Módulo 2 - Trabalho de Implementação (2021/1 REMOTO)

Computação Concorrente (MAB-117)
Prof. Silvana Rossetto

[1]Instituto de Computação/UFRJ — 15 de setembro de 2021

1. Descrição

Considere um arquivo com valores numéricos inteiros em ordem aleatória. O objetivo deste trabalho é gerar um arquivo de saída com os mesmos valores, porém parcialmente ordenados (ordem crescente) em blocos de tamanho N. Um exemplo reduzido é mostrado abaixo, fazendo $N = 5$.

```
     <Arquivo de entrada>

20
50 2 3 4 0
2 7 7 -1 2
1 -9 9 4 0
-5 2 3 5 7

     <Arquivo de saída>

0 2 3 4 50
-9 0 1 4 9
-1 2 2 7 7
-5 2 3 5 7
```

A primeira linha do arquivo de entrada contem um número inteiro com a quantidade de valores no restante do arquivo. Podemos ver que o arquivo de saída contem os mesmos valores do arquivo de entrada, porém, cada bloco consecutivo de $N = 5$ valores agora está ordenado em ordem crescente. Outro ponto a se observar é que os blocos do arquivo de saída **podem, mas não precisam manter a mesma posição do arquivo de entrada** (no exemplo acima, vemos que o segundo e terceiro bloco de $N = 5$ valores alternam suas posições no arquivo de saída).

Para desevolver uma solução concorrente para esse problema, as seguintes decisões de projeto já são determinadas:

1. A **leitura dos blocos de valores do arquivo para a memória e a ordenação interna desses blocos** será feita usando o padrão **produtores/consumidores**. Teremos **apenas uma thread produtora** que será responsável por ler os blocos do arquivo e trazê-los para um buffer compartilhado na memória. Teremos C $(C >= 1)$ **threads consumidores** que tomarão os blocos de valores carregados no buffer isoladamente e os ordenarão em ordem crescente.
2. A **escrita dos blocos ordenados para o arquivo de saída** será feita usando o padrão **leitores/escritores**. Nesse caso teremos apenas threads escritoras. As C threads que ordenaram os blocos escreverão os resultados no arquivo de saída comum, respeitando o requisito de que apenas uma thread pode escrever de cada vez.

3. O número de threads C e o tamanho do bloco N deverá ser passado na entrada do programa, assim como os nomes dos arquivos de entrada e de saída.
4. O número de entradas no buffer do padrão produtores/consumidores deve ser fixo e igual a 10 (ou seja, deve ser capaz de armazenar 10 blocos de tamanho N cada um).
5. A quantidade de valores no arquivo de entrada deverá sempre ser um múltiplo de 10 para simplificar a divisão em blocos.

Dessa forma, é possível processar arquivos de entrada de qualquer tamanho. A solução poderá ser implementada na linguagem C, usando a biblioteca Pthreads, ou em Java.

2. Etapas do trabalho

A execução do trabalho deverá ser organizada nas seguintes etapas:

1. Compreender bem a descrição do trabalho, o formato dos dados de entrada e saída e o pre-projeto da solução concorrente;
2. Completar o projeto da solução concorrente para o problema e as estruturas de dados que serão usadas;
3. Construir um conjunto de casos de teste para avaliação da solução proposta (testes de corretude e testes de desempenho);
4. Implementar a solução projetada, avaliar a sua corretude, refinar a implementação e refazer os testes.
5. Avaliar o ganho de desempenho obtido, considerando diferentes dimensões do arquivo de entrada e do número de threads consumidoras C criadas (não é necessário ter uma versão sequencial da solução, apenas comparar o ganho de desempenho obtido variando-se o número de threads que processam a ordenação).
6. Redigir o relatório.

3. Artefatos que deverão ser entregues

- **Relatório final**: descrição do projeto completo da solução concorrente e dos testes de corretude e de desempenho realizados e outras discussões (vide modelo para o relatório em anexo);
- **Código fonte**: link para o repositório do código desenvolvido.

4. Critérios de avaliação

Os seguintes itens serão avaliados no trabalho com o respectivos pesos:

- Projeto final da solução concorrente e discussões associadas (no relatório): **1 ponto**
- Interface de uso, organização e documentação do código fonte (no código fonte): **1 ponto**
- Descrição dos casos de testes de corretude e de desempenho realizados (no relatório): **2 pontos**
- Execução correta da solução concorrente (no relatório e na correção): **5 pontos**
- Ganho de desempenho obtido com a solução concorrente (no relatório e na correção): **1 ponto**

O trabalho pode ser feito individualmente ou em **dupla** (preferencial). Os integrantes da equipe poderão ser chamados pela professora para explicar o trabalho.