

Sistemas Operacionais - Trabalho Prático I

Exclusão mútua

O primeiro trabalho prático da disciplina de Sistemas Operacionais consiste na implementação de primitivas para garantir a exclusão mútua (*mutex*) inteiramente por *software* (ou seja, utilizando espera ocupada). As primitivas a serem implementadas são *enter_region()* (ou *lock()*) e *leave_region()* (ou *unlock()*).

As primitivas implementadas não devem utilizar semáforos nem mecanismos de exclusão mútua já presentes na biblioteca *pthread*s. Escolha algum algoritmo que implementa exclusão mútua por *software* e que seja generalizado para N processos / *threads* - por exemplo o algoritmo de Dekker, o algoritmo da padaria (*Lamport's Bakery Algorithm*) ou o algoritmo de Peterson.

Para demonstrar a sua implementação, utilize a biblioteca *pthread*s e descreva duas aplicações. A primeira aplicação consiste no problema dos produtores / consumidores (disponibilizada na Aula 07 em material auxiliar). As regiões críticas devem ser protegidas por sua implementação de *mutex* (o restante da sincronização com semáforos contadores deve ser mantida). A segunda aplicação consiste no problema dos leitores e escritores (com prioridade para os escritores, disponibilizada na Aula 08 nos slides), que deve ter as regiões críticas protegidas por sua implementação de *mutex*. O número de *threads* nas duas aplicações deve ser parametrizável (definido em uma macro ou passado como parâmetro para o programa).

O trabalho deve ser realizado em duplas. Não esqueçam de identificar-se em um comentário no início do código enviado por um dos integrantes pelo Moodle.