Trabalho - Uso de Locks e Variáveis de Condição

EEL/770 - DEL/Poli/UFRJ 2023/1

1 Entrega e Pontuação

A data de entrega dos trabalhos é 18/07/2022 às 23:59 e tem a pontuação de 1,8 pontos. O sistema fecha após o prazo final. Não haverá adiamento da entrega. O código deve ser entregue pelo Moodle no mesmo link no qual este enunciado está disponível. O código deve estar todo em um único arquivo .zip ou .rar.

Coloque apenas seu numero da dupla no arquivo .zip ou .rar, no formato "duplaXX.zip" ou "duplaXX.rar" . Ao zipar um diretório, certifique-se que tal diretório possua o nome "duplaXX". Ou seja, não zipe um diretório com um nome genérico (p.ex., "Trabalho 1") para depois alterar o nome do arquivo zip. ATENÇÃO: Não coloque executáveis no arquivo .zip ou .rar. Enviem apenas o código fonte, com o Makefile.

Não aceitarei trabalhos enviados de outra forma, que não seja pelo Moodle.

2 Objetivo do Trabalho

O livro disponível em http://greenteapress.com/semaphores/LittleBookOfSemaphores.pdf possui diversos problemas de sincronização, apresentando sua solução com semáforos. O objetivo deste trabalho é implementar um problema (veja Seção 3) desse livro, com locks e variáveis de condição.

3 Escolha e implementação do problema

Você deverá implementar a solução do problema "The Santa Claus problem" (Seção 5.5).

O livro apresenta soluções e dicas. Você poderá usar a solução do autor do livro, mas adaptada para variáveis de condição, como também poderá utilizar sua própria solução. A ideia desse trabalho é ver um pouco, na prática, o uso dessas primitivas de sincronização. Se quiser entender a solução do autor, você deverá assistir os vídeos de Semáforos da disciplina, referente ao Conjunto 8.

A maior parte dos problemas do livro são analogias com o mundo real, então você não conseguirá implementar de fato a função. Por exemplo, no problema do barbeiro, você não vai fazer uma função que corte o cabelo :) . Você deve apenas imprimir algo do tipo "Cortando o Cabelo".

As entidades do problema que competem por recursos devem ser threads. Por exemplo: cada produtor/consumidor é uma thread; cada filósofo (video de Deadlocks do Conjunto 8) é uma thread, etc. Os recursos podem ser considerados como variáveis do programa. Por exemplo, no jantar dos filósofos, você pode ter cinco variáveis inteiras (isto é hashi1,hashi2,hashi3,hashi4,hashi5) que assumem valor 1 caso estejam em uso por um filósofo e 0 senão.

Quando um problema não possuir um número fixo de threads, seu programa deverá ser facilmente parametrizável para ser testado com um número arbitrário de threads. Por exemplo, no problema do jantar dos filósofos, o número de threads é sempre cinco. Entretanto, no problema do Dinning Hall, o seu código deverá funcionar para um número arbitrário de estudantes (threads). A parametrização do número

de threads pode ser feita por solicitação no início do programa ou por macro. Nesse último caso, deixe explícito no código quais são as macros de parametrização.

Dica 1: Apesar de não ser obrigatório para o trabalho, assistir os vídeos do Conjunto 8 pode ajudar no entendimento da solução do livro e nos problemas de deadlock que podem aparecer. O vídeo de deadlocks, por exemplo, mostra mais uma analogia, com o problema do jantar dos filósofos.

Dica 2: No Moodle há vídeos de revisão de C, que podem ajudar no trabalho.

4 Regras

- O trabalho deve ser feito como o mesmo grupo do laboratório;
- O programa deve ser escrito em C;
- Utilize a API pthreads para threads, locks e variáveis de condição. Não use outra API de threads;
- Não use semáforos;
- O código deve vir acompanhado de Makefile. Códigos que não forem entregues em condições de serem compilados não serão corrigidos;
- Lembrem-se de escolher nomes adequados para as variáveis e identar;
- Muito importante: O código deve ser comentado para permitir o entendimento do raciocínio utilizado por vocês para escrever o programa e para solucionar o problema escolhido;
- Caso o número de threads do problema não seja fixo, seu código deve ser facilmente parametrizável para um número arbitrário de threads (veja Seção 3);
- Seu programa deverá executar por várias iterações. Escolha um critério de parada para evitar que o código execute indefinidamente. Por exemplo, no jantar dos filósofos, o critério de parada pode ser definir que o programa finaliza quando 100 ações de comer forem executadas. Outro exemplo, no problema de produtores/consumidores, é definir que o programa finaliza quando houver, pelo menos, 100 ações de produção e 50 ações de consumo. O critério de parada deve ser facilmente parametrizável, por solicitação no início do programa ou por macro;
- Seu código deve ser livre de deadlocks. Uma dica é fazer um script de teste que chame milhares de vezes seu código em sequência. Essa é uma forma rudimentar de testar a existência de deadlocks, mas não garante a sua ausência. Lembre-se que deadlocks são difíceis de detectar, tente desenvolver o código pensando em eliminá-los. ;
- Caso se detecte cópia em trabalhos, seja entre colegas (desta turma ou de turmas anteriores), seja da Internet, o(a) aluno(a) receberá zero na nota do trabalho. No caso de cópia entre colegas, esta punição se dará em todos(as) os(as) envolvidos(as).
- É possível usar pequenos trechos de código da Internet, como do Stack Overflow, para funções secundárias como imprimir conteúdo de variáveis, solicitar entradas de usuários, etc. Nesse caso, deixe explícito no comentário o link utilizado.
- O ChatGPT pode ajudar vocês, mas usem com moderação!
- Vocês podem tirar dúvidas comigo por e-mail. Mas aceitarei dúvidas enviadas por e-mail somente até 28/06/2023 às 23:59
- Não haverá adiamento do prazo. O prazo foi escolhido de forma que eu consiga corrigir os trabalhos até o final do período.

5 Plataforma

O código poderá ser desenvolvido para o Linux, para o MacOS, ou para qualquer outro sistema UNIX. Informe no início do código qual sistema operacional você usou para fazer o trabalho. No caso do Linux, informe também a distribuição (Ubuntu, Debian, VM fornecida na disciplina, etc.).

O WSL 2 do Windows pode ser usado. Entretanto, seu programa será corrigido pelo professor no Linux, utilizando a mesma distribuição que você utilizou no WSL.