



Processos Concorrentes

Walter Fetter Lages

w.fetter@ieee.org

Universidade Federal do Rio Grande do Sul

Escola de Engenharia

Departamento de Engenharia Elétrica

ENG04008 Sistemas de Tempo Real



Introdução

- Sistemas de tempo real são inerentemente concorrentes
- Programas concorrentes podem ser vistos como um conjunto de programa sequenciais executados em paralelo
- Linguagens de programação concorrente incorporam a noção de processo
- Um processo possui apenas um fluxo de controle

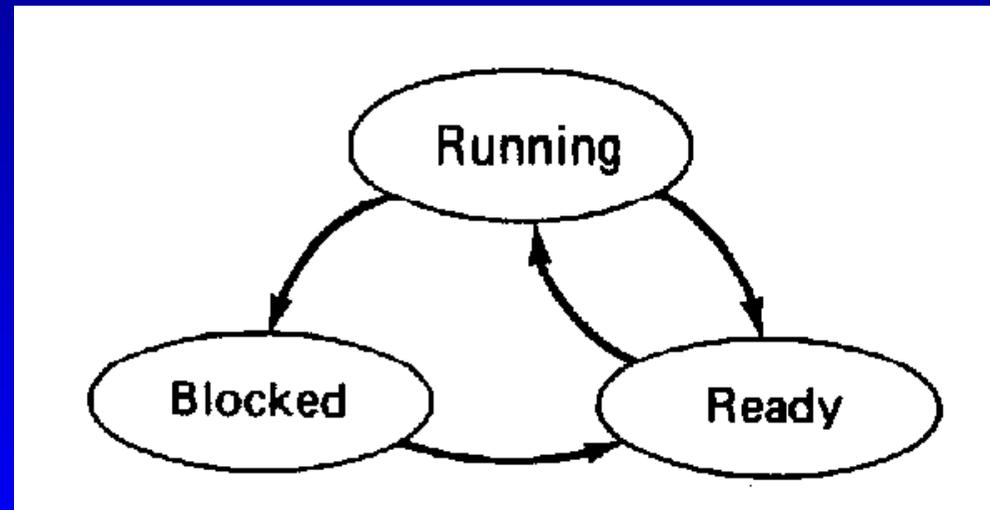


Execução de Processos

- Multiplexados em uma única CPU
 - Pseudoparalelismo
- Multiplexados em diversas CPUs com memória compartilhada
 - Paralelismo espacial
- Multiplexados em diversas CPUs sem memória compartilhada
 - Sistemas distribuídos
- Execução de instruções em pipeline
 - Paralelismo temporal

Estados de um Processo

- Pronto
 - Esperando que a CPU fique livre
 - Estado inicial
- Executando
- Bloqueado
 - Esperando algum evento externo



Implementação de Processos



- Contexto do processo
 - Registradores da CPU
 - Contador de programa
 - Ponteiro de pilha
 - Alocação de memória
 - Arquivos abertos
 - Etc...
- Cada processo tem seus segmentos
 - Código
 - Dados
 - Pilha



Chaveamento de Processos

- Realizado pelo sistema operacional
 - O contexto do processo é salvo na tabela de processos
 - Suporte de hardware em alguns casos
 - É selecionado um novo processo entre os que estão no estado de pronto (scheduling)
 - O contexto do novo processo é obtido da tabela de processos e carregado na CPU
- Do ponto de vista do processo é como se ele nunca tivesse deixado de executar



Suporte à Concorrência

- Linguagens concorrentes
 - Mais legível e mais fácil de manter
 - Torna o programa mais portátil
 - Dispensa o uso de sistema operacional
 - O compilador pode fazer verificações
- Sistemas operacionais
 - Possibilita o uso de diversas linguagens em um único sistema
 - Implementar a noção de concorrência de determinadas linguagens pode ser difícil em certos sistemas operacionais



Programação Concorrente

- Facilidades necessárias
 - Expressão da concorrência através da noção de processos
 - Sincronização de processos
 - Comunicação entre processos
- Tipos de interação entre processos
 - Independentes
 - Cooperantes
 - Competidores



Modelos de Concorrência

- Estrutura
 - Estática
 - Número de processos fixo
 - Dinâmica
 - Número de processos variável
- Nível
 - Aninhado
 - Pode ser definida uma hierarquia de processos
 - Plano



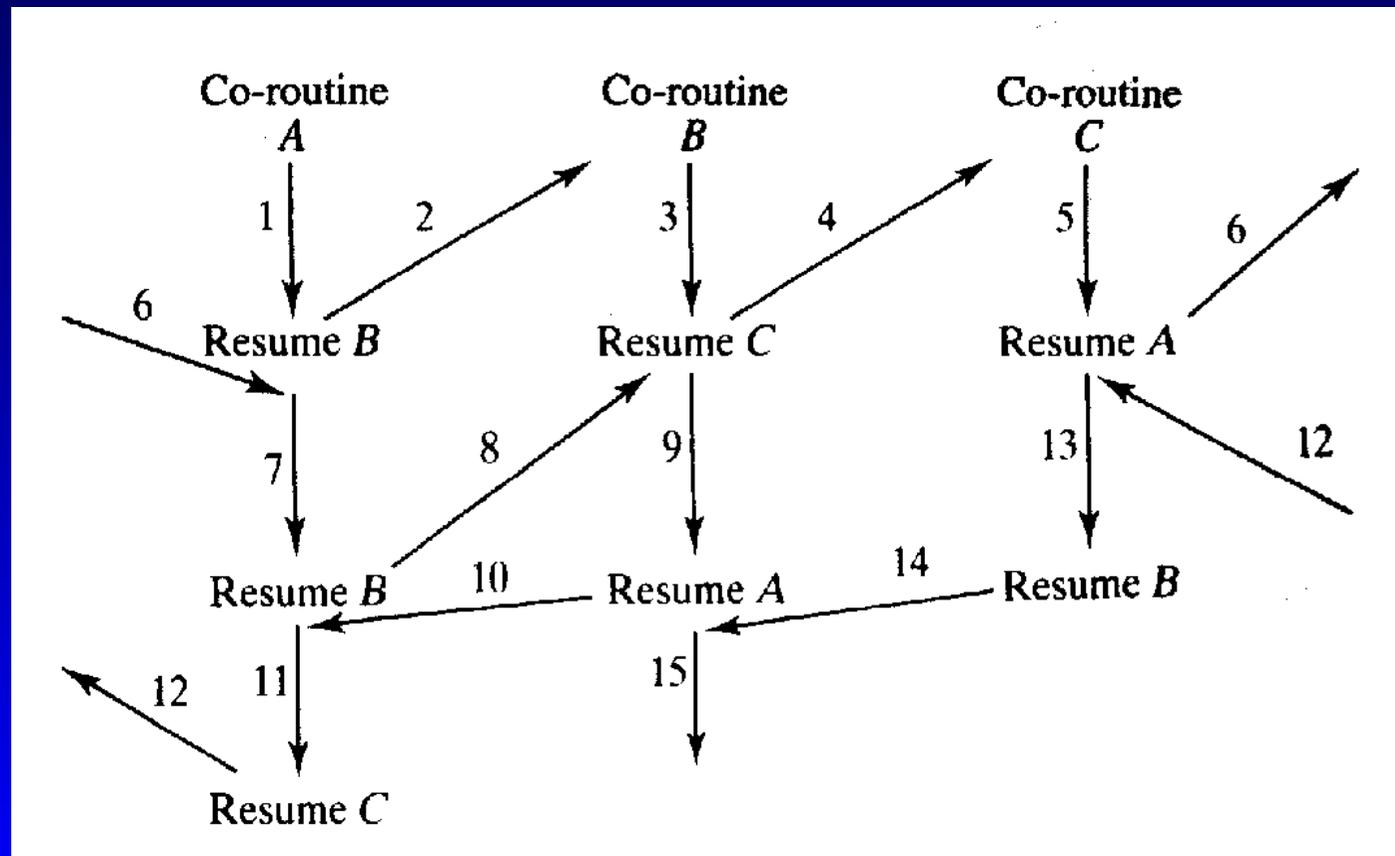
Modelos de Concorrência

- Granularidade
 - Grossa ou Fina
- Inicialização
 - Parâmetros passados na inicialização
 - Parâmetros comunicados após a inicialização
- Terminação
 - Ao completar o corpo do processo
 - Suicídio
 - Aborto
 - Erro
 - Nunca
 - Quando não for mais necessário

Representação de Concorrência



- Co-rotinas
 - Transferência explícita do controle através de resume

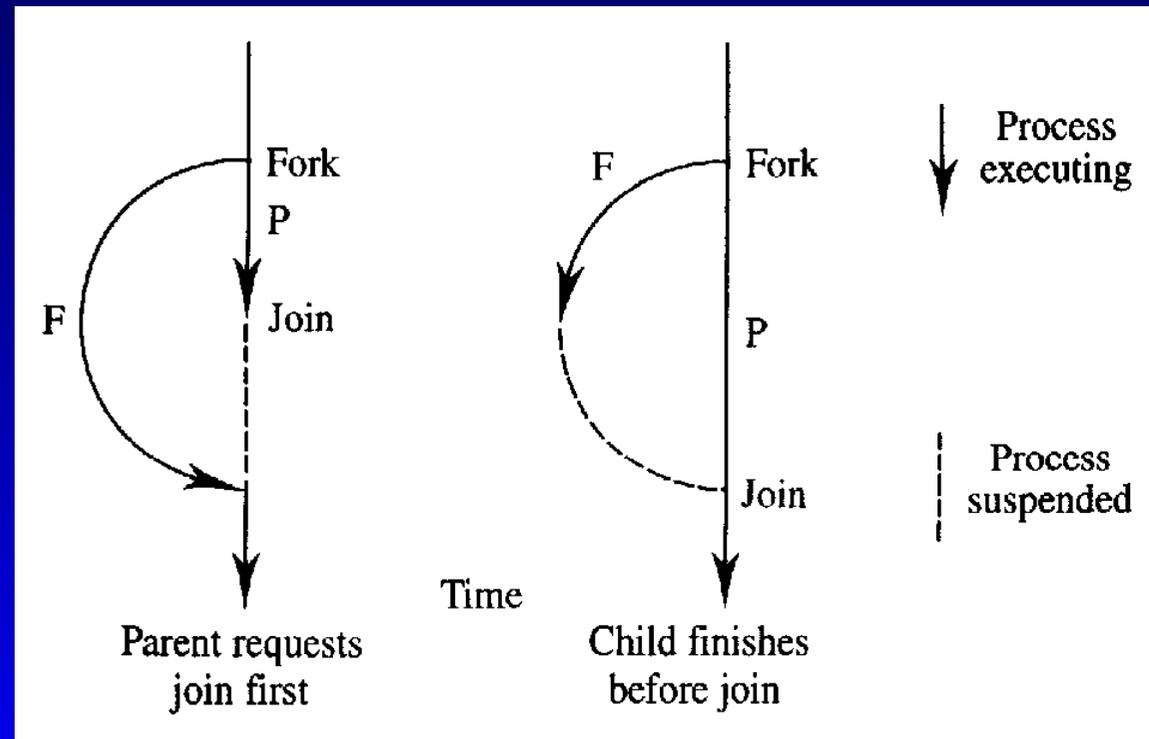


Representação de Concorrência



- Fork/join
 - Fork dispara um processo em paralelo (filho)
 - Join força o sincronismo entre os processos

```
function F return ... ;  
.  
.  
end F;  
  
procedure P;  
  ...  
  C := fork F;  
  .  
  .  
  J := join C;  
  ...  
end P;
```

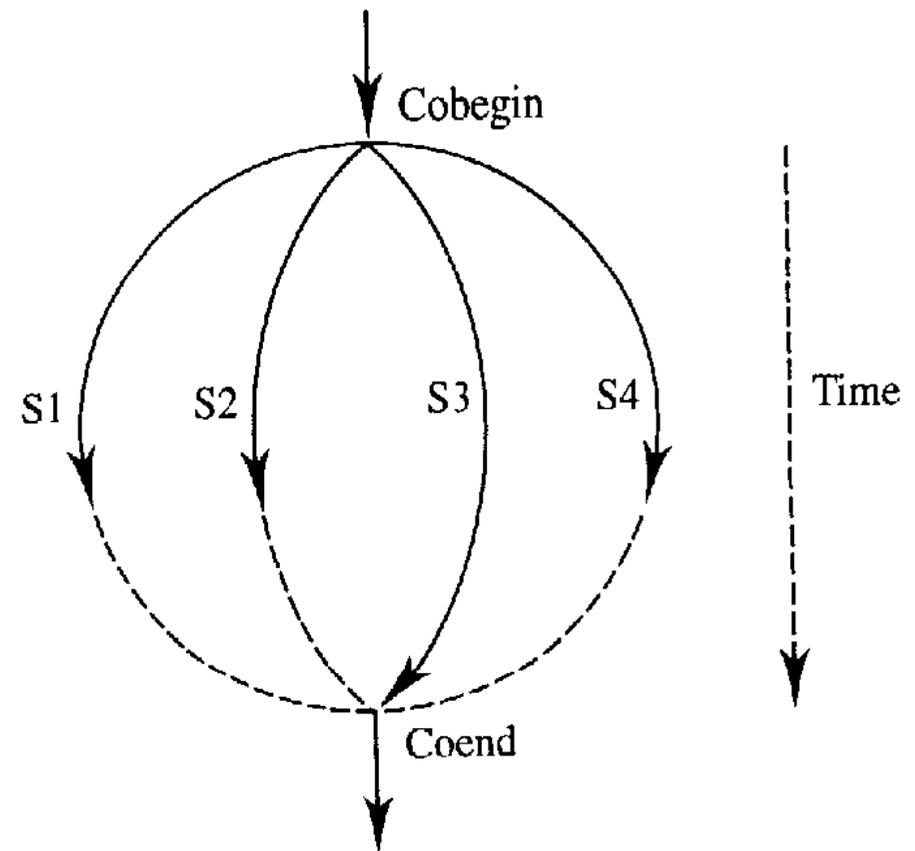


Representação de Concorrência



- cobegin/coend
 - Forma estruturada de explicitar o paralelismo

```
cobegin  
  S1;  
  S2;  
  S3;  
  .  
  .  
  .  
  Sn  
coend
```



Representação de Concorrência



- Declaração explícita de processos

```
MODULE main;
  TYPE dimension = (xplane, yplane, zplane);

  PROCESS control (dim : dimension);
    VAR position : integer; (* absolute position *)
        setting : integer; (* relative movement *)
  BEGIN
    position := 0; (* rest position *)
    LOOP
      new_setting (dim, setting);
      position := position + setting;
      move_arm (dim, position)
    END
  END control;

BEGIN
  control(xplane);
  control(yplane);
  control(zplane)
END main.
```



Threads

- Semelhantes a processos, no entanto compartilham o mesmo segmento de dados
- No POSIX threads podem existir dentro de processos