

SISTEMAS OPERACIONAIS – 2019.1

Engenharia da Computação

PROF. FERNANDO PARENTE GARCIA

EXERCÍCIO DE SEMÁFOROS

ALUNO: Luis Filipe de Lima Sales

Q.5 /

Problema da busca/inserção/remoção: Três tipos de threads compartilham acesso a uma lista encadeada: busca, inserção e remoção. Threads de busca meramente examinam a lista; Assim podem executar concorrentemente entre si. Threads de inserção adicionam itens no final da lista; inserções devem ser mutuamente exclusivas para impedir duas inserções de ocorrerem simultaneamente. No entanto, uma inserção pode ocorrer em paralelo com qualquer quantidade de buscas. Finalmente, remoções excluem um nó em qualquer local da lista, e pode ocorrer somente uma remoção por vez. Além disso, durante uma remoção não pode ocorrer nenhuma busca e nenhuma inserção. Os algoritmos abaixo propõem uma solução. Descreva detalhadamente os problemas encontrados na implementação proposta, e em seguida proponha alterações para solucionar estes problemas.

SEMAPHORE MUTEX1 = 1; SEMAPHORE MUTEX2 = 1; INT BUSCA = 0; ~~int~~ semaphore mutex3 = 1

THREAD BUSCA: WHILE (TRUE) { DOWN(MUTEX2); IF (BUSCA == 0) DOWN(MUTEX1); BUSCA++; UP(MUTEX2); <i>buscar na lista()</i> ; DOWN(MUTEX2); BUSCA--; IF (BUSCA == 0) UP(MUTEX1); UP(MUTEX2); }	THREAD INSERÇÃO: WHILE (TRUE) { DOWN(MUTEX1 <i>mutex3</i>); <i>inserir na lista()</i> ; UP(MUTEX1 <i>mutex3</i>); }	THREAD REMOÇÃO: WHILE (TRUE) { DOWN(MUTEX1); <i>remover na lista()</i> ; UP(MUTEX1); } <i>if BUSCA == 0</i>
---	---	---

*Insersão e Remoção
podem ocorrer em
PARALELO*

~~Da maneira que está, uma busca pode ocorrer junto com uma remoção, se BUSCA ≠ 0.~~

~~Da maneira que está, a remoção não verifica se não há nenhuma busca ocorrendo, e a busca *insere* a inserção.~~

Busca:

~~while (true):~~

~~down(mutex2)~~