

Universidade Federal do ABC
MCTA026-13 - Sistemas Operacionais
2019.Q1

Lista de Exercícios 9

Prof. Emílio Francesquini

24 de abril de 2019

Lista de termos cuja definição você **deve** saber:

- Condição de corrida
- Sincronização de processos
- Seção/Região crítica
- Kernels preemptivos vs. kernels não preemptivos
- Solução de Peterson para sincronização
- Travas/Mutexes, semáforos e monitores
- Problemas clássicos - Jantar dos filósofos

Exercícios

1. Explique a razão do porquê o Windows, o Linux e o Solaris implementam diversos mecanismos de travas. Descreva em quais circunstâncias eles utilizam spin-locks, mutexes e semáforos. Em cada caso não deixe de explicar o porquê do mecanismo ser necessário.
2. O que significa busy-wait ou, em bom português, laço de espera ocupada. Por que este mecanismo é útil?
3. Crie um exemplo onde a exclusão mútua seja violada caso as operações `wait()` e `signal()` de um semáforo não forem executadas atomicamente.
4. Descreva como um semáforo binário pode ser utilizado para criar um mecanismo de exclusão mútua entre um número arbitrário de processos.

5. Condições de corrida são possíveis em muitos sistemas computacionais. Considere um sistema bancário que mantém o saldo de uma conta corrente utilizando duas funções: `deposito(valor)` e `retirada(valor)`. Essas duas funções recebem o valor que deve ser depositado ou retirado da conta. Assuma que um casal compartilhem uma conta corrente. O marido efetua uma retirada ao mesmo tempo que a esposa estiver fazendo um depósito. Mostre como, nestas circunstâncias, uma condição de corrida pode surgir e o que pode ser feito para prevenir que haja qualquer problema no saldo final.
6. Assuma que um sistema computacional tenha diversos cores. Para cada um dos cenários abaixo, descreva qual é o melhor mecanismo de sincronização (um spinlock ou um mutex onde os processos ficam dormindo enquanto o mutex estiver em uso):
 - (a) A trava é mantida travada por períodos de curta duração
 - (b) A trava é mantida travada por longos períodos
 - (c) Um thread pode ser colocado para dormir enquanto detém o uso da trava
7. Assuma que o tempo para fazer uma troca de contexto é T . Sugira um valor em função de T para manter o processo esperando em um spinlock e a partir do qual valor passa a valer a pena utilizar um mutex (onde a thread é colocada para dormir)?
8. Servidores podem ser projetados para limitar o número de conexões abertas. Por exemplo, um servidor pode querer manter N conexões no máximo a qualquer momento. Logo que N conexões estiverem abertas, o servidor passará a não aceitar conexões adicionais até que uma conexão já estabelecida seja finalizada. Explique como podemos utilizar semáforos para limitar o número de conexões a N .
9. O Problema de barbeiro dorminhoco. Uma barbearia consiste em uma sala de espera com n cadeiras, e um salão para o barbeiro com uma cadeira de barbear. Se não houver clientes para atender, o barbeiro vai dormir. Se entrar um cliente na barbearia e todas as cadeiras estiverem ocupadas, o cliente não fica. Se o barbeiro estiver ocupado, mas houver cadeiras disponíveis, o cliente vai esperar em uma das cadeiras livres. Se o barbeiro estiver dormindo, o cliente vai acordá-lo. Escreva um programa para coordenar o barbeiro e os clientes utilizando as ferramentas de sincronização de sua preferência.
10. O problema dos fumantes. Considere um sistema com três processos fumantes e um processo agente. Cada fumante está continuamente enrolando um cigarro e fumando-o em seguida. Mas para enrolar e

fumar um cigarro, o fumante precisa de três ingredientes: fumo, papel e fósforos. Um dos processos fumantes tem papel, outro tem fumo e o terceiro tem os fósforos. O agente tem um suprimento infinito de todos os três materiais. O agente coloca dois dos ingredientes na mesa. O fumante que tem o outro ingrediente faz o cigarro e o fuma, sinalizando o agente na conclusão. O agente coloca outros dois dos ingredientes e o ciclo se repete. Escreva um programa que sincronize o agente e os fumantes usando as ferramentas de sincronização de sua preferência.