

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

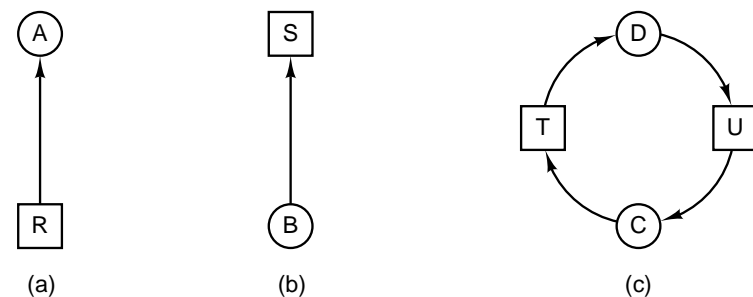
1. Exclusão mútua
 - todo recurso está ou associado a um processo ou disponível
2. Posse e espera
 - processos que retêm recursos podem solicitar novos recursos
3. Não preempção
 - recursos concedidos previamente não podem ser tomados à força
4. Espera circular
 - deve haver uma cadeia circular de dois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

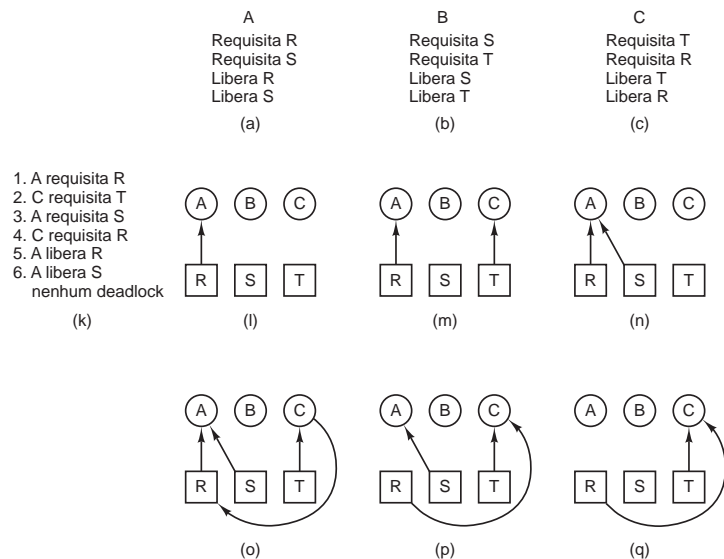


- (a) A alocou R
- (b) B solicitou S (e está bloqueado, esperando a alocação)
- (c) C e D em deadlock sobre T e U

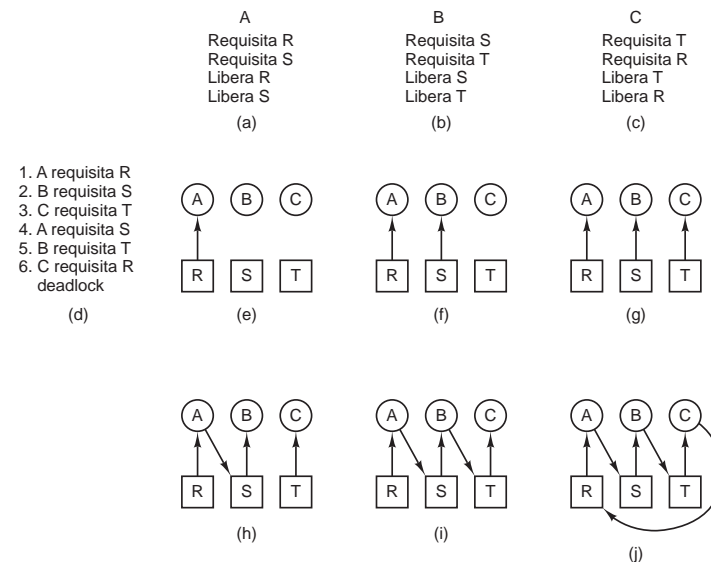
Estratégias para tratar deadlocks

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

Evitando um deadlock



Como ocorre 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

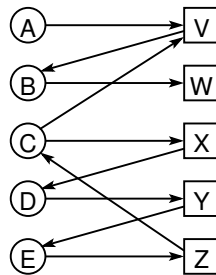
Algoritmo para detecção de ciclos

- 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

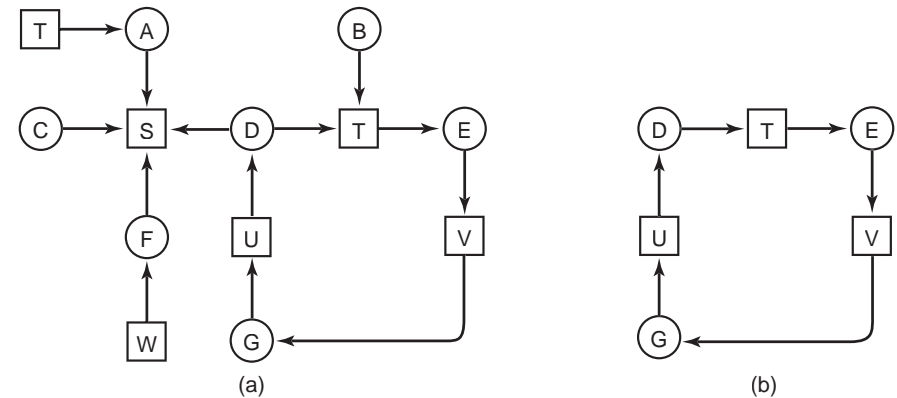
Para todos os nós do grafo, execute os passos a seguir (L é uma lista de nós):

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)



Executando o algoritmo (2/2)



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

