

Revisão P1

Sistemas Operacionais

Erickson Müller

03 de outubro

Conteúdos

1. Processos
2. Threads
3. Impasses
4. Gerenciamento de memória

1 Processos

O sistema operacional tem o modo kernel/supervisor e o modo usuário, o único ator que roda em modo kernel é o SO.

Árvore de processos. O sistema operacional usa a hierarquia de processos para reservar a memória, carregar as partes do processo.

O primeiro processo da árvore inicia o sistema operacional

Os processos podem criar algum processo filho para caso necessite abrir outro programa para executar o comando.

O pipe é um canal de comunicação entre dois processos.

Um processo é a instanciação de um arquivo executável.

1.1 Preemptividade

A multiprogramação permite executar múltiplos processos "ao mesmo tempo", na realidade todos os processos são executados em um ciclo de *50ms* por vez. O *timer* da cpu é responsável por calcular esse ciclo, o timer é uma parte de hardware que tem um clock de 1hz e vai decrescendo de 50 até 0.

Em sistemas operacionais, uma troca de contexto (também conhecido como chaveamento ou mudança de contexto) é o processo computacional de armazenar e restaurar o estado (contexto) de uma CPU de forma que múltiplos processos possam compartilhar uma única instância de CPU.

Cada thread tem sua própria pilha.

1.2 Modelo em quatro partes

O espaço de endereçamento da memória de um processo é dividido em quatro partes:

1. Code/Text
2. Data
3. Heap/Lacuna
4. Stack/Pilha

Pilha e Heap crescem em sentidos opostos.

1.3 Modelo em três partes

O espaço de endereçamento da memória de um processo é dividido em três partes:

1. Code/Text
2. Stack/Pilha
3. Heap + Data

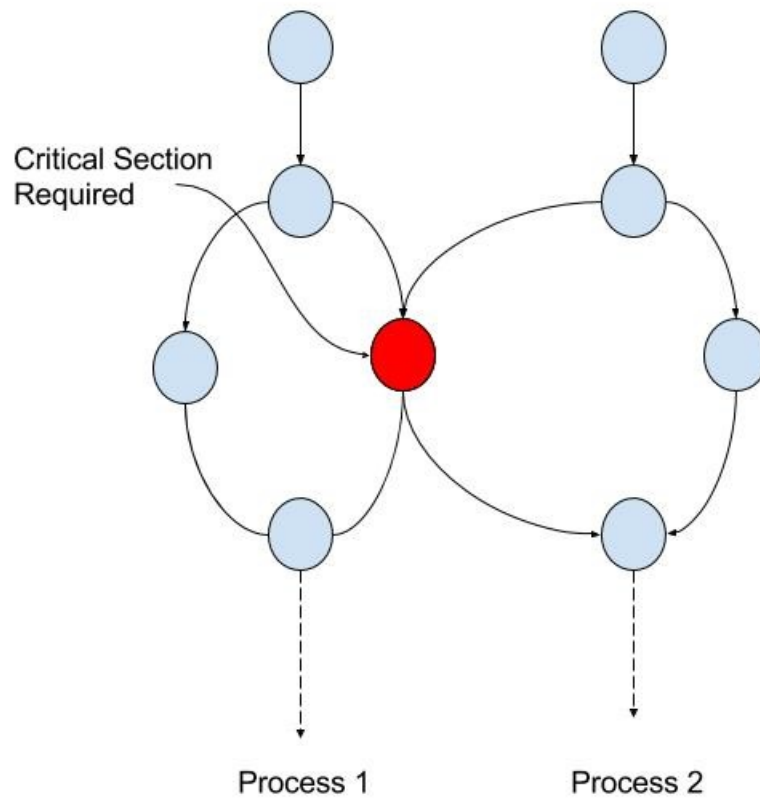
Pilha e Heap crescem em sentidos opostos.

2 Threads

2.1 Região Crítica

Condições para a solução de uma **exclusão mútua**:

- Nunca dois processos simultaneamente em uma região crítica.
- Nenhuma afirmação/exigência sobre velocidades ou números de CPUs.
- Nenhum processo executando fora de sua região crítica pode bloquear outros processos
- Nenhum processo deve esperar eternamente para entrar em sua região crítica (deadlock??)



3 Impasses

recurso, introdução a deadlocks, algoritmo do avestruz, detecção e recuperação de deadlocks, evitando deadlocks, prevenção de deadlocks, outras questões

Em semáforos, o consumidor "acorda" o produtor e vice-versa, para impedir o deadlock.

Os recursos são executados nessa ordem, solicitar -> usar -> liberar.

3.1 Condições para Deadlock

1. Condição de exclusão mútua
2. Condição de posse e espera
3. Condição de não preempção
4. Condição de espera circular

Deadlocks não acontecem em sistemas preemptivos, pois basta salvar o processo e retornar posteriormente.

4 Gerenciamento de Memória

,