



Aula 10

Engenharia de Sistemas Embarcados
 Prof. Abel Guilhermino
 Tópico: Sistemas de Tempo Real




Agenda


- Conceitos Gerais
- Processos de Tempo Real
 - Periódico, Aperiódicos e Esporádicos
- Escalonamento de Tempo Real
 - Tarefas Dependentes
 - Periódicas
 - Aperiódicas



Grupo de Engenharia da Computação - CIn / UFPE

Introdução

- Avanço da tecnologia tem aumentado;
- Aplicações com requisitos temporais tornam-se comuns e variam em função de:
 - Complexidade
 - Garantia no atendimento temporal
- Computação em tempo real não é o mesmo que computação rápida



Grupo de Engenharia da Computação - CIn / UFPE

Aplicações de tempo real





Tempo Real







Grupo de Engenharia da Computação - CIn / UFPE

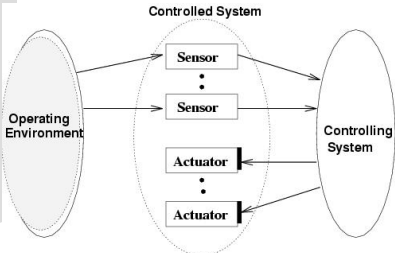

Definição

- Sistema de Tempo Real (STR) é um sistema computacional que deve reagir a estímulos oriundos do seu ambiente em prazos específicos.
- Em cada reação, o sistema de tempo real deve entregar um resultado correto dentro de um prazo específico, sob pena de ocorrer uma *falha temporal*.



Grupo de Engenharia da Computação - CIn / UFPE

Sistema de tempo real típico

Grupo de Engenharia da Computação - CIn / UFPE

Classificação dos STR

- Ponto de vista da segurança:
 - Hard real-time systems
 - O STR crítico é aquele que tem um comportamento determinístico, ou seja, o prazo para execução de uma tarefa (*deadline*) não pode ser violado. Ex: Freio ABS, Controle de Tráfego aéreo.
 - Soft real-time systems
 - É aquele que também tem o tempo como parâmetro fundamental, mas uma falha é aceitável. Ex: leitor de DVD



Grupo de Engenharia da Computação - CIn / UFPE



Hard real-time

- Subdivisões
 - Sistemas de Tempo Real Crítico **Seguros em Caso de Falha** (*fail safe*). Onde um ou vários estados seguros podem ser atingidos em caso de falha.
 - Ex: parada obrigatória de trens.
 - Sistemas de Tempo Real Crítico **Operacional em Caso de Falha** (*fail operational*), que na presença de falhas parciais, podem se degradar fornecendo alguma forma de serviço mínimo
 - (Ex: Sistema de controle de voo que após detectar a falha funciona mesmo com comportamento degradado)



Grupo de Engenharia da Computação - CIn / UFPE



Previsibilidade

- Um STR é dito *previsível* (*predictable*) no domínio lógico e temporal quando, independentemente de variações ocorrendo à nível de hardware, da carga e de falhas, o comportamento do sistema pode ser antecipado, antes de sua execução.



Grupo de Engenharia da Computação - CIn / UFPE



Garantia de Previsibilidade

- Fontes de não determinismo:
 - Arquitetura do hardware
 - Sistema Operacional
 - Linguagens de Programação
 - Ex: Laços não limitados de programa; Instruções de prefetch do processador, memória cache, etc.
- Necessidade do uso de ferramentas e metodologias que permitem verificar o comportamento do sistema e sua implementação como previsíveis em cada etapa do ciclo de desenvolvimento do STR.



Grupo de Engenharia da Computação - CIn / UFPE



Garantia de Previsibilidade

- Jensen et al. estendeu este conceito e associa a previsibilidade como uma antecipação probabilística do comportamento do sistema, baseada em estimativas ou simulações que estipulam prazos e probabilidades a serem atendidos.



Grupo de Engenharia da Computação - CIn / UFPE



Classificação dos STR

- Do ponto de vista da implementação:
 - Sistemas de Resposta Garantida (*guaranteed response system*) onde existem recursos suficientes para suportar a carga máxima e o cenário de falhas definida;
 - Adequados para os sistemas críticos
 - Sistema de Melhor Esforço (*best effort system*) quando a estratégia de alocação dinâmica de recursos se baseia em estudos probabilísticos sobre a carga esperada e os cenários de falhas aceitáveis.
 - Adequados para sistemas não críticos



Grupo de Engenharia da Computação - CIn / UFPE



Problema Tempo Real

- O problema tempo real consiste em especificar, verificar e implementar sistemas ou programas que, mesmo com recursos limitados, apresentam comportamentos previsíveis, atendendo as restrições temporais impostas pelo ambiente ou usuário.
- Solução
 - Vendo como um problema de programação concorrente, há duas abordagens usadas:
 - Síncrona
 - Esterel, StateCharts, Signal, Lustre
 - Assíncrona
 - CSP, Ada, Real-Time Concurrent C



Grupo de Engenharia da Computação - CIn / UFPE



Escalonamento de Tempo Real



Escalonamento de tempo real

- Através do escalonamento é decidido a ordem com que as tarefas/processos são executados.
- Quando as noções de tempo e de concorrência são tratadas explicitamente, técnicas de escalonamento formam o ponto central na previsibilidade de sistemas de tempo real.



Grupo de Engenharia da Computação - CIn / UFPE



Modelo de Tarefas

- Uma simples aplicação de tempo real é constituída tipicamente de várias tarefas.
- Uma tarefa de tempo real, além da correção lógica (“*correctness*”), deve satisfazer seus prazos e restrições temporais (“*timeliness*”)
- Modelo de tarefas:
 - Restrições Temporais
 - Relações de Precedência e de Exclusão



Grupo de Engenharia da Computação - CIn / UFPE



Restrições Temporais

- Todas as tarefas em tempo real tipicamente estão sujeitas a prazos “*deadlines*”.
- A princípio, uma tarefa deve ser concluída antes de seu *deadline*
- A conclusão das tarefas após seu *deadline* define dois tipos de tarefas:
 - Tarefas Críticas (*hard*)
 - A tarefa quando completada depois de seu *deadline* pode causar falhas catastróficas no STR e em seu ambiente.
 - Tarefas Não Críticas (*soft*)
 - A tarefa quando completada depois de seu *deadline*, no máximo implica em uma diminuição no desempenho do sistema



Grupo de Engenharia da Computação - CIn / UFPE



Restrições Temporais

- Outra característica temporal em STR está na regularidade das ativações das tarefas
- Os modelos de tarefas comportam 2 tipos de tarefas segundo suas frequências de ativações:
 - Tarefas Periódicas
 - quando as ativações do processamento de uma tarefa ocorrem uma só ativação por intervalo regular chamado período.
 - Tarefas Aperiódicas
 - quando a ativação de uma tarefa responde a eventos internos ou externos aleatoriamente.



Grupo de Engenharia da Computação - CIn / UFPE



Tarefas Periódicas

- Tarefas periódicas pela regularidade e pela previsibilidade são tarefas críticas.
 - C = tempo de computação da tarefa
 - P = período da tarefa
 - D = deadline
 - J = Release Jitter (pior situação de liberação de tarefa)

tempos de chegada (a_i)
 tempos de liberação (r_i)
 tempos de início (s_i)
 tempos de término (ct_i)
 "deadlines" absolutos (d_i)

Figura 2.1: Ativações de uma tarefa periódica

Greco Grupo de Engenharia da Computação - CIn / UFPE

Tarefas Aperiódicas

- Já as aperiódicas, pela falta de previsibilidade de suas ativações, normalmente são tarefas não críticas.
 - C = tempo de computação da tarefa
 - P = período da tarefa
 - D = deadline
 - J = Release Jitter (pior situação de liberação de tarefa)

tempos de chegada (a_i)
 tempos de liberação (r_i)
 tempos de início (s_i)
 tempos de término (ct_i)
 "deadlines" absolutos (d_i)

Figura 2.2: Ativações de uma tarefa aperiódica

Greco Grupo de Engenharia da Computação - CIn / UFPE

Escalonamento

- Trata do procedimento de ordenar tarefas em uma fila prontas para executar;
- O escalonador é responsável por implementar uma política de escalonamento ao ordenar para execução sobre o processador, um conjunto de tarefas
- Os escalonadores produzem escalas que se forem realizáveis (*feasible*), garantem o cumprimento das restrições temporais impostas às tarefas de tempo real.

Greco Grupo de Engenharia da Computação - CIn / UFPE

Escalonamento

- Os diferentes critérios de ordenação do conjunto de tarefas relatam a grande variedade de algoritmos. Tais algoritmos podem ser:
 - Preemptivo
 - Execução das tarefas é preemptada e retorna depois;
 - Preempção ocorre para executar tarefas de mais alta prioridade;
 - Oferece alta escalonabilidade;
 - Envolve um alto *overhead* do escalonamento devido a troca de contexto.
 - Não-preemptivo
 - Uma vez que uma tarefa inicia sua execução, essa completa a função completamente sem interrupção;
 - Oferece baixa escalonabilidade;
 - Menos *overhead* devido a menos troca de contexto.

Greco Grupo de Engenharia da Computação - CIn / UFPE

Escalonamento

- Além disso, os algoritmos de escalonamento podem ser:
 - Estáticos
 - Quando o mecanismo de ordenação das tarefas é feito tomando como base parâmetros fixos atribuídos às tarefas do conjunto de tarefas.
 - Usado para escalonamento de tarefas periódicas
 - Dinâmicos
 - São baseados em parâmetros que mudam em tempo de execução com a evolução do sistema
 - Usado para escalonamento de tarefas de ambos periódicas e aperiódicas.

Greco Grupo de Engenharia da Computação - CIn / UFPE

Teste de Escalonabilidade

- Teste de Escalonabilidade:
 - conjuntos de tarefas escalonáveis
 - conjuntos de tarefas não escalonáveis
 - teste suficiente
 - teste exato
 - teste necessário
 - Aumento da complexidade dos conjuntos de tarefas

Figura 2.4 : Tipos de testes de escalonabilidade

Greco Grupo de Engenharia da Computação - CIn / UFPE



Escalonamento de Tarefas Periódicas

UFPE
Centro de Informática

Utilização de uma Tarefa

- Serve como uma medida da ocupação do processador pela tarefa.
 - $U_i = C_i/P_i$ (se tarefa T_i for periódica)
 - $U_i = C_i/Min_i$ (se tarefa T_i for aperiódica)
 - Onde C é o tempo máximo de computação
 - P é o período
 - Min é o intervalo mínimo entre requisições
 - Utilização do Processador
 - $U = \sum U_i$ de tal forma que $U \leq m$
 - Onde m = número de processadores

Greco Grupo de Engenharia da Computação - CIn / UFPE

Algoritmos de Escalonamento

- Rate Monotonic Scheduling (RMS)
- Earliest Deadline First (EDF)
- Least Laxity First (LLF)
- Deadline Monotonic Scheduling (DMS)

| | $D_i = P_i$ | $D_i \leq P_i$ |
|----------------------------|--|---|
| Prioridade Estática | RMS Teste de Utilização do Processador | DMS Análise Exata |
| Prioridade Dinâmica | EDF/LLF Teste de Utilização do Processador ($U \leq 1$) | EDF Teste baseado na demanda do processador |

Greco Grupo de Engenharia da Computação - CIn / UFPE

Rate Monotonic Scheduling (RMS)

- O algoritmo produz escalas em tempo de execução através de escalonadores preemptivos, dirigidos a prioridade.
- Modelo de tarefas:
 - São periódicas e independentes
 - Deadline coincide com o período ($D_i=P_i$)
 - Tempo de computação (C_i) de cada tarefa é conhecido e constante (WCCT - Worst Case Computation Time)
 - O tempo de chaveamento entre tarefas é assumido como nulo.

Greco Grupo de Engenharia da Computação - CIn / UFPE

Rate Monotonic Scheduling (RMS)

- Política de Atribuição de Prioridades
 - Quanto menor o período, maior sua prioridade no conjunto.
 - Análise de escalonabilidade baseia-se no cálculo da Utilização
 - Condição Suficiente
 - Condição necessária e suficiente (períodos múltiplos, $U \rightarrow 0,69$)

$$U = \sum_i^n C_i/P_i \leq n(2^{1/n} - 1)$$

$$U = \sum_i^n C_i/P_i \leq 1$$

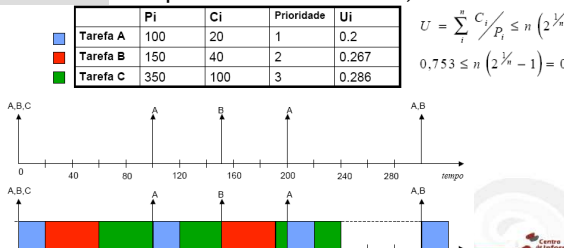
Greco Grupo de Engenharia da Computação - CIn / UFPE

Exemplo (RMS)

- Avaliar escalonabilidade de um conjunto de tarefas periódicas usando RMS;

| | P_i | C_i | Prioridade | U_i |
|----------|-------|-------|------------|-------|
| Tarefa A | 100 | 20 | 1 | 0.2 |
| Tarefa B | 150 | 40 | 2 | 0.267 |
| Tarefa C | 350 | 100 | 3 | 0.286 |


$$U = \sum_i^n C_i/P_i \leq n(2^{1/n} - 1)$$

$$0,753 \leq n(2^{1/n} - 1) = 0,779$$


Greco Grupo de Engenharia da Computação - CIn / UFPE


Earliest Deadline First (EDF)

- Escalonamento baseado em prioridades: a escala é produzida em tempo de execução por um escalonador preemptivo dirigido a prioridades.
- É um esquema de prioridade dinâmica
- Modelo de tarefas idêntica ao RMS
 - São periódicas e independentes
 - Deadline coincide com o período ($D_i=P_i$)
 - Tempo de computação (C_i) de cada tarefa é conhecido e constante (WCCT - Worst Case Computation Time)
 - O tempo de chaveamento entre tarefas é assumido como nulo.



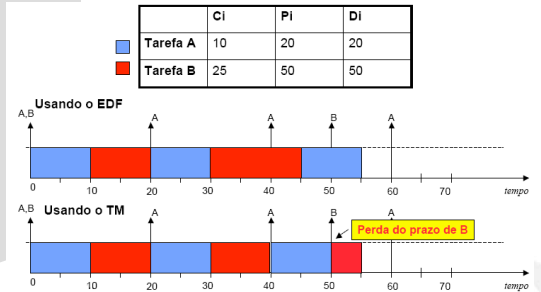

Earliest Deadline First (EDF)

- Política de Atribuição de Prioridades
 - A tarefa mais prioritária é a que tem o deadline d_i mais próximo do tempo atual.
 - A cada chegada de tarefa, a fila de pronto é reordenada considerando a nova distribuição de prioridades
 - O novo valor de deadline absoluto é determinado considerando o número de períodos que antecede a atual ativação (K): $d_{ik} = kP_i$.
- Teste de escalonabilidade
 - Condição necessária e suficiente $U = \sum_i \frac{C_i}{P_i} \leq 1$.




Exemplo (EDF)

| | C_i | P_i | D_i |
|----------|-------|-------|-------|
| Tarefa A | 10 | 20 | 20 |
| Tarefa B | 25 | 50 | 50 |

Deadline Monotonic Scheduling(DMS)

- Estende o modelo de tarefas do RMS.
- Modelo de tarefas:
 - São periódicas e independentes
 - Deadline coincide com o período ($D_i=P_i$)
 - Em substituição a: ($D_i \leq P_i$)
 - Tempo de computação (C_i) de cada tarefa é conhecido e constante (WCCT - Worst Case Computation Time)
 - O tempo de chaveamento entre tarefas é assumido como nulo.



Deadline Monotonic Scheduling(DMS)


- Política de Atribuição de Prioridades
 - Determina prioridade baseada em d_i .
 - Menor o d_i maior a prioridade.
 - Teste de escalonabilidade
 - Condição Suficiente

$$U = \sum_i \frac{C_i}{D_i} \leq n (2^{1/n} - 1)$$

Diferença em relação ao RMS

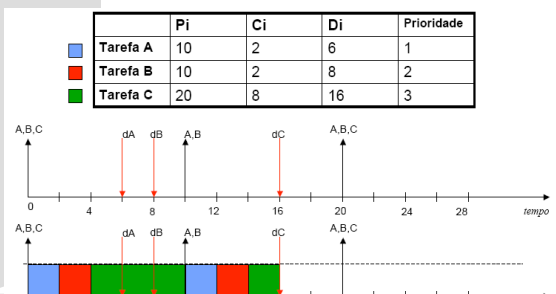
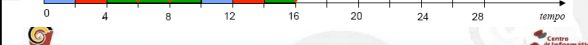
Exemplo:

| tarefas periódicas | C_i | P_i | D_i | p_i |
|--------------------|-------|-------|-------|-------|
| tarefa A | 2 | 10 | 6 | 1 |
| tarefa B | 2 | 10 | 8 | 2 |
| tarefa C | 8 | 20 | 16 | 3 |



Deadline Monotonic Scheduling(DMS)

| | P_i | C_i | D_i | Prioridade |
|----------|-------|-------|-------|------------|
| Tarefa A | 10 | 2 | 6 | 1 |
| Tarefa B | 10 | 2 | 8 | 2 |
| Tarefa C | 20 | 8 | 16 | 3 |




Tarefas Dependentes





Tarefas Dependentes

- Até o presente momento as tarefas eram apresentadas como independentes.
- Em um ambiente multitarefa o compartilhamento de recursos é implícito e determina alguma forma de relação de exclusão entre tarefas.
- Comunicações entre tarefas, no mesmo processador, podem se dar através de variáveis compartilhadas, usando mecanismos como: **semáforos e monitores** para implementar a **exclusão mútua** entre as tarefas.



Grupo de Engenharia da Computação - CIn / UFPE



Exemplo de Problema


Processo A

LOAD x,Ra
ADD 1,Ra
STORE Ra,x


Processo B

LOAD x,Rb
SUB 1,Rb
STORE Rb,x

| Processo | comando | x | Ra | Rb |
|----------|------------|---|----|----|
| A | LOAD x,Ra | 2 | 2 | * |
| A | ADD 1,Ra | 2 | 3 | * |
| B | LOAD x,Rb | 2 | * | 2 |
| B | SUB 1, Rb | 2 | * | 1 |
| A | STORE Ra,x | 3 | 3 | * |
| B | STORE Rb,x | 1 | * | 1 |




Grupo de Engenharia da Computação - CIn / UFPE




Soluções

- Exclusão Mútua
 - Solução mais simples para tarefas dependentes;
 - Impede que dois ou mais processos acessem um mesmo recurso simultaneamente;
 - Enquanto um processo estiver acessando determinado recurso, todos os demais processos que queiram acessá-lo deverão esperar pelo término da utilização do recurso.
 - Soluções para exclusão mútua:
 - Test-and-set
 - Algoritmo de Dekker
 - Algoritmo de Peterson




Grupo de Engenharia da Computação - CIn / UFPE




Semáforos

- Mecanismo de sincronização não estruturado que permite implementar, de forma simples, a exclusão mútua entre processos.
- Um semáforo é uma variável inteira, não negativa, que só pode ser manipulada por duas instruções: DOWN e UP.
 - As instruções UP e DOWN são indivisíveis, ou seja, a execução dessas instruções não pode ser interrompida.




Grupo de Engenharia da Computação - CIn / UFPE




Monitores

- Mecanismo de sincronização estruturado de alto nível tendo em vista que são implementados pelo compilador.
- A comunicação do processo com o monitor é feita unicamente através de chamadas a seus procedimentos e dos parâmetros passados.



Grupo de Engenharia da Computação - CIn / UFPE






Escalonamento de Tarefas Aperiódicas





Tarefas Aperiódicas

- As tarefas periódicas são assumidas como críticas, necessitando de garantias de tempo de projetos para situações de pior caso.
- As aperiódicas podem envolver diferentes requisitos temporais: críticos, não críticos, e ainda sem requisitos temporais.
 - Basicamente, tarefas que necessitam um bom tempo médio de resposta.
- Solução:
 - Abordagem Híbrida baseada nas sobras de processador.




Grupo de Engenharia da Computação - CIn / UFPE




Tarefas Aperiódicas

- As sobras nas escalas de carga periódica, são determinadas estaticamente, em tempo de projeto, e posteriormente, em tempo de execução, são atribuídas ao processamento aperiódico usando o conceito de servidor.
 - Background Server
 - Polling Server
 - Deferrable Server
 - Priority Exchange Server
 - Sporatic Server




Grupo de Engenharia da Computação - CIn / UFPE




Background Server (BS)

- Este servidor é extremamente simples.
- A idéia corresponde em atender as requisições aperiódicas quando a fila de prontos envolvendo tarefas periódicas está vazia.
- A determinação de prioridades nesta abordagem é feita atribuindo, segundo o RMS, as prioridades mais altas para as tarefas periódicas.
- Como consequência, BS apresenta tempos de resposta muito altos para cargas aperiódicas.

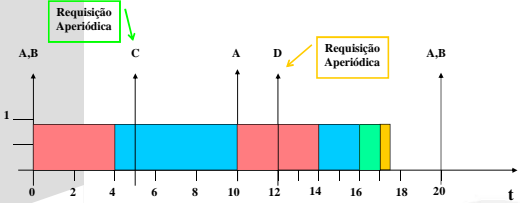


Grupo de Engenharia da Computação - CIn / UFPE




Ex. BS

| Tarefas | Tempo de Computação C_i | Período P_i | Deadline D_i | Prioridade p_i |
|-----------|---------------------------|---------------|----------------|------------------|
| T_A | 4 | 10 | 10 | 2 |
| T_B | 8 | 20 | 20 | 3 |
| Aperiod.C | 1 | - | - | - |
| Aperiod.D | 0,5 | - | - | - |




Podemos observar neste exemplo que as tarefas aperiódicas só serão executadas depois de toda a carga periódica, causando um maior tempo de resposta.

From: Prof. Sérgio Cavalcanti, UFPE




Grupo de Engenharia da Computação - CIn / UFPE




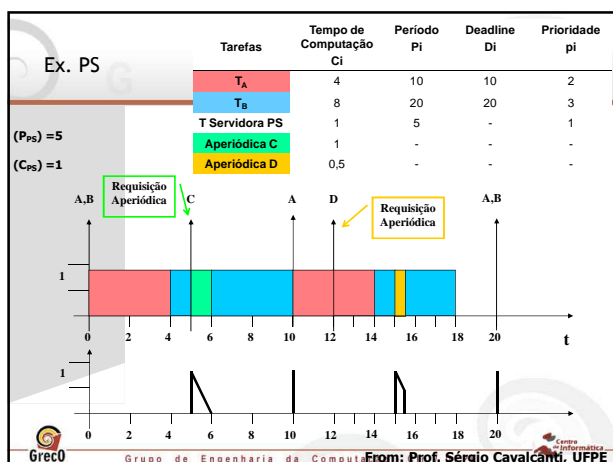
Polling Server (PS)

- Consiste na definição de uma tarefa periódica para atender a carga aperiódica.
- Esta tarefa servidora tem período (P_{PS}), prioridade e capacidade de computação (C_{PS}) definidos.
- Em cada ativação, a tarefa servidora executa as tarefas aperiódicas pendentes durante seu tempo de computação.
- A prioridade das tarefas é atribuída com base no RMS.
- Quando houver requisições aperiódicas pendentes, a tarefa PS se suspende até sua nova chegada, no início do próximo período
 - Neste caso sua carga é entregue para a execução de tarefas periódicas pendentes.



Grupo de Engenharia da Computação - CIn / UFPE





Deferrable Server (DS)

- DS cria tarefa aperiódica, geralmente de alta prioridade, para servidores de requisições aperiódicas;
- Recebe uma prioridade segundo RMS
- Diferente do PS e DS, conserva sua capacidade (C_{DS}), mesmo quando não existir requisições durante a ativação da tarefa DS.
- No início de cada período da tarefa servidora, sua capacidade de processamento é restaurada

From: Prof. Sérgio Cavalcanti, UFPE

