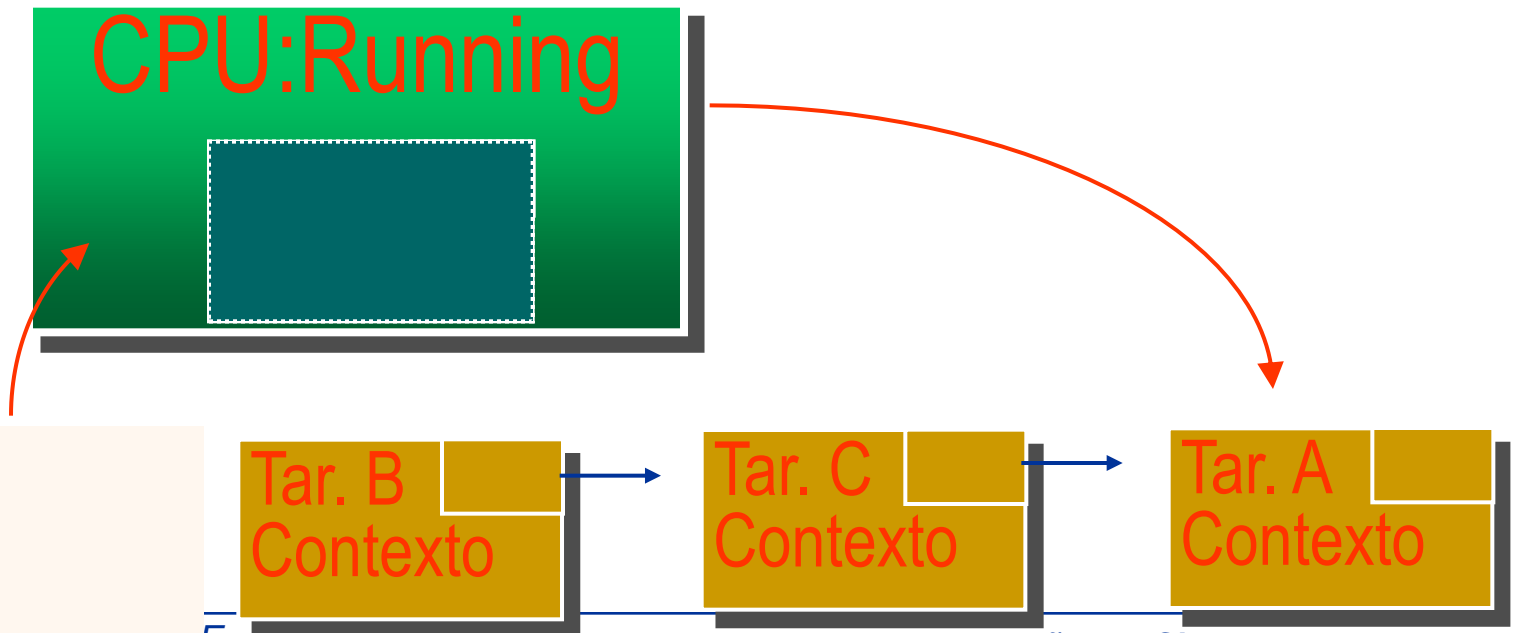
ID do Processo
Estado
Prioridade
Program Counter
Ponteiros da Memória
Contexto (regs.)
I/O Status
Informações gerais <ul style="list-style-type: none">• tempo de CPU• limites, usuário, etc.

Filas de Escalonamento



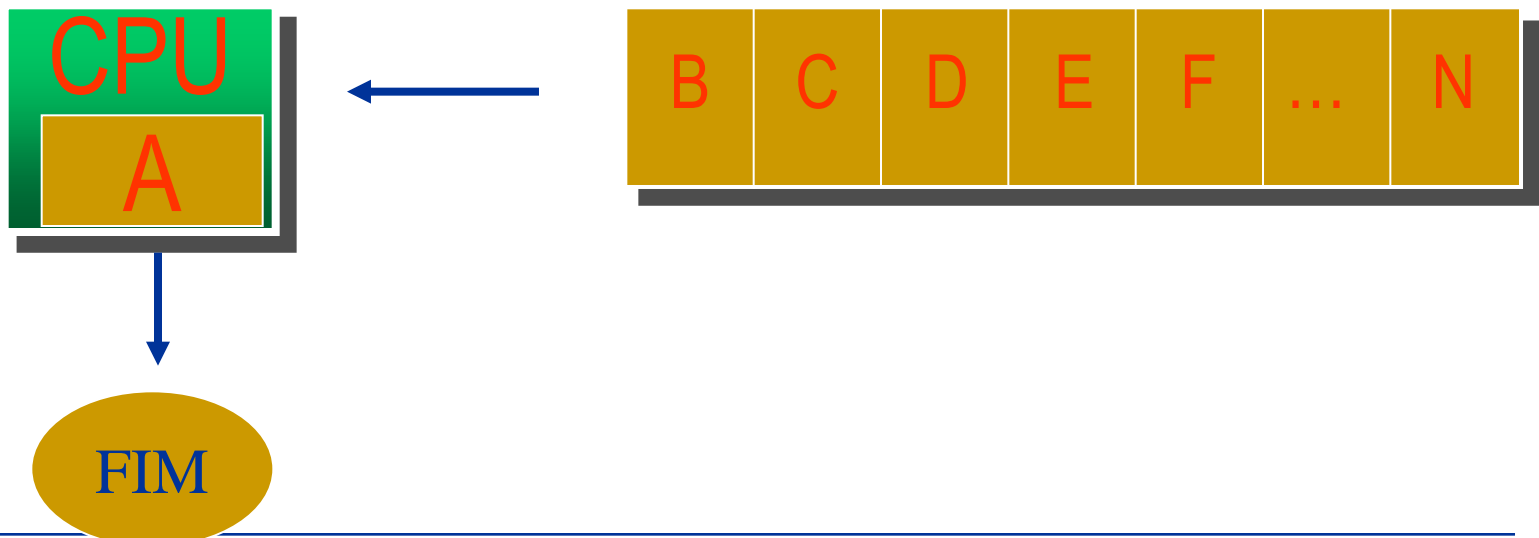
Round-Robin

- Uso de uma lista de processos sem prioridade
- Escalonamento preemptivo
- Simples e justo
- Bom para sistemas interativos



First-In First-Out (FIFO)

- Uso de uma lista de processos sem prioridade
- Escalonamento não-preemptivo
- Simples e justo
- Bom para sistemas em batch



Tipos de Escalonamento

Exemplo

- Três processos
- Cada um consome 1 hora de CPU
- Calcule o tempo de resposta considerando
 - a) Round-robin (time-slice = 10ms)
 - b) FIFO

Características de Escalonamento

- Justiça (fairness)
 - Todos os processos têm chances iguais de uso dos processador
- Eficiência
 - O objetivo é manter o processador 100% ocupado
- Tempo de Resposta
 - Tempo entre a requisição da ação e a obtenção do resultado
- Throughput
 - Número de tarefas por unidade de tempo

Garantia Dinâmica e de Melhor Esforço

- Não existe garantia que os deadlines serão cumpridos
- Sempre que uma tarefa é ativada ocorre uma análise da sua escalonabilidade
- Passível de sofrer sobrecarga
- Capaz de fornecer análise probabilística
 - Simulação, teoria das filas de tempo real, etc
- Algumas abordagens oferecem Garantia Dinâmica
 - Garante o deadline (ou não) no início da ativação

Garantia Dinâmica e de Melhor Esforço

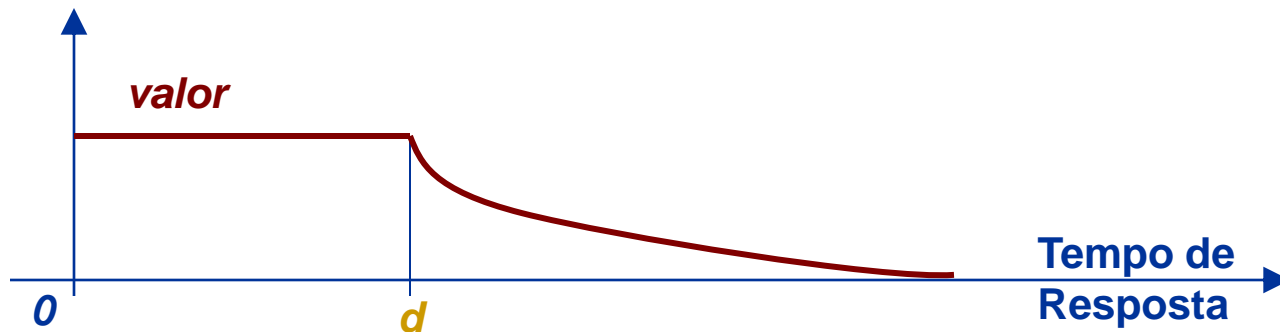
Descarte de Tarefas na Sobrecarga

- As tarefas executadas cumprem o deadline
- Mais apropriado para tarefas com deadline firm
- Pode cancelar ativações individuais ou tarefas completas
- Objetivo é maximizar o número de tarefas executadas
- Tarefas podem ter importância (peso) diferentes
 - Maximiza o somatório dos pesos das tarefas executadas
- Abordagem semelhante: aumenta o período das tarefas

Garantia Dinâmica e de Melhor Esforço

Perda de deadlines na sobrecarga

- Em sobrecarga ATRASADA algumas tarefas
- Possui uma função que indica o valor de cada tarefa em função do seu instante de conclusão (time-value function)
- Objetivo é maximizar o valor total do sistema (somatório de todas as tarefas)



Garantia Dinâmica e de Melhor Esforço

Redução da Precisão na Sobrecarga

- Em sobrecarga diminui a precisão de algumas tarefas
- Objetivo é fazer o possível dentro do tempo disponível
- Exemplos:
 - Ignorar bits menos significativos de cada pixel
 - Trabalhar com amostras de áudio menos precisas
 - Alterar resolução e tamanho de imagem na tela
 - Simplificar animações
 - Usar algoritmos de controle mais simples
 - Interromper pesquisa em algoritmos de inteligência artificial

Garantia Dinâmica e de Melhor Esforço

Sistemas de Prioridade Dinâmica

- *Earliest-Deadline-First (EDF)*
 - Escalonamento preemptivo de prioridade dinâmica
 - O deadline é calculado quando a tarefa é disparada.
 - O escalonador põe para executar a tarefa com menor deadline
- *Least Slack Scheduling*
 - Escalonamento não-preemptivo de prioridade dinâmica
 - A tarefa com menor sobra (*slack*) tem maior prioridade

Referências

- **Livro de Sistemas de Tempo Real**
Jean- Marie Farines, Joni da Silva Fraga, Rômulo Silva de Oliveira. Escola de Computação'2000 - IME- USP
[http:// www. lcmi. ufsc. br/ gtr/ livro/ principal. Htm](http://www.lcmi.ufsc.br/gtr/livro/principal.Htm)
- **IEEE Computer Society, Technical Committee on Real- Time Systems (IEEE- CS TC- RTS)**
[http:// www. cs. bu. edu/ pub/ ieee- rts](http://www.cs.bu.edu/pub/ieee- rts)
- **The Concise Handbook Of Real-Time Systems.**
TimeSys Corporation, Versão 1.1, 2000.
<http://www.timesys.com>