
UNIP – Universidade Paulista
Campus Tatuapé – SP
Ciência da Computação
Engenharia de Software

Projeto de Software de Tempo Real

Prof. Marcelo Nogueira

Projeto de software de tempo real

- Projetando sistemas de software embutido cujo comportamento é guiado pelo tempo

Objetivos

- Compreender o conceito de um sistema de tempo real e por que esses sistemas são, em geral, implementados como um conjunto de processos concorrentes
- Conhecer um processo de projeto para sistemas de tempo real
- Compreender o papel de um executivo de tempo real
- Compreender arquiteturas comuns de processos para sistemas de monitoramento e controle e sistemas de aquisição de dados

Tópicos

- Projeto de sistemas
- Executivos de tempo real
- Sistemas de monitoramento e controle
- Sistemas de aquisição de dados

Sistemas de tempo real

- Sistemas para acompanhar e controlar seu ambiente
- Inevitavelmente associados com seus componentes de hardware
 - Sensores: Coletor de dados de seu ambiente de sistema
 - Atuadores: Mudança (de alguma forma) de seu ambiente de sistema
- Sistemas de tempo real devem responder dentro de um determinado tempo

Definição

- Um sistema de tempo real é um sistema cujo funcionamento correto depende dos resultados produzidos por ele e do tempo em que esses resultados são produzidos
- Um sistema de tempo real leve é aquele cuja operação será degradada, se os resultados não forem produzidos de acordo com os requisitos de tempo especificados
- Um sistema de tempo real rígido é aquele cuja operação será incorreta, se os resultados não forem produzidos de acordo com a especificação de tempo

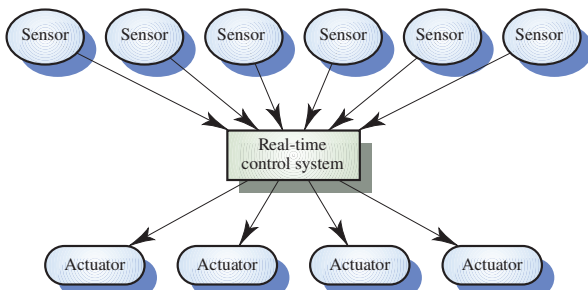
Sistema de Estímulos/Respostas

- De acordo com o estímulo, o sistema deve produzir uma resposta dentro do prazo especificado
- Estímulo periódico. Ocorrem em intervalos de tempo previsíveis
 - Por exemplo, examinar o sensor a cada 50 milissegundos e reagir de acordo com o valor (estímulo) desse sensor
- Estímulo aperiódico. Ocorrem irregularmente
 - Por exemplo, uma interrupção que indica que uma transferência de E/S foi concluída

Considerações arquiteturais

- Um sistema de tempo real tem de responder a estímulos que ocorrem em diferentes momentos, e sua arquitetura deve ser organizada de modo que o controle seja transferido ao manipulador apropriado para aquele estímulo
- Isso é impraticável em programas seqüenciais
- Os sistemas de tempo real são normalmente projetados como um conjunto de processos concorrentes e cooperantes

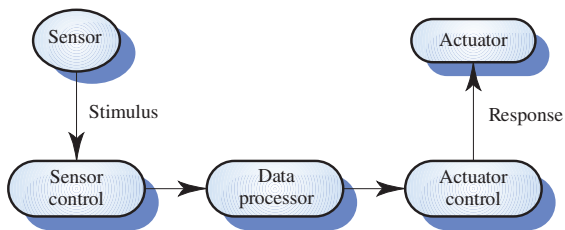
Modelo geral de um sistema de tempo real



Elementos do sistema

- Controle de sensor
 - Coleta informações dos sensores.
- Processador de dados
 - Realiza o processamento das informações coletadas e calcula o sistema de resposta
- Controle de atuador
 - Gera sinais de controle para o atuador

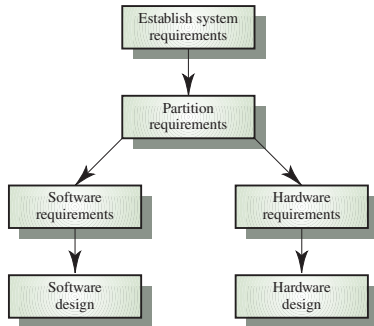
Processos de controle sensor/atuador



Projeto de sistemas

- Projeto de sistema envolve decidir quais capacidades do sistema devem ser implementadas em softwares e quais em hardware
- Decisões de projetos devem ser feitas com base em requisitos de sistema não-funcionais
- Hardwares apresentam desempenho muito melhor, porém fornecer desempenho de hardware significa que o projeto pode ser estruturado para ter facilidade de adaptação e as considerações de desempenho podem ficar em segundo lugar

Projeto de Hardware e software



©Ian Sommerville 2000

Engenharia de Software 6ª edição, Capítulo 13

Slide 13

Processo de projeto de sistemas de tempo real

- Identificar os estímulos que o sistema deve processar e as respostas associadas
- Identificar as restrições de tempo para cada estímulo e cada resposta associada
- Agregar o processamento de estímulo e de resposta em uma série de processos simultâneos. Um processo pode ser associado a cada classe de estímulo e resposta

©Ian Sommerville 2000

Engenharia de Software 6ª edição, Capítulo 13

Slide 14

Processo de projeto de sistemas de RT

- Projetar algoritmos para cada estímulo e resposta. Para indicar a quantidade de processamento requerida e do tempo necessário
- Projetar um sistema de escalonamento que assegure que os processos sejam iniciados a tempo de atender a seus prazos
- Integrar o sistema sob o controle de um executivo de tempo real

©Ian Sommerville 2000

Engenharia de Software 6ª edição, Capítulo 13

Slide 15

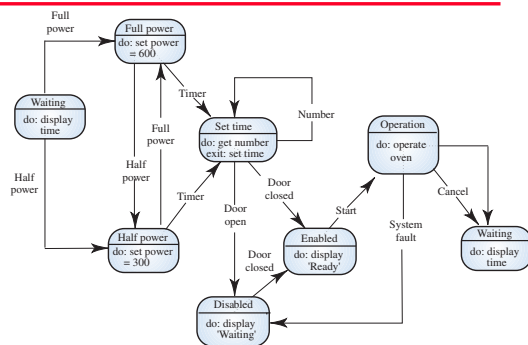
Restrições de tempo

- São necessárias extensas avaliações e simulações, a fim de verificar se o sistema atenderá às suas restrições de tempo
- Pode significar que certas estratégias como projeto orientado a objeto não podem ser utilizados devido à sobrecarga adicional envolvida
- Pode significar que os recursos da linguagem de baixo nível de programação têm que ser usados por motivos de desempenho

Modelagem de máquinas de estado

- O efeito de um estímulo, em um sistema em tempo real, pode desencadear uma transição de um estado para outro.
- Máquinas de estado finito podem ser usadas para modelar sistemas de tempo real.
- No entanto, falta modelos de estrutura de máquinas de estado finito. Mesmo sistemas simples pode ter um modelo complexo.
- A UML inclui notações para a definição de modelos de máquina de estado

Máquina de estado de um forno microondas



Programação de tempo real

- Os sistemas de tempo real rígidos ainda são programados em linguagem assembly, de modo que prazos curtos podem ser cumpridos
- Linguagens como o C permite o desenvolvimento de programas muito eficientes, mas não inclui princípio para lidar com a concorrência ou com o gerenciamento de recursos compartilhados
- A linguagem Ada foi projetada para a implementação de sistemas embutidos e possui um bom mecanismo de propósito geral para sincronização de tarefas

Java como uma linguagem de R-T

- Java tem alguma compatibilidade para processos concorrentes (threads e métodos sincronizados) e pode ser usada para alguns sistemas de tempo real leves
- Java 2.0 não é adequada para programar sistemas de tempo real rígidos ou sistemas em que os processos tenham prazos rígidos
 - Não é possível especificar o momento em que threads devem ser executados
 - A coleta de lixo (garbage) não pode ser controlada
 - Não é possível descobrir os tamanhos das filas associadas com recursos compartilhados
 - A implementação da Máquina Virtual Java varia
 - Não permite a análise detalhada de espaço de run-time ou de processador

Executivos de tempo real

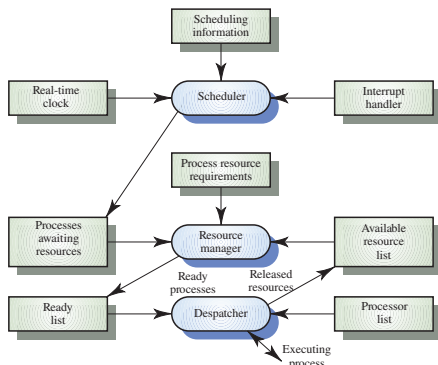
- Um executivo de tempo real gerencia a alocação de processos e recursos em um sistema de tempo real
- Responsável por iniciar e interromper os processos apropriados, de modo que os estímulos possam ser manipulados, e alocar a memória e os recursos do processador
- Não inclui recursos complexos, como gerenciamento de arquivo

Componentes de um executivo

- Um relógio de tempo real
 - Fornece informações para escalonar processos periodicamente.
- Um manipulador de interrupções
 - Gerencia os pedidos aperiódicos de serviços.
- Um escalonador
 - Responsável por examinar os processos que podem ser executados e escolher um deles para execução.
- Um gerenciador de recursos
 - Aloca a memória apropriada e os recursos de processador.
- Um despachador
 - Responsável por iniciar a execução de um processo.

Outras capacidades executivas

- Gerenciador de configurações
 - Responsável pela configuração dinâmica do hardware do sistema, os módulos de hardware podem ser tirados de serviço e o sistema pode ser atualizado pela adição de novo hardware, sem interromper o sistema
- Gerenciador de defeitos
 - Responsável por detectar defeitos de hardware e de software e tomar as medidas apropriadas para a recuperação a partir desses defeitos



Componentes de um executivo de tempo real

Prioridade de processos

- Os estímulos processados por um sistema de R-T têm diferentes níveis de prioridade
- Nível de interrupção
 - Mais alta prioridade é alocado para processos que precisam de uma resposta muito rápida
- Nível de relógio
 - Alocado aos processos periódicos
- Dentro desses, mais níveis de prioridade pode ser atribuídos

Interromper serviço

- O controle é transferido automaticamente para um local pré-determinado da memória
- Este local contém uma instrução de salto para interromper um serviço de rotina
- Outras interrupções são desabilitadas, o controle e serviço de interrupção são devolvidos ao processo interrompido
- Interromper as rotinas de serviço deve ser curto, simples e rápido

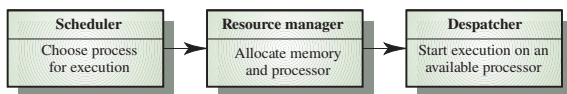
Processos periódicos

- Na maioria dos sistemas de tempo real, haverá várias classes de processos periódicos, cada uma com diferentes períodos (o tempo entre a execução de processos), tempos de execução e prazos de conclusão (o tempo em que o processamento deve ser concluído)
- Os processos periódicos são aqueles que precisam ser executados em intervalos de tempo previamente especificados para a aquisição de dados e o controle de atuador
- O executivo precisa escolher o processo apropriado para a execução

Gerenciamento de processos

- Se ocupa de gerenciar o conjunto de processos concorrentes
- Processos periódicos são executados em intervalos de tempo previamente especificados
- O executivo utiliza seu relógio de tempo real para determinar quando um processo deve ser executado
- Período de processo – tempo entre as execuções
- Prazos de processo – o tempo em que o processamento deve ser concluído

Gerenciamento de processos de executivo de tempo real



Ações de um executivo

- O escalonador escolhe o próximo processo a ser executado. A escolha depende da prioridade do processo, dos períodos de processo, dos tempos de execução esperados e dos prazos de conclusão dos processos prontos
- O gerenciador de recursos aloca a memória e um processador para o processo a ser executado
- O Despachador inicia o processo a partir de uma lista pronta, carrega-o para um processador e começa a execução

Estratégias de escalonamento

- Escalonamento não preemptivo (non pre-emptive)
 - Uma vez escalonado o processo para execução, ele é executado até ser concluído ou até ser interrompido, por alguma razão (p. ex. aguardando por uma E/S)
- Escalonamento preemptivo (pre-emptive)
 - A execução de um processo pode ser interrompida se um processo de prioridade mais alta solicitar o serviço
- Algoritmos de escalonamento
 - Round-robin (em que cada processo é executado por vez)
 - Monotónico (em que a prioridade é dada ao processo de período)
 - prazo de término mais curto

Sistemas de monitoramento e controle

- Uma classe importante de sistemas de tempo real
- Verificam os sensores e realizam ações dependendo da leitura do sensor
- Os sistemas de monitoramento realizam uma ação quando algum valor excepcional é detectado pelo sensor
- Os sistemas de controle controlam os atuadores de hardware, dependendo do valor dos sensores associados

Sistema de alarme

- O sistema utiliza tipos diferentes de sensores, incluindo detectores de movimento, sensores de janelas e sensores de porta para detectar presença de intrusos
- Quando o sensores indicada a presença de intrusos, o sistema liga as luzes e automaticamente chama a polícia
- O sistema deverá prever a possibilidade de uma operação sem energia elétrica

Processo de projeto de sistema R-T

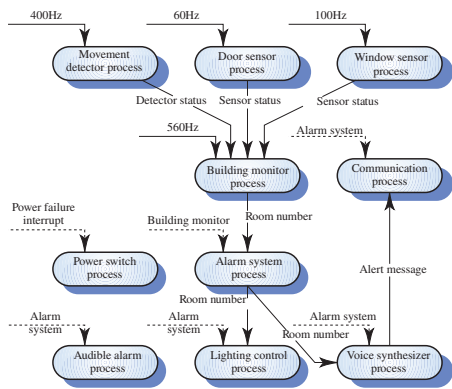
- Identificação de estímulos e das respostas associadas
- Define as restrições de tempo associadas com cada estímulo e resposta
- Atribui funções de sistema para os processos
- Projeta algoritmos para o processamento dos estímulos e das respostas geradas
- Projeta um sistema de controle que garante que os processos serão executados no prazo

Estímulos processados

- Falha de energia
 - Estímulo gerado pelo monitor de circuito. Quando recebido, o sistema deve alterar para o circuito de backup de energia
- Alarme contra intrusos
 - Estímulo gerado por um dos sensores do sistema. A resposta consiste em fazer um chamado para a polícia, ligar o alarme sonoro e as luzes do prédio

Requisitos de tempo

Estímulo/respostas	Requisitos de tempo
Interrupção por falta de energia	A transferência para a fonte de energia reserva deve ser concluída dentro de um prazo de 50 milissegundos
Alarme de porta	Cada alarme de porta deve ser verificado duas vezes por segundo
Alarme de janela	Cada alarme de janela deve ser verificado duas vezes por segundo
Detector de movimento	Cada detector de movimento deve ser verificado duas vezes por segundo
Alarme sonoro	O alarme sonoro deve ser ligado dentro de meio segundo a partir do momento em que um alarme é ativado por um sensor
Interruptores de luz	As luzes devem ser ligadas dentro de meio segundo a partir do momento em que um alarme é ativado por um sensor
Comunicações	O chamado para a polícia deve ser iniciado dentro de dois segundos a partir do momento em que um alarme é ativado por um sensor
Sintetizador de voz	Uma mensagem sintetizada deve estar disponível dentro de quatro segundos segundo a partir do momento em que um alarme é ativado por um sensor



Arquitetura de processo

```
// See http://www.software-engin.com/ for links to the complete Java code for this
// example

class BuildingMonitor extends Thread {

    BuildingSensor win, door, move ;

    Siren    siren = new Siren () ;
    Lights  lights = new Lights () ;
    Synthesizer synthesizer = new Synthesizer () ;
    DoorSensors doors = new DoorSensors (30) ;
    WindowSensors windows = new WindowSensors (50) ;
    MovementSensors movements = new MovementSensors (200) ;
    PowerMonitor pm = new PowerMonitor () ;

    BuildingMonitor()
    {
        // initialise all the sensors and start the processes
        siren.start () ; lights.start () ;
        synthesizer.start () ; windows.start () ;
        doors.start () ; movements.start () ; pm.start () ;
    }

    Building_monitor process 1
}
```

```
public void run ()
{
    int room = 0 ;
    while (true)
    {
        // poll the movement sensors at least twice per second (400 Hz)
        move = movements.getVal () ;
        // poll the window sensors at least twice/second (100 Hz)
        win = windows.getVal () ;
        // poll the door sensors at least twice per second (60 Hz)
        door = doors.getVal () ;
        if (move.sensorVal == 1 | door.sensorVal == 1 | win.sensorVal == 1)
        {
            // a sensor has indicated an intruder
            if (move.sensorVal == 1)    room = move.room ;
            if (door.sensorVal == 1)    room = door.room ;
            if (win.sensorVal == 1)     room = win.room ;

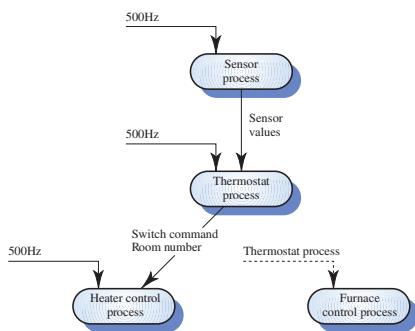
            lights.on (room) ; siren.on () ; synthesizer.on (room) ;
            break ;
        }
    }
    lights.shutdown () ; siren.shutdown () ; synthesizer.shutdown () ;
    windows.shutdown () ; doors.shutdown () ; movements.shutdown () ;
} // run
} //BuildingMonitor

Building_monitor process 2
}
```

Sistema de controle

- Um sistema de alarme é essencialmente um sistema de vigilância. Ele coleta os dados de sensores, mas não em tempo real de controle de atuador
- Os sistemas de controle são semelhantes, mas, em resposta aos valores dos sensores, o sistema envia sinais para controlar os atuadores
- Um exemplo de um sistema de monitoramento e controle é um sistema que controla a temperatura e interruptores ligados e desligados

Sistema de controle de temperatura



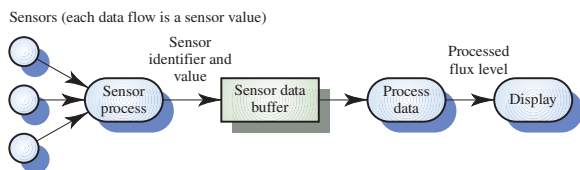
Sistema de aquisição de dados

- Coletam dados de sensores para subsequente processamento e análise.
- Processo de coleta de dados e processamento dos processos podem ter diferentes períodos e prazos.
- A coleta de dados pode ser mais rápida do que o processamento.
- Buffers em circular ou anel são um mecanismo de nivelamento de diferenças de velocidade

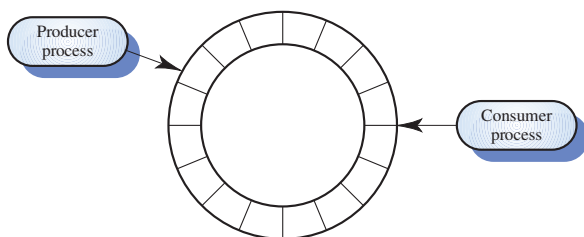
Coleta de dados de um reator

- Um sistema que coleta dados de sensores monitorando o fluxo de nêutrons em um reator nuclear.
- Os dados do sensor são colocados em um buffer em anel para serem processados.
- Um buffer em anel é implementado como um processo concorrente de modo que os processos de coleta e processamento possam ser sincronizados

Monitoramento de fluxo



Buffer em anel



Exclusão mútua

- O processo que produz os dados (produtor) adiciona as informações ao buffer. O processo que utiliza os dados (consumidor) obtém as informações do buffer
- O processo produtor e consumidor podem acessar simultaneamente para excluir a mesmo elemento.
- O sistema deve assegurar que o produtor não tente adicionar informações a um buffer cheio e que o consumidor não obtenha informações de um buffer vazio.

```
class CircularBuffer
{
    int bufsize ;
    SensorRecord [] store ;
    int numberOfEntries = 0 ;
    int front = 0, back = 0 ;

    CircularBuffer (int n) {
        bufsize = n ;
        store = new SensorRecord [bufsize] ;
    } // CircularBuffer

    synchronized void put (SensorRecord rec) throws InterruptedException
    {
        if ( numberOfEntries == bufsize)
            wait () ;
        store [back] = new SensorRecord (rec.sensorId, rec.sensorVal) ;
        back = back + 1 ;
        if (back == bufsize)
            back = 0 ;
        numberOfEntries = numberOfEntries + 1 ;
        notify () ;
    } // put
}
```

Java implementation of a ring buffer 1

```
synchronized SensorRecord get () throws InterruptedException
{
    SensorRecord result = new SensorRecord (-1, -1) ;
    if (numberOfEntries == 0)
        wait () ;
    result = store [front] ;
    front = front + 1 ;
    if (front == bufsize)
        front = 0 ;
    numberOfEntries = numberOfEntries - 1 ;
    notify () ;
    return result ;
} // get
} // CircularBuffer
```

Java implementation of a ring buffer 2

Pontos-chave

- Um sistema em tempo real é um sistema de software que precisa responder a eventos em tempo real, sua precisão não depende apenas dos resultados que ele produz, mas também do momento em que esses resultados são produzidos
- Um modelo geral para sistema de tempo real envolve associar um processo com sensores e atuadores
- O projeto de arquitetura de um sistema de tempo real geralmente envolve organizar o sistema como um conjunto de processos concorrentes que interagem entre si

Pontos-chave

- Um executivo em tempo real é responsável pelo gerenciamento de processo e recursos.
- Os sistemas de monitoramento e controle varrem periodicamente um conjunto de sensores emitindo comandos para os atuadores
- Os sistemas de aquisição de dados são geralmente organizados de acordo com um modelo produtor-consumidor
- Embora Java tenha recursos de apoio à concorrência, esta não é uma linguagem adequada para o desenvolvimento de sistemas de tempo real

Bibliografia

- SOMMERVILLE, Ian, Engenharia de Software. 6 ed. São Paulo: Addison Wesley, 2003.
- <http://www.comp.lancs.ac.uk/computing/resources/IanS/SE6/Slides/PPT/ch13.ppt>
