Modelagem de Sistemas de Tempo Real

Sérgio Campos

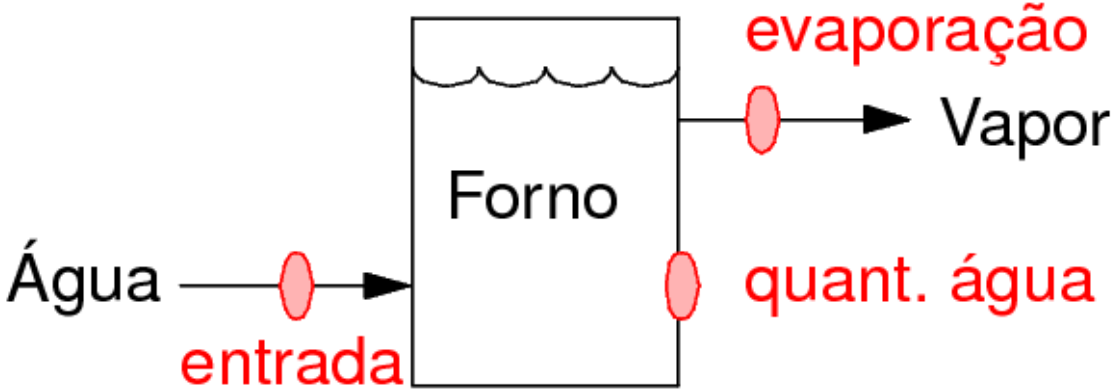
Antecedentes

- Sistemas reativos,
- Processos: comunicação, sincronização,
- Escalonamento,
- Sistemas distribuidos.
- Jobs, processors & timing constraints
- Hard & soft real-time
- Deadlines
- Escalonamento On-Line X Off-Line
- Modelos periódico e aperiódico

Sistemas Reativos

São programas quem não produzem resultados nem terminam!!!

- Sua tarefa é interagir continuamente com o ambiente em que se encontram.
- Exemplos: sistemas operacionais, controladores de máquinas industriais ou de usinas de energia.
- Elemento fundamental : Concorrência.
- Sistemas reativos por definição executam em paralelo com o ambiente.
  - O ambiente contém vários elementos que funcionam em paralelo.



Caldeira

Caldeira ? — Chernobyl

Sistemas Reativos

Concorrência acrescenta uma grande complexidade ao sistema:

- Programas concorrentes são os mais difíceis de se escrever

Devido a ser utilizados em situações críticas, o projeto de sistemas reativos deve seguir regras especiais para se garantir sua corretude.

Sistemas reativos são normalmente *event driven*.

```
while (!fim_do_mundo) {
  switch (event) {
    case a: handle(a); break;
    case b: handle(b); break;
    ...
  };
}
```

Exemplo: Produtor/Consumidor

```
producer()
{
  while (true) {
    if (new_item(&item))
      buffer[++p] = item;
  };
}

consumer()
{
  while (true) {
    if (p != c) consume(buffer[c++]);
  };
}
```

Jobs, Processors, Timing Constraints

- Unidade de trabalho: job;
- Conjunto de jobs: task.

- Exemplos de jobs: transmissão de mensagem, cálculo de função de controle, leitura de dados do sensor.

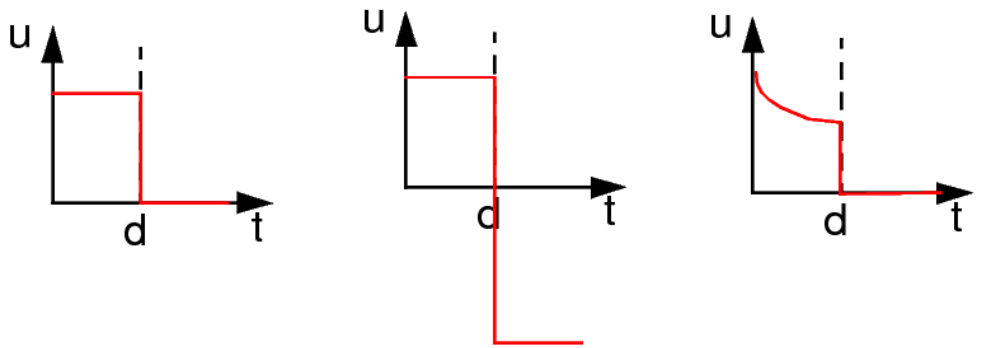
- Jobs utilizam recursos. Exemplos de recursos usados: rede, CPU, interface com sensor.
  - Nome genérico usado: processador

- Timing constraints:
  - Release time: momento em que o job está pronto para executar.
  - Deadline: momento no qual a execução tem que terminar
  - Período: tempo entre execuções consecutivas de um job.

Hard & Soft Real-Time

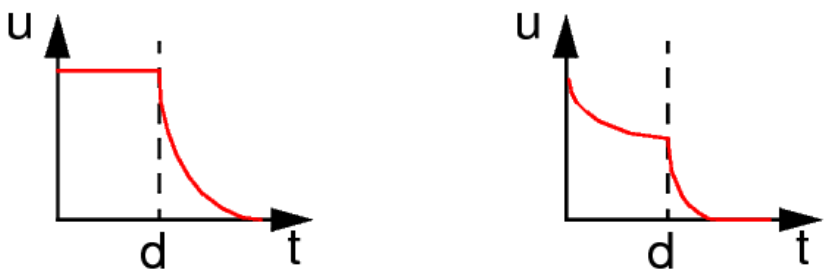
Hard real-time: guaranteed

- Controle aviões/carros e processos industriais.



Soft real-time: best effort

- Exemplos: multimedia



"Dureza" pode ser como: Probabilidade de falha menor que 10^{-7}

Deadlines

Porque deadlines ?

- Tempo de resposta do sistema:
  - Tela do piloto tem que ser atualizada 30 vezes por segundo
  - Tempo entre detectar o buraco e endurecer amortecedores depende da velocidade do carro.
- Controle:
  - Se a função de controle não for computada na frequência certa o sistema pode oscilar.

E se algumas forem perdidas ?

- Hard real-time: 
- Soft real-time: depende...
  - Vídeo: algumas não atrapalha.
  - Áudio: muito mais grave, o ouvido nota.
  - Monitor médico: ok, desde que avisado.
  - Centrais telefônicas: ok, desde que seja raro
    - GOS: % de chamadas perdidas

Escalonamento On-Line X Off-Line

Em algumas aplicações todos os tasks são conhecidos a priori.

A validação destes sistemas é feita durante o projeto:

- O produto final é correto por construção.
- Exemplo: sistemas embutidos

Em outras aplicações não se sabe o conjunto de tasks a executar:

- Cada nova tarefa a cumprir deve ser checada;
- O algoritmo que decide se aceita ou não a nova tarefa é chamado de controle de admissão.
- Exemplo: servidor de vídeo.

Modelando Sistemas de Tempo-Real

Para que possamos analisar sistemas de tempo-real precisamos de abstrair de detalhes:

No nosso modelo um sistema de tempo-real é composto por:

- *Processadores* P_i: fazem o trabalho pesado. Podem ser de diversos tipos
  - processadores; rede; disco.
- *Recursos*: necessários para a execução
  - memória; mutexes; mensagens.

  Recursos são:
  - reutilizáveis; escassos; mutex

Exemplo: 1 CPU, 2 processos, 1 mutex:

- Modelo: P_1, j_1, j_2, R_1 (CPU), R_2 (mutex).

Notem que a modelagem enfatiza os aspectos de concorrência: paralelismo e sincronização

Requisitos Temporais

- *Release time*: hora em que o job pode começar
  - Fixo: r_i
  - Jitter: $[r_i^- .. r_i^+]$ — causado por imprecisão nas medidas
  - Esporádico: variável aleatória — distribuição da probabilidade de que o evento aconteça.
- *Deadline*: absoluta ou relativa ao release time.
- *Tempo de execução* : como saber ?
  - Usa-se um intervalo $[e_i^- .. e_i^+]$.
  - A fim de se determinar escalonabilidade normalmente assume-se o pior caso, e_i^+.
    - Pessimismo leva a perda de eficiência;
    - Para aplicações críticas isto é aceitável;
    - Análise é simplificada e torna-se possível;
    - A maior parte do sistema não é crítica.

"E tem aquela história do cara que morreu afogado em um riacho com profundidade média de 30cm..."

Modelo Periódico

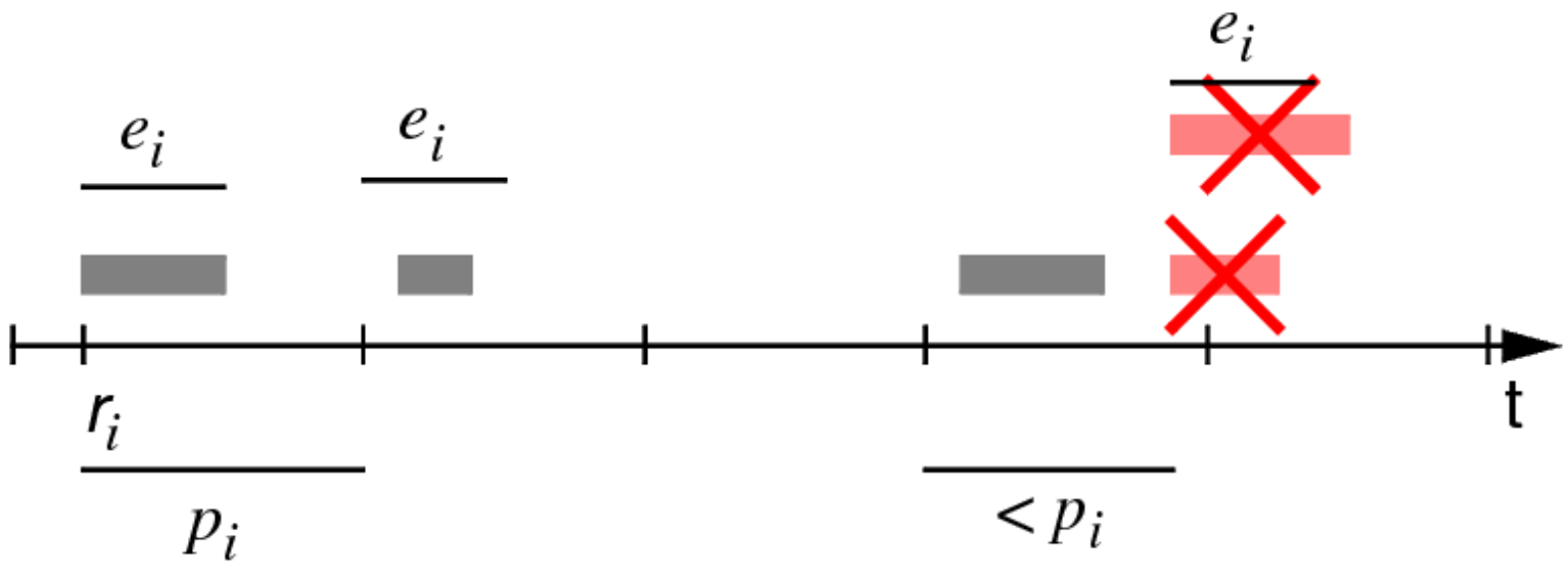
Aplicação natural em diversos casos:

- Controle;
- Monitoração;
- Multimedia

Modelo Periódico

Cada tarefa (task) T_i consiste de uma sequência de jobs que são executados pelo menos a p_i unidades de tempo. Cada execução gasta tempo e_i (o máximo entre todos jobs).

- $U_i = e_i/p_i$;
- $D_i = R_i$ (por enquanto);
- Hiperperíodo = MMC$(p_1, p_2, ..., p_n)$.



Modelo Aperiódico

Tipicamente respostas a eventos independentes

- Exemplo, o motorista pode tirar o pé do acelerador a qualquer instante.

Uma tarefa aperiódica consiste de uma sequência de jobs aperiódicos:

- Exemplo, todos os jobs que tratam de alarmes pertencem à mesma tarefa aperiódica

Caracterização:

- Distribuição de probabilidades de ocorrência de eventos;
- Todos os jobs têm as mesmas características estatísticas.

Nomenclatura:

- Aperíodicos: soft aperíodicos;
- Esporádicos: hard aperíodicos.

Dependências entre Processos

Pode ser de várias formas:

- Explícita:
  - Processo P_1 chama o Processo P_2
  - Gera um grafo de precedências
- Através de dados compartilhados:
  - `P1.lock() || P2.lock()`
- Temporal:
  - `Apply.brakes(); Engine.reverse();`

Parâmetros do Sistema

- Preemptividade:
  - "Preemptáveis": e.g. uso da CPU
  - Não "preemptáveis": envio de mensagens — preempção ineficiente
- Importância:
  - Navegacão do avião X controle do filme mostrado
- Execução opcional:
  - Condicional: só em alguns ifs
  - Falta de tempo
  - Situações especiais: emergência
  - Normalmente uma parte e' obrigatoria e outra opcional