

Sistemas Operacionais

Prof. Rafael Obelheiro
rafael.obelheiro@udesc.br



Deadlocks

Sumário

- 1 Noção de deadlock
- 2 Modelagem de deadlock
- 3 Tratamento de deadlocks

Conceito informal de deadlock

- Em um sistema multiprogramado, os processos competem por recursos do sistema
 - ▶ CPU, memória, dispositivos de E/S, tabelas do SO, ...
- Em determinadas situações, os recursos que são alocados exclusivamente a um processo não podem ser retirados à força desse processo
 - ▶ gravador de CD, impressora, ...
- Se P1 detém o recurso X e quer Y, e P2 detém Y e quer X, temos um **deadlock** ou **impasse**
 - ▶ P1 e P2 bloqueiam e nenhum dos dois pode progredir

Recursos

- Deadlocks ocorrem quando se garante aos processos **acesso exclusivo** a recursos
- **Recursos preemptíveis**: podem ser retirados de um processo sem problemas
 - ▶ CPU, memória
- **Recursos não preemptíveis**: podem provocar a falha do processo se forem retirados
 - ▶ impressora, gravador de CD, scanner, mutexes
- Deadlocks só ocorrem com recursos não preemptíveis

Utilização de recursos

- Para utilizar um recurso, o processo tipicamente
 1. solicita o recurso
 2. usa o recurso
 3. libera o recurso
- Quando uma solicitação falha, o processo espera até que o recurso esteja disponível
 - solicitação bloqueia
 - solicitação retorna erro, e o processo fica em loop
- Se um processo não libera os seus recursos após usá-los, a probabilidade de ocorrer um deadlock aumenta

Condições para ocorrência de deadlock

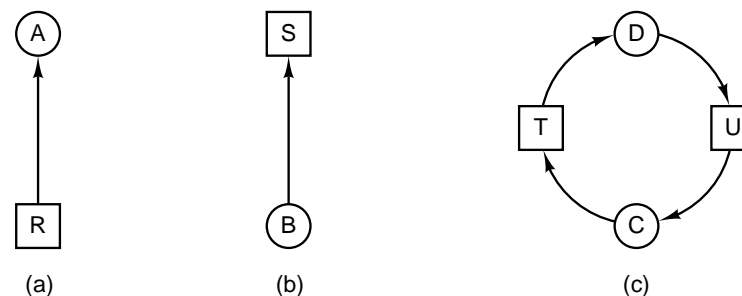
1. Exclusão mútua
 - ▶ todo recurso está ou associado a um processo ou disponível
2. Posse e espera
 - ▶ processos que retêm recursos podem solicitar novos recursos
3. Não preempção
 - ▶ recursos concedidos previamente não podem ser tomados à força
4. Espera circular
 - ▶ deve haver uma cadeia circular de dois ou mais processos
 - ▶ cada um está à espera de recurso retido pelo membro seguinte dessa cadeia

Definição formal de deadlock

- *“Um conjunto de processos está em situação de deadlock se todo processo pertencente ao conjunto estiver esperando por um evento que somente um outro processo desse mesmo conjunto poderá provocar”*
- Normalmente o evento é a liberação de um recurso atualmente retido
- Nenhum dos processos pode...
 - ▶ executar
 - ▶ liberar recursos
 - ▶ ser acordado

Modelagem de deadlock

- Grafo dirigido de alocação de recursos

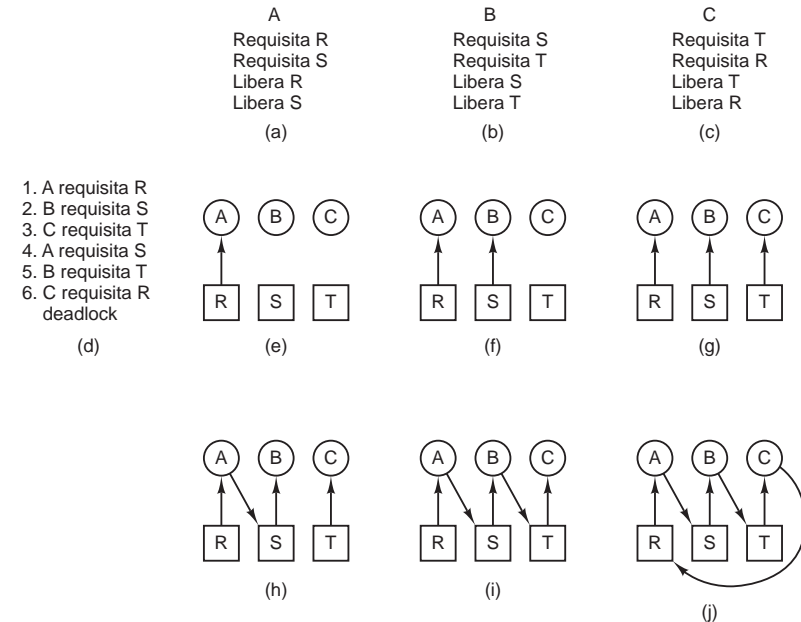


- (a) A alocou R
- (b) B solicitou S (e está bloqueado, esperando a alocação)
- (c) C e D em deadlock sobre T e U

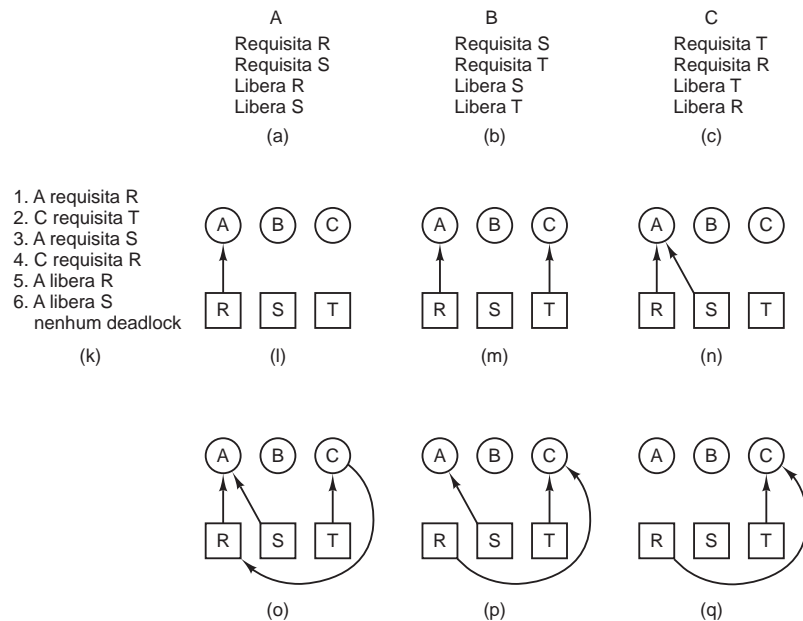
Estratégias para tratar deadlocks

1. Ignorar por completo o problema
2. Detecção e recuperação
3. Evitar dinamicamente a ocorrência
 - alocação cuidadosa de recursos
4. Prevenção
 - negação de uma das quatro condições necessárias

Como ocorre um deadlock



Evitando um deadlock



Algoritmo do avestruz

- “Enterre a cabeça na areia e finja que o problema não existe”



- Ignora a existência de deadlocks; se algum ocorrer, o usuário que resolve
- Baseia-se no princípio que deadlocks são infrequentes na prática
 - é mais provável que o sistema trave antes por outro motivo
 - evita o custo associado aos mecanismos de tratamento de deadlocks
 - ★ desempenho, conveniência
- Estratégia usada no UNIX e no Windows

Detecção de deadlocks

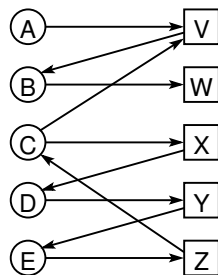
- Um algoritmo simples para detectar a ocorrência de deadlock se baseia no grafo de alocação de recursos
- A ideia é monitorar a alocação de recursos e disparar um procedimento de recuperação caso um ciclo seja encontrado no grafo

Algoritmo para detecção de ciclos

Para todos os nós do grafo, execute os passos a seguir (L é uma lista de nós):

1. $L = []$, todos os arcos são desmarcados
2. Insira o nó atual em L e verifique se ele aparece duas vezes; se aparecer, há um ciclo em L
3. Ache um arco desmarcado saindo do nó corrente
 - 3a. se houver, marque o arco e visite o nó, voltando ao passo 2
 - 3b. se não houver, retire o nó corrente de L e retorne ao nó anterior, voltando ao passo 3

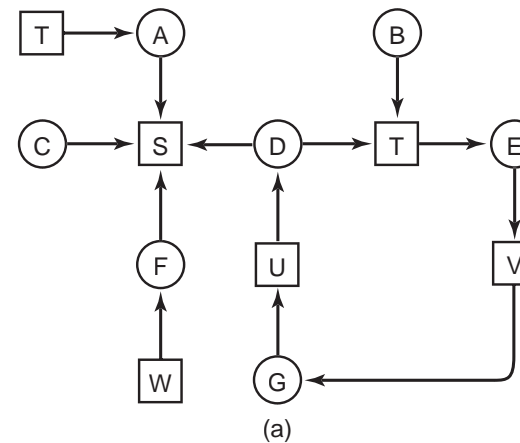
Executando o algoritmo (1/2)



iniciando em E: $L = [E Z C X D Y E]$

processos em deadlock: C, D, E

Executando o algoritmo (2/2)



iniciando em B: $L = [B T E V G U D T]$

processos em deadlock: D, E, G

Métodos de recuperação



- **Preempção**
 - retira um recurso de algum outro processo
 - depende da natureza do recurso
- **Reversão de estado**
 - armazena periodicamente o estado do processo (checkpointing)
 - reinicia um processo do estado salvo (checkpoint) em caso de deadlock
 - ★ tudo o que foi feito desde o checkpoint é perdido e precisa ser refeito
- **Eliminação de processos**
 - escolhe um processo para ser eliminado, quebrando o ciclo
 - o processo escolhido deve deter recursos que estão causando o deadlock
 - preferencialmente se escolhe um processo que possa ser reiniciado sem grandes consequências

Exercícios

Sugerem-se os seguintes exercícios do Tanenbaum, cap. 6:

- 2, 3, 4, 5, 6, 7, 8, 17, 18, 26 (3ª edição)
- 2, 4, 5, 9, 11, 13, 22, 23, 30 (4ª edição)

Bibliografía Básica

-  Andrew S. Tanenbaum e Herbert Bos.
Sistemas Operacionais Modernos, 4ª Edição. Capítulo 6.
Pearson Prentice Hall, 2016.
-  Carlos A. Maziero.
Sistemas Operacionais: Conceitos e Mecanismos. Capítulo 13.
Editora da UFPR, 2019.
<http://wiki.inf.ufpr.br/maziero/doku.php?id=socm:start>