

Laboratório 5

Nome: Riquelme Freitas Gomes

DRE: 120032785

Atividade 1

2. O TAD lista implementado nesses arquivos poderia ser compartilhado por threads de um programa concorrente? Com qual finalidade e de que forma?

Sim, o TAD lista implementado poderia ser compartilhado por threads de um programa concorrente. A finalidade seria a de utilizá-lo em situações em que várias threads precisam acessar e manipular o mesmo TAD lista, cooperando na realização de diversas ações sobre a lista como inserção, exclusão e pesquisa de elementos. De forma que seja garantida a exclusão mútua entre as threads conforme necessário, evitando problemas com a execução concorrente das operações de leitura e escrita e evitando a corrida de dados.

4. O que poderia acontecer se o programa não implementasse exclusão mútua no acesso às operações da lista encadeada?

Se o programa não implementasse exclusão mútua no acesso às operações da lista encadeada poderiam ocorrer diversos problemas na execução do programa, como corrida de dados, acesso a uma área de memória inválida e corrupção de estruturas de dados, tornando o programa instável e propenso a erros.

6. O que acontece com o tempo de execução do programa quando aumentamos o número de threads? Por que isso ocorre?

Quando aumentamos o número de threads, o tempo de execução do programa aumenta. Isso ocorre porque o programa usa mutex para garantir exclusão mútua ao acessar a lista compartilhada e quando o número de threads aumenta, isso leva a um aumento no tempo de espera das threads, já que elas ficam enfileiradas e precisam competir pela posse do mutex. Além disso, quanto mais threads, maior é o overhead e mais trocas de contexto ocorrem, ao passo que não há ganho de tempo por concorrência nas ações delimitadas pela exclusão mútua.

Atividade 2

4. Em quais cenários o uso do rwlock pode ser mais vantajoso que o uso do lock de exclusão mútua?

O rwlock pode ser mais vantajoso do que o lock de exclusão mútua em cenários nos quais a concorrência entre ações de leitura é permitida e que exigem, proporcionalmente, uma quantidade grande/dominante de operações de leitura.

Atividade 3

2. Em que tipo de aplicação concorrente o padrão de sincronização coletiva com barreira precisa ser usado?

O padrão de sincronização coletiva com barreira precisa ser utilizado em aplicações concorrentes onde as threads precisam coordenar suas atividades em etapas ou fases específicas do processo e não podem prosseguir para a próxima

etapa até que todas as threads tenham concluído a etapa atual. Isto é, nos casos em que se pretende suspender a execução das threads de uma aplicação em um dado ponto do código e somente permitir que as threads prossigam quando todas elas tiverem chegado naquele ponto.