Deadlocks

Calebe de Paula Bianchini

Definição

- “Quando dois trens se aproximarem um do outro em um cruzamento, ambos deverão parar completamente e nenhum dos dois deverá ser acionado até que o outro tenha partido” [Kansas/USA, sec. XX]
- Em SO: um processo solicita um recurso que está ocupado, fica em espera, e esse recurso nunca é liberado

Introdução

- Um recurso requisitado por um processo deve seguir a seqüência:
  - Pedido: solicita o recurso, ou espera esse ser liberado
  - Uso: utiliza o recurso
  - Liberação: o processo libera o recurso
- Mas, com isso também pode acontecer deadlock?

Introdução

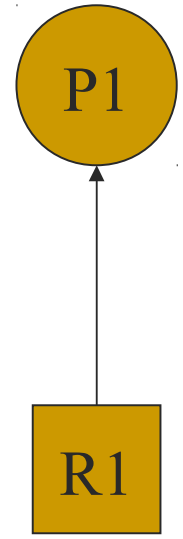
Processo 1	Processo 2

while(1) {
wait(A);
wait(B);
trabalha();
signal(B);
signal(A);
}
	while(1) {
	wait(B);
	wait(A);
	trabalha();
	signal(A);
	signal(B);
	}

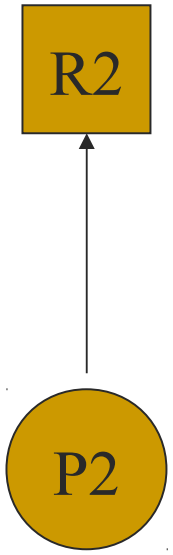
Condições

- Condições para acontecer um Deadlock
  - Exclusão mútua
    - Recurso em modo não-compartilhado
  - Posse e espera
    - Posse de um recurso e espera para obter outro recurso
  - Não-preempção
    - Só o processo que mantém o recurso pode liberá-lo
  - Espera circular
    - Os processos precisam do recursos entre si

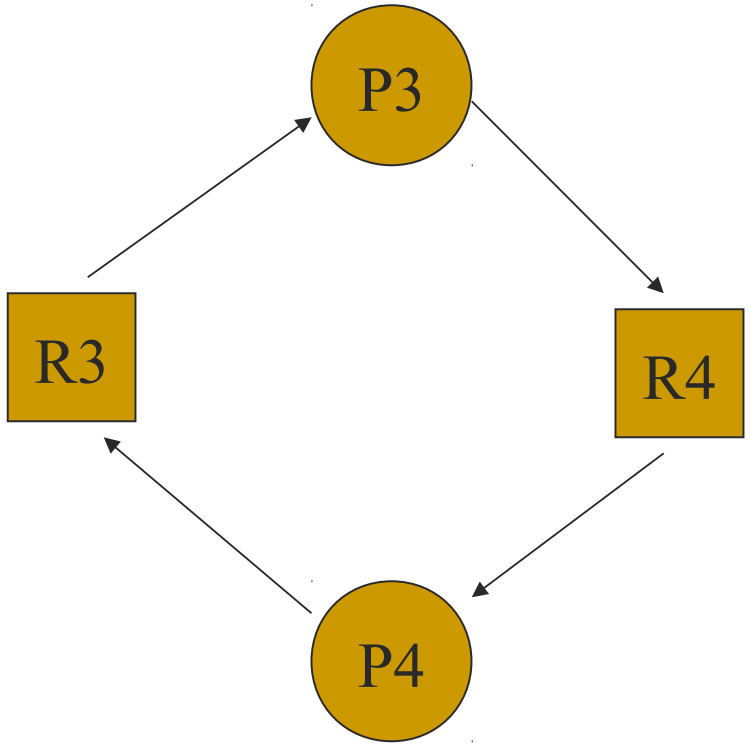
Notação



Posse



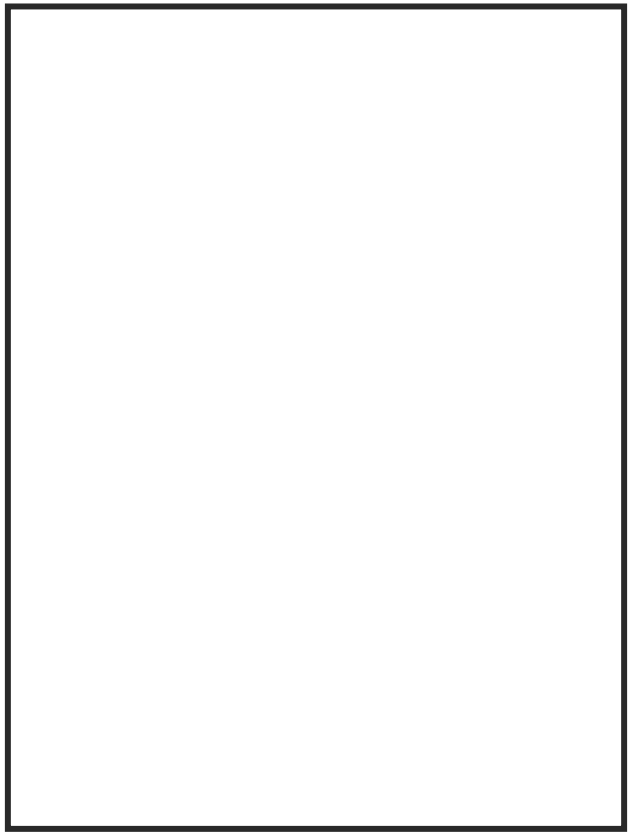
Requisição



Deadlock

Grafo

- Indício em um Grafo Alocação de Recursos
  - Existência de ciclos



Estratégias

- Estratégias usadas para tratamento de Deadlocks
  - Ignorar o problema
  - Detectar e recuperar
  - Evitar
  - Prevenir

Ignorar o problema

- Algoritmo do Avestruz
  - “Coloca a cabeça dentro da terra e finge que nada está acontecendo”
  - Qual a freqüência aceitável? (engenheiros)
    - Quem sabe de 5 em 5 anos por motivos qualquer
  - Mas, e cálculos? (matemáticos)
    - Inaceitável (demora nas respostas!)
  - Windows/Linux/Unix

Detecção & Recuperação

- Grafos de alocação de recurso
- Existência de ciclo
- Caso positivo, recupera-se:
  - Suspensão de um dos processos
  - *Checkpoint* e *RollBack* para um estado anterior seguro
  - Seleção e término de um dos processos

Evitar

- Precisa-se ter algumas informações sobre os processos para evitar deadlocks
- É necessário um seqüência no escalonamento para a execução dos processos
- Algoritmo do Banqueiro – impraticável!

Prevenir

- Uma das 4 condições de deadlocks devem ser negadas!
  - Exclusão mútua: nunca há exclusividade
  - Posse & espera: o processo aloca tudo de uma vez (ou não aloca)
  - Não-preempção: liberar os recursos já obtidos
  - Espera circular: ordenação prioritária dos recursos