

Sistemas Operacionais - Trabalho Prático I

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 pthreads. Escolha algum algoritmo que implementa exclusão mútua por software e que seja generalizado para N processos / threads - por exemplo o algoritmo de Dekker ou o algoritmo de Peterson.

Para demonstrar a sua implementação, utilize a biblioteca pthreads e descreva duas aplicações. A primeira aplicação consiste no problema dos produtores / consumidores (consulte o material de aula sobre sincronização com semáforos). 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, consulte o material de aula com exercícios sobre sincronização), 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.

O trabalho deverá ser postado no Moodle até as 08hs do dia 16/10/2018.