

Trabalho Prático 2 – 20 pontos – (versão 01)

1 Objetivo

O objetivo deste trabalho é que o(a) aluno(a) tenha contato direto com conceitos estudados na disciplina SIN 142 e desenvolva um *software* para gerenciar um recurso que precisa de exclusão mútua.

2 Descrição

O trabalho tem inspiração na contagem valores de várias fontes numa mesma variável e ao mesmo tempo, como contagem de visualizações, comentários, *likes*, *dislikes* de um vídeo no *Youtube* ou um *post* em alguma rede social. O vídeo *Why Computers Can't Count Sometimes* de Tom Scott (vídeo “original”: https://youtu.be/RY_2gElt3SA, e com legendas: <https://youtu.be/GfxkfG34Iw0> – no canal do Scott está bloqueado para envio de legendas) explica esse desafio.

Para os computadores contar é simples, fácil e rápido, contudo quando várias *threads* fazem isso ao mesmo tempo numa mesma variável, vários valores podem ser alcançados como resultado final. Neste contexto de muitas *threads* e contagem de valores é necessário desenvolver um programa/*software* para gerenciar a contagem correta dos valores (e.g., quantidade de *likes*), sendo obrigatório a exclusão mútua durante cada contagem. O projeto deve ser feito em grupos de 5 pessoas.

2.1 Contagem

O problema contará com 10 *threads* tentando atualizar a contagem de *views* (visualizações), *likes* (curtidas) e *dislikes* (descurtidas) de um vídeo. Cada *thread* deverá ler uma linha do arquivo atualizarValores.txt e atualizar a contagem dos 3 valores citados anteriormente. Cada linha terá: *views likes dislike*, separados por um espaço. Cada *thread* irá ler uma linha e atualizar os valores, as outras *threads* as próximas linhas até o fim do arquivo.

Gerencie para que mais de uma *thread* não leia a mesma linha. Depois de uma atualização nos valores, verifique no arquivo diminuirValores.txt na linha lida pela *thread* atual para descontar da contagem dos 3 valores. Esses valores para descontar são ditos como não válidos, por motivos diversos (e. g., usuário repetido), para contabilizar na atualização anterior. No final o programa terá que gerenciar o trabalho das *threads* (simulando as atualizações) e como o resultado o mesmo que um programa sequencial chegaria.

Para obter os arquivos de valores (atualizarValores.txt e diminuirValores.txt), utilize o programa sin142trabalhoPratico2GeraValores.c, alterando a *seed* para a soma dos integrantes do grupo.

3 Implementação

Para a implementação do trabalho utilize a linguagem C++ ou Java. O código não precisa ter interface gráfica (GUI).

4 Avaliação

Será avaliado o código fonte e o trabalho escrito, com 20 pontos no total, com 14 para código e 6 para trabalho escrito.

4.1 Código fonte

Será analisado as boas práticas de programação (e.g., refatoração, indentação) e se a solução

está correta.

4.2 Trabalho Escrito

No relatório será analisado a escrita, a descrição da solução desenvolvida. Coloque no relatório um breve introdução ao problema/cenário; uma descrição rápida da linguagem escolhida e seu suporte para a programação concorrente; um manual detalhado do *software* desenvolvido; principais partes do código e descreva seu funcionamento.

* O trabalho escrito deve ter de 3 até 9 páginas.

* Busque por referências na literatura e trabalhos similares para utilizar como base/referência.

5 Observações

A entrega do trabalho será via PVANet até **04/12/2019**. Um arquivo zip (“mat1_mat2_mat3_mat4_mat5.zip”) contendo os arquivos do projeto. Os projetos serão comparados em busca de similaridades. Os alunos podem ser escolhidos para explicar o trabalho. A ferramenta *Git* deve ser utilizada para controle de versão, envie o repositório (verificação de pelo menos 10 *commits* em dias variados).

6 Possibilidades de Bônus

Algumas funcionalidades extras podem ser implementadas para gerar uma bonificação na nota. Dentre as possibilidades estão:

- (+ 0,5) Uso de recursos de interface gráfica (GUI), entenda por qualquer coisa além do console (CLI).
- (+ 0,5) Uso do *Latex* para geração do trabalho escrito, assim deverá enviar o código fonte (.tex).

7 Dicas

- Faça um programa sequencial primeiro e depois coloque *threads*/concorrência nele.
- Não deixe para última hora.
- Organize a equipe e reserve um tempo na semana para trabalhar no projeto.
- A troca de informações e o trabalho em conjunto é importante no aprendizado e desenvolvimento do trabalho.
- Todos do grupo devem saber como os programas funcionam.

Bons estudos e bom trabalho :)!