

Projeto de Sistemas de Tempo Real

Centro de Informática - Universidade Federal de Pernambuco
Engenharia da Computação

Kiev Gama

kiev@cin.ufpe.br

Slides elaborados pelo professor Marcio Cornélio

O autor permite o uso e a modificação dos *slides* para fins didáticos



UNIVERSIDADE FEDERAL DE PERNAMBUCO

Sistemas de tempo real

- São sistemas que monitoram e controlam seu ambiente.
- Associados, inevitavelmente, a dispositivos de hardware
 - **Sensores**: Coletam dados do ambiente do sistema;
 - **Atuadores**: Mudam (de alguma maneira) o ambiente do sistema.
- O tempo é crítico. Sistemas de tempo real DEVEM responder dentro de tempos especificados.

Definição

- Sistema de tempo real
 - Funcionamento correto do sistema depende dos resultados produzidos pelo sistema e do tempo em que estes resultados são produzidos.

Definição

- **Rígido (*hard*)**
 - Perda de um prazo estabelecido é uma falha
 - Marcapasso, sistema de controle de motores de carros
- **Leve (*soft*)**
 - Operação degradada se resultados não forem obtidos de acordo com requisitos de temporização especificados
 - A utilidade de um resultado após um deadline é degradada, levando à degradação da qualidade do sistema

Sistemas estímulo-resposta

- Dado um estímulo, o sistema deve produzir uma resposta dentro de um tempo especificado.
- **Estímulos periódicos.** Estímulos que ocorrem em intervalos de tempo previsíveis
 - Por exemplo, um sensor de temperatura pode ser estimulado 10 vezes por segundo.
- **Estímulos aperiódicos.** Estímulos que ocorrem em tempos imprevisíveis
 - Por exemplo, uma falha de sistema de energia pode disparar uma interrupção que deve ser processada pelo sistema.

Considerações de arquitetura

- Resposta a estímulos em tempos diferentes
- Estímulo recebido e controle transferido para tratador correto
- Uso de processos concorrentes cooperativos
- Plataforma: sistema operacional de tempo- real

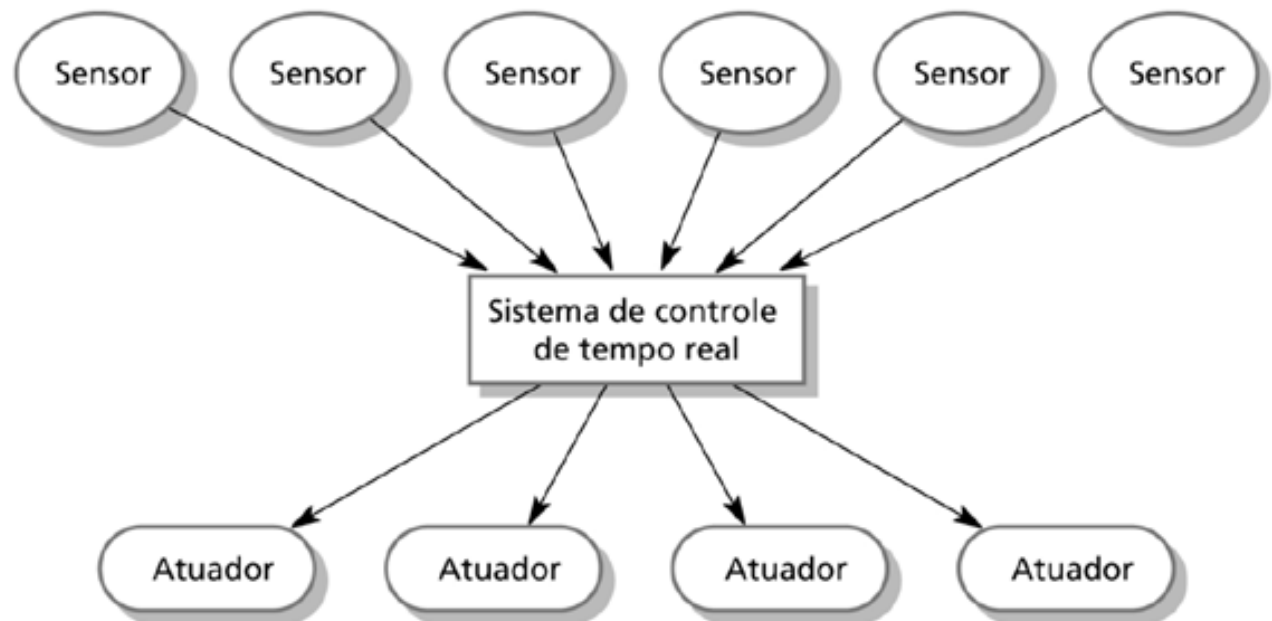
Elementos de sistema

- Processos de controle de sensores
 - Coletam informações de sensores e podem armazenar informações coletadas em resposta a um estímulo de sensor.
- Processador de dados
 - Realiza o processamento de informações coletadas e calcula a resposta de sistema.
- Processos de controle de atuadores
 - Geram sinais de controle para os atuadores.

Um modelo de sistema de tempo real

Figura 15.1

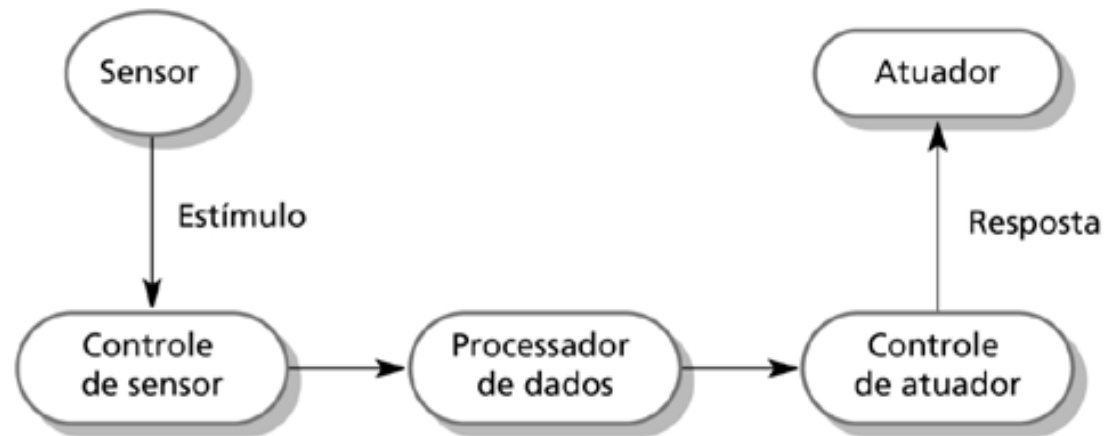
Modelo geral de um sistema de tempo real.



Processos Sensor-Atuador

Figura 15.2

Processos de controle de sensor-atuador.



Programação de tempo real (rígido)

- Linguagem assembly para assegurar que os prazos finais (*deadlines*) sejam atendidos
- Linguagens como C
 - Programas eficientes sejam escritos
 - Sem construções para apoiar a concorrência ou o gerenciamento de recursos compartilhados.

Java como uma linguagem de tempo real

- A linguagem Java apóia a concorrência leve (threads e métodos sincronizados) e pode ser usada para alguns sistemas de tempo real leves
- Versões iniciais do Java (ex: Java 2.0) não eram adequadas para programação de tempo real pesada
- Versões de tempo real em Java que estão disponíveis atualmente, solucionam problemas como:
 - Impossibilidade de especificar o tempo de execução do *thread*;
 - *Timing* diferente em máquinas virtuais diferentes;
 - Coleta de lixo (*garbage collection*) incontrolável;
 - Impossibilidade de descobrir tamanhos de filas para recursos compartilhados;
 - Impossibilidade de acessar hardware do sistema;
 - Impossibilidade de fazer análise de espaço ou de *timing*.
- Real time specification for Java
- Javolution
 - Biblioteca para aplicações críticas e com requisitos de real time

Linguagens para tempo real

- Real time Java
 - Duas novas threads
 - Real-time threads (sem inversão de prioridade)
 - No-heap real-time threads (não há interrupção pelo coletor de lixo)
 - Dois tipos de áreas de memória
 - Memória imortal: objetos destruídos apenas com término do programa. Não é alvo de coleta de lixo
 - Memória de escopo: usada apenas enquanto um processo executa uma seção (escopo) de um programa (método, por exemplo). Objetos automaticamente destruídos quando saem do escopo. Não é alvo de coleta de lixo.

Projeto de sistemas

- Projetar hardware e software associados ao sistema e particionar funções para hardware ou software.
- Componentes devem ser projetados independente de como serão implementados (se em hardware ou em software)
- Em geral, *bottom-up*
 - Hardware
 - Software de apoio
 - Temporização

Processo de projeto de sistemas de tempo real (RT)

- Identificar estímulos e respostas
- Para cada estímulo e resposta, identificar as restrições de tempo
- Escolha de plataforma de execução
- Agregar o estímulo e o processamento de resposta em processos concorrentes
- Projetar algoritmos para processar cada classe de estímulo e resposta.
- Projetar um sistema de escalonamento
 - processos iniciados a tempo de cumprir os prazos

Restrições de timing

- Simulação e experimentos extensos para assegurar que as restrições são atendidas pelo sistema
 - Estímulos aperiódicos
- Certas estratégias de projeto, tais como projeto orientado a objetos, não podem ser usadas por causa do overhead adicional envolvido.
- Características de linguagem de programação de baixo nível têm de ser usadas por razões de desempenho.

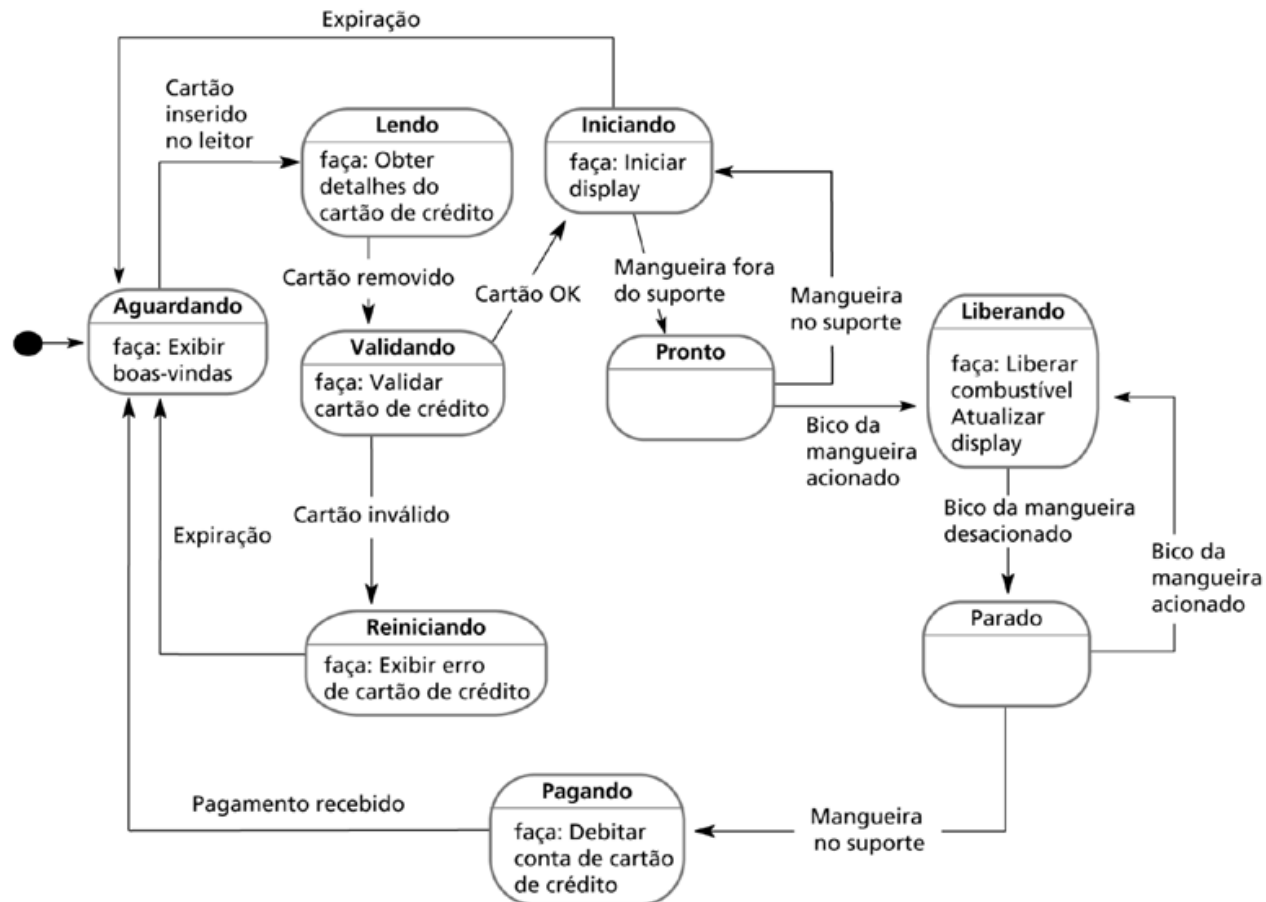
Modelagem de sistema de tempo real

- Resposta a eventos com intervalos irregulares. Mudança de estado do sistema como consequência
- Modelo de máquinas de estados finitos podem ser usadas para modelagem de sistemas de tempo real

Modelo de estados de uma bomba de combustível

Figura 15.3

Modelo de máquina de estado de uma bomba de combustível.



Sistemas operacionais de tempo real

- Responsáveis pelo gerenciamento do processo e pela alocação de recursos (processador e memória).
- Podem ser baseados em um kernel padrão que é usado sem modificação, ou modificado por uma aplicação particular.
- Normalmente não inclui recursos, tais como gerenciamento de arquivos
 - RTOS's para sistema de grande porte (controle de processo, telecomunicações) podem ter gerenciamento de armazenamento de disco e defeitos

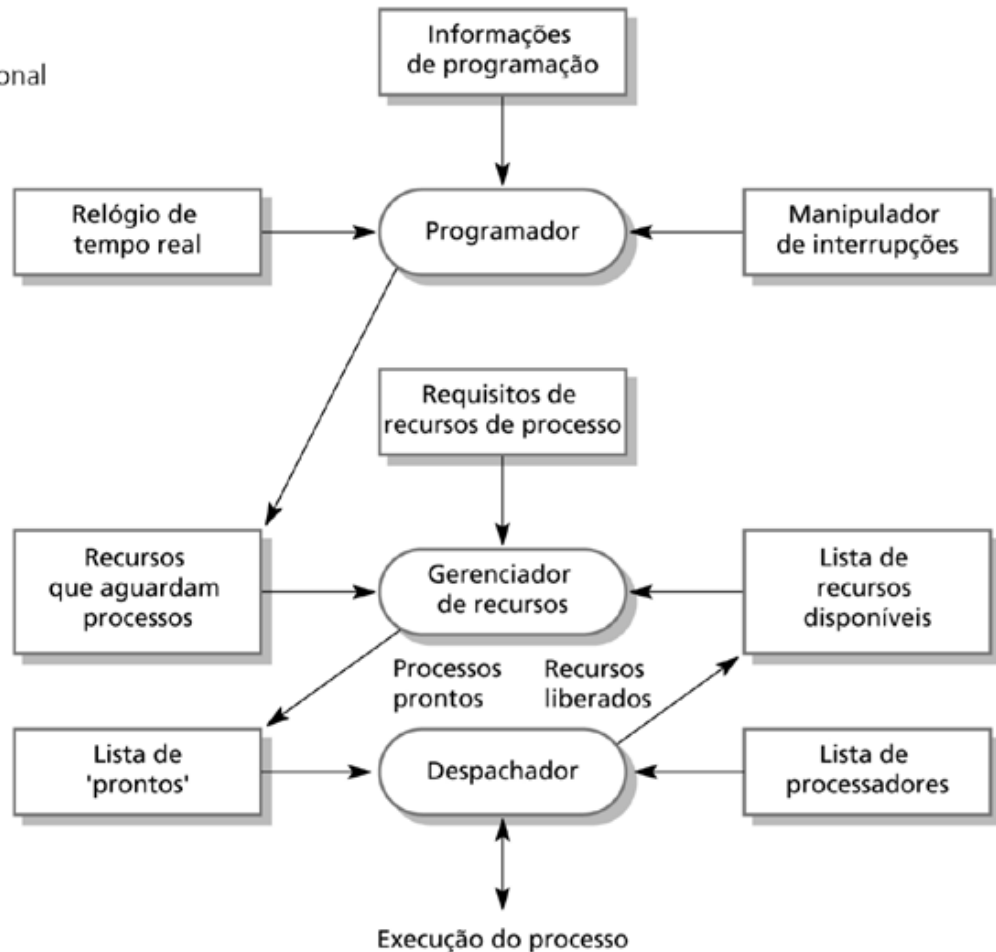
Componentes de sistemas operacionais

- Relógio de tempo real
 - Fornece informações para programar processos periodicamente.
- Tratador de interrupções
 - Gerencia solicitações aperiódicas por serviços.
- Programador (escalonador)
 - Escolhe o próximo processo a ser executado.
- Gerenciador de recursos
 - Aloca recursos de memória e processador.
- Despachador
 - Inicia a execução de processo.

Componentes de sistema operacional de tempo real

Figura 15.4

Componentes de um sistema operacional de tempo real.



Gerenciamento de processos

- Tratamento de eventos externos dentro do prazo de processamento
- Processos com prioridades diferentes
- **Nível de interrupção.**
 - Nível de prioridade mais alto. É alocado para processos que necessitam de uma resposta muito rápida.
- **Prioridade nível de relógio.**
 - Nível de prioridade é alocado aos processos periódicos.
- Níveis adicionais de prioridade podem ser definidos.

Processos periódicos

- Na maioria dos sistemas de tempo real, haverá várias classes de processos periódicos, com diferentes
 - Períodos (o tempo entre execuções)
 - Tempos de execução
 - Deadlines (o tempo pelo qual o processamento deve ser realizado).

Gerenciamento de processos

- Está relacionado ao gerenciamento do conjunto de processos concorrentes.
- Os processos periódicos são executados em intervalos de tempo pré-especificados.
- O RTOS usa o relógio de tempo real para determinar quando executar um processo levando em conta:
 - O período de processo – tempo entre execuções.
 - Deadline de processo – o tempo pelo qual o processamento deve ser realizado.

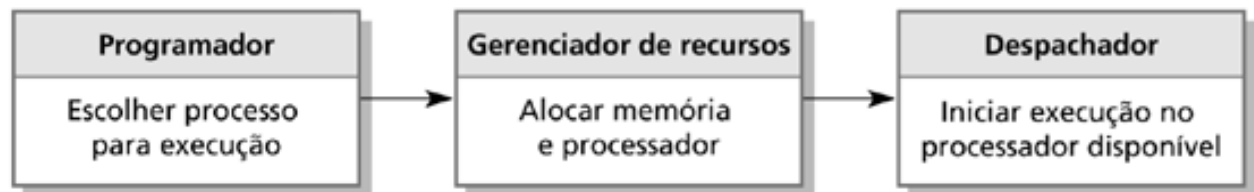
Chaveamento de processos

- O programador escolhe o próximo processo para ser executado pelo processador. Isso depende de uma estratégia de escalamento que pode levar em conta a prioridade do processo.
- O gerenciador de recursos aloca memória e um processador para o que o processo seja executado.
- O despachador toma o processo da 'lista de prontos', o carrega em um processador e inicia sua execução.

Gerenciamento de processos

Figura 15.5

Ações de RTOS necessárias para iniciar um processo.



Serviço de interrupção

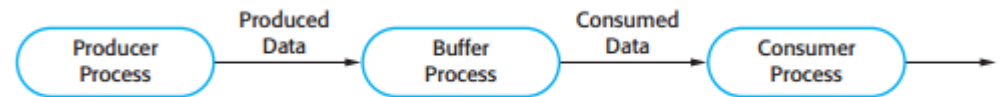
- O controle é transferido automaticamente para uma posição de memória predeterminada.
- Essa posição contém uma instrução para pular para uma rotina de serviço de interrupção.
- As interrupções adicionais são desabilitadas, a interrupção é atendida e o controle retorna ao processo interrompido.
- Rotinas de serviço de interrupção DEVEM ser curtas, simples e rápidas.

Estratégias de programação

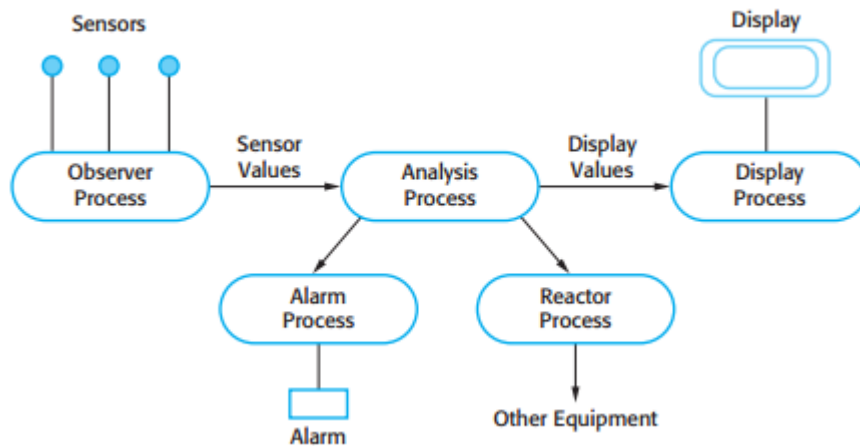
- Programação não preemptiva
 - Uma vez que o processo foi programado para execução, ele será executado até a conclusão ou até que seja interrompido por alguma razão (por exemplo, aguardando por E/S).
- Programação preemptiva
 - A execução de um processo pode ser interrompida se um processo de maior prioridade solicitar serviço.
- Algoritmos de escalamento
 - *Round-robin*;
 - *Rate monotonic*;
 - Deadline mais curto primeiro.

Padrões arquiteturais de sistemas de tempo real

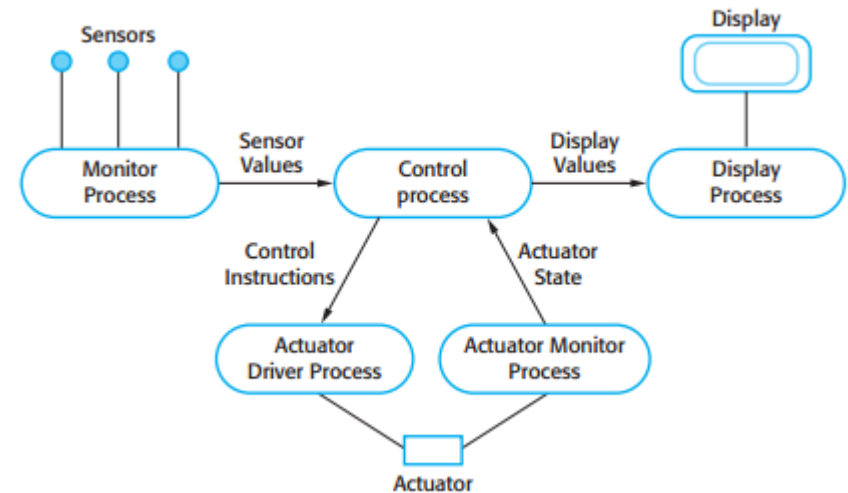
Process Pipeline



Observe and react



Environmental control



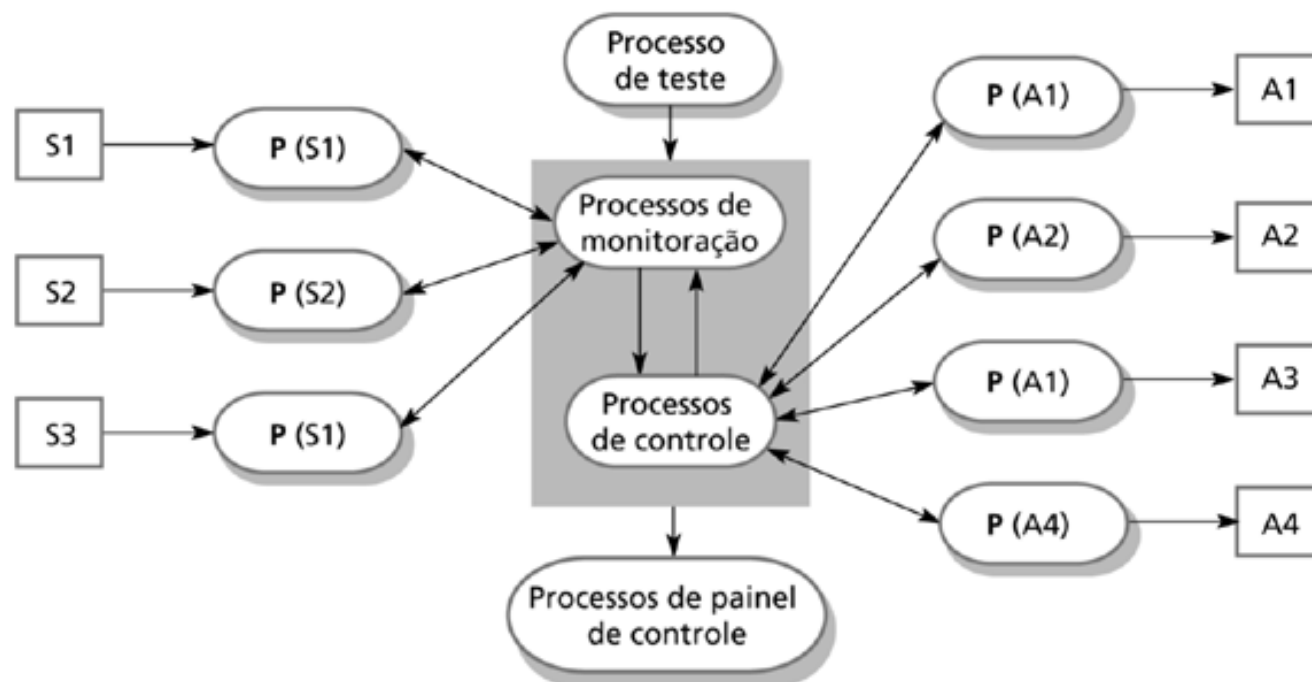
Sistemas de monitoração e controle

- Verificam continuamente os sensores e executam ações dependendo da leitura dos sensores.
- Os sistemas de monitoração examinam os sensores e enviam os resultados.
- Os sistemas de controle controlam continuamente os atuadores de hardware dependendo do valor dos sensores associados.

Arquitetura genérica

Figura 15.6

Arquitetura genérica para um sistema de monitoração e controle.



Sistema de alarme contra roubos

- É necessário um sistema que monitore os sensores das portas e janelas para detectar a presença de intrusos em um edifício.
- Quando um sensor indica uma interrupção, o sistema liga as luzes em torno da área e chama a polícia automaticamente.
- O sistema deve incluir alimentação para operação sem a fonte principal de energia.

Sistema de alarme contra roubos

- Sensores
 - Detectores de movimento, sensores de janela, sensores de porta;
 - 50 sensores de janela, 30 sensores de porta e 200 detectores de movimento;
 - Sensor de queda de tensão.
- Ações
 - Quando um intruso for detectado, a polícia é chamada automaticamente;
 - As luzes são ligadas nas salas com sensores ativos;
 - Um alarme audível é acionado;
 - O sistema chaveia automaticamente para a energia de *backup* quando uma queda de tensão é detectada.

Processo de projeto de sistema de tempo real

- Identificar estímulos e respostas associadas.
- Definir as restrições de timing associadas a cada estímulo e resposta.
- Alocar funções de sistema para processos concorrentes.
- Projetar algoritmos para processamento de estímulos e geração de respostas.
- Projetar um sistema de programação que assegure que os processos serão sempre programados para atender aos seus deadlines.

Estímulos a serem processados

- Falha de energia
 - Gerado aperiodicamente por um monitor de circuito. Quando recebido, o sistema deve chavear para a energia de backup no espaço de 50 ms.
- Alarme de intrusos
 - Estímulos gerados pelos sensores de sistema. A resposta a esse estímulo é chamar a polícia, ligar as luzes do edifício e o alarme audível.

Requisitos de *timing*

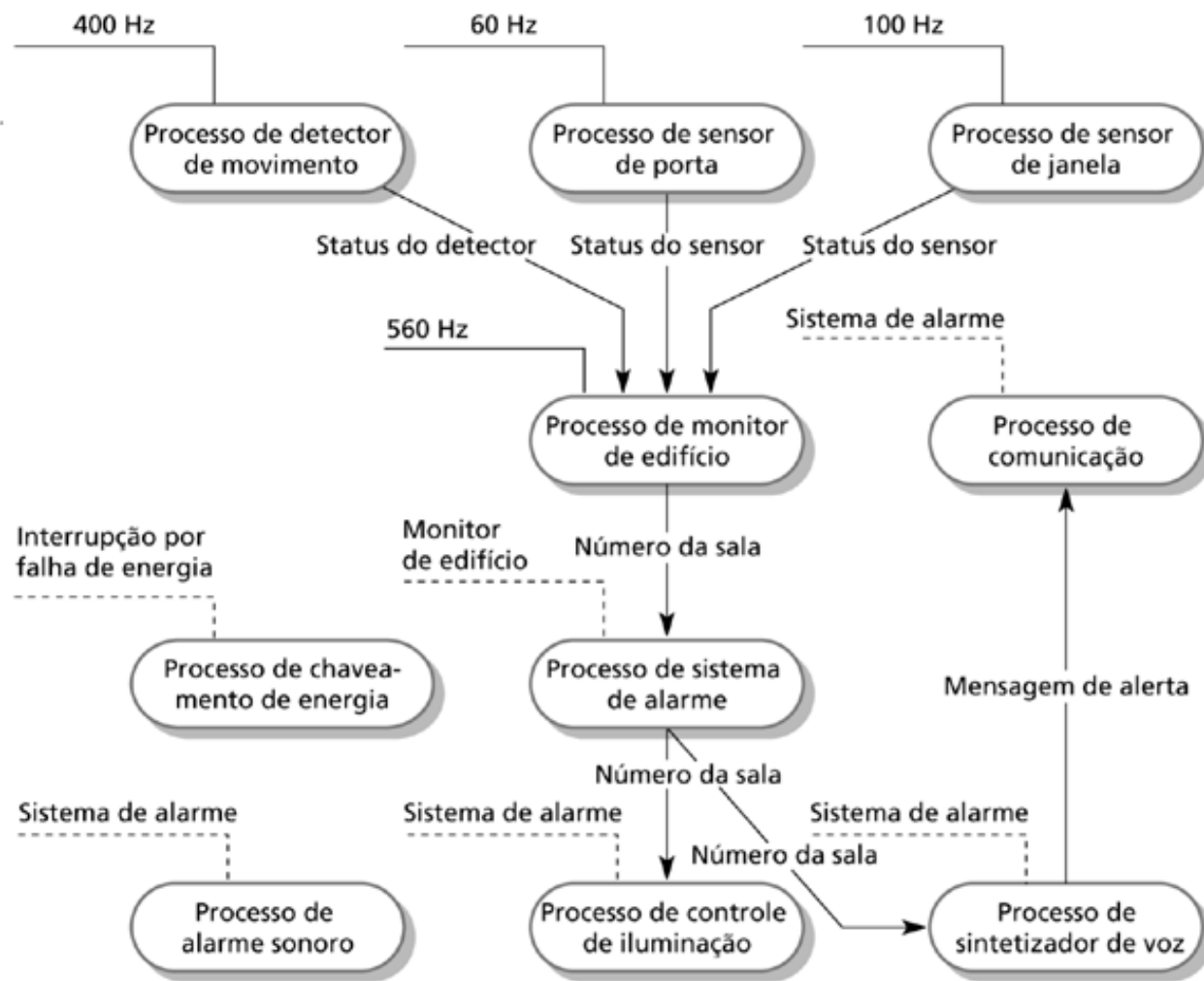
Tabela 15.1 Requisitos de tempo de estímulo–resposta

| Estímulo–resposta | Requisitos de tempo |
|----------------------------------|---|
| Interrupção por falha de energia | O chaveamento para a energia de reserva deve ser concluído dentro de um prazo de 50 ms. |
| Alarme de porta | Cada alarme de porta deve ser lido duas vezes por segundo. |
| Alarme de janela | Cada alarme de janela deve ser lido duas vezes por segundo. |
| Detector de movimento | Cada detector de movimento deve ser lido duas vezes por segundo. |
| Alarme sonoro | Um alarme sonoro deve ser acionado dentro de meio segundo após um alarme ter sido acionado por um sensor. |
| Interruptor de luzes | As luzes devem ser ligadas dentro de meio segundo após um alarme ter sido acionado por um sensor. |
| Comunicações | O chamado para a polícia deve ser iniciado dentro de 2 segundos após um alarme ter sido exposto por um sensor. |
| Sintetizador de voz | Uma mensagem sintetizada deve estar disponível dentro de 4 segundos após um alarme ter sido acionado por um sensor. |

Processos do sistema de alarme contra roubos

Figura 15.7

Arquitetura do processo de sistema de alarme de intrusos.



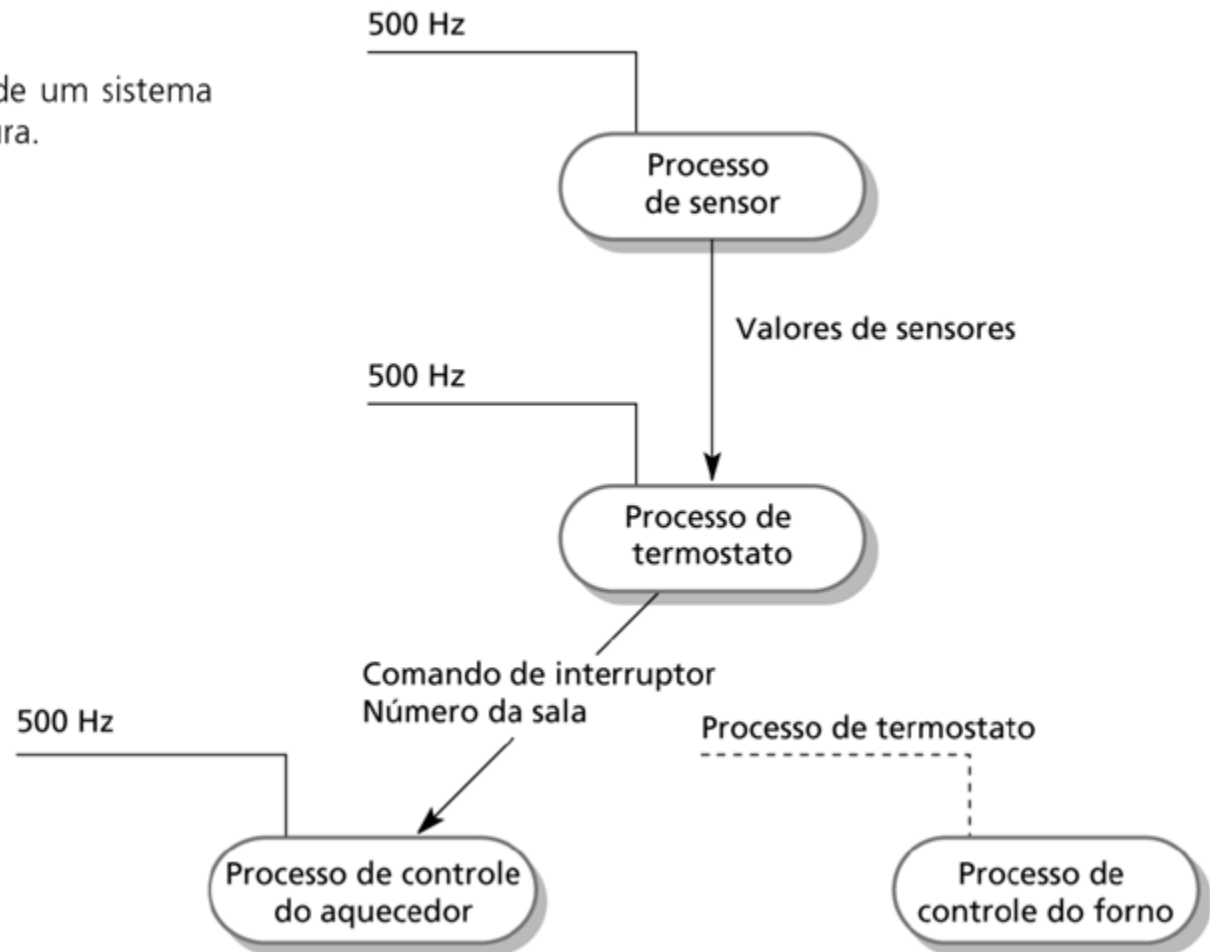
Sistemas de controle

- Um sistema de alarme contra roubos é primariamente um sistema de monitoração. Ele coleta dados dos sensores mas não controla atuadores de tempo real.
- Os sistemas de controle são similares mas, em resposta aos valores dos sensores, **o sistema envia sinais de controle para os atuadores.**
 - Ex: sistema que monitora temperatura, e liga e desliga aquecedores.

Um sistema de controle de temperatura

Figura 15.9

Arquitetura do processo de um sistema de controle de temperatura.



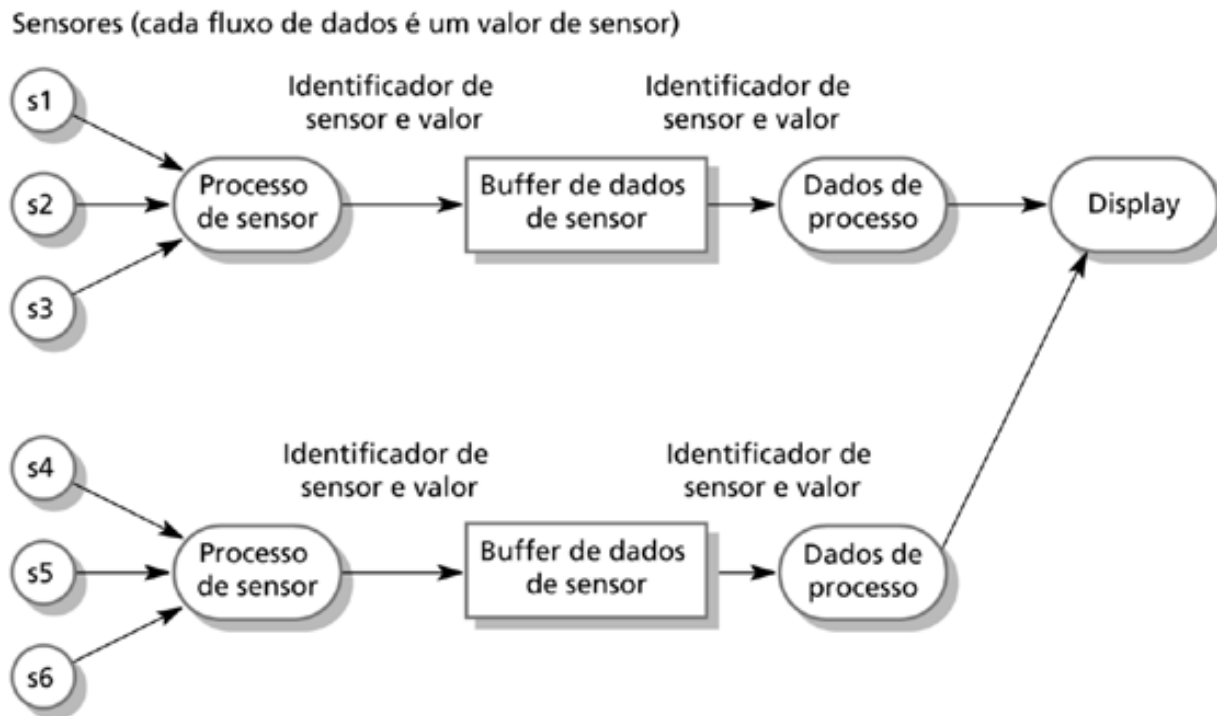
Sistemas de aquisição de dados

- Coleta dados de sensores para processamento e análise posterior.
- Processos de coleta de dados e processos de processamento podem ter períodos e deadlines diferentes.
- A coleta de dados pode ser mais rápida que o processamento, por exemplo, a coleta de informações sobre uma explosão.
- Buffers circulares ou em anel são mecanismos para amenizar as diferenças de velocidade.

Arquitetura de aquisição de dados

Figura 15.10

Arquitetura genérica de sistemas de aquisição de dados.



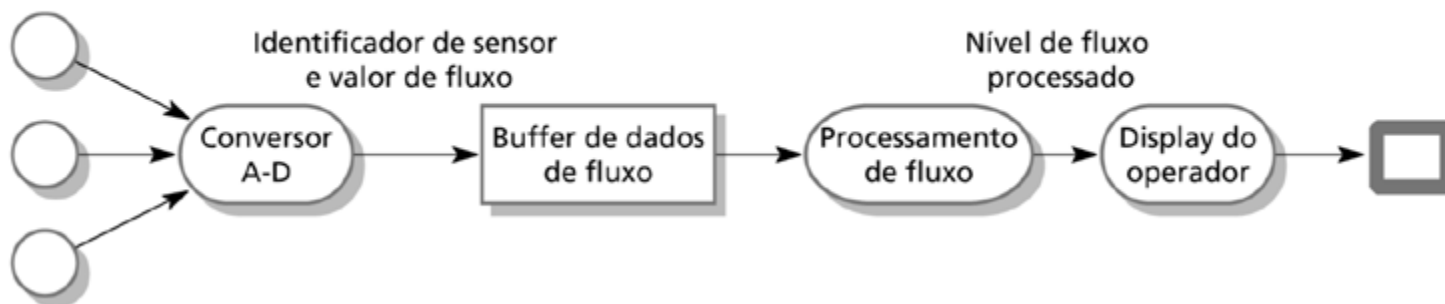
Coleta de dados de reator

- Um sistema coleta dados de um conjunto de sensores que monitora o fluxo de nêutrons de um reator nuclear.
- Os dados de fluxo são colocados em um buffer em anel para processamento posterior.
- O buffer em anel em si, é implementado como um processo concorrente, de tal maneira que os processos de coleta e de processamento podem ser sincronizados.

Monitoração de fluxo de reator

Figura 15.11 Aquisição de dados de fluxo de nêutrons.

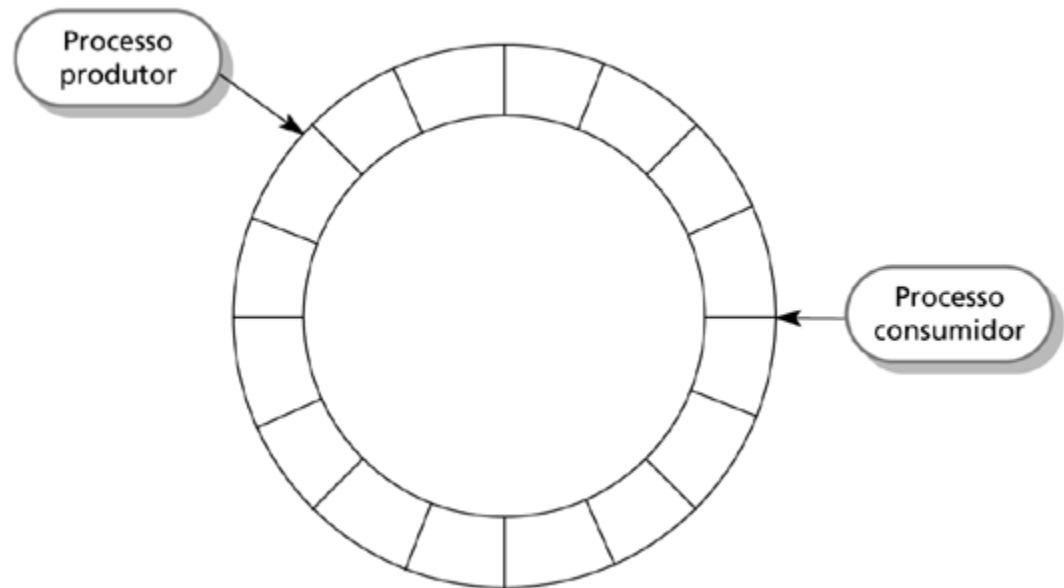
Sensores de fluxo de nêutrons



Um *buffer* em anel

Figura 15.12

Um buffer em anel para aquisição de dados.



Exclusão mútua

- Processos de produtor coletam dados e os adicionam no buffer; já os processos de consumidor retiram dados do buffer e os tornam disponíveis.
- Processos de produtor e consumidor devem ser mutuamente excluídos do acesso ao mesmo elemento.
- O buffer deve assegurar que os processos de produtor não tente adicionar informações a um buffer cheio e o consumidor não retire informações de um buffer vazio.

Resumindo

- Sistemas de tempo real estão ligados a monitoramento e controle
- O tempo é fator crítico
 - Hard e soft real time
- Software envolvido neste sistema necessita características específicas
 - Ex: SO com suporte a multitarefa é necessário
- Projeto de Sistemas de Tempo real
 - Devem levar em conta as restrições de tempo
 - Padrões Arquiteturais

Referências

- SOMMERVILLE, I. Engenharia de Software. 9ª. Ed. São Paulo: Pearson Education, 2011
– Capítulo 20