

Módulo 3 - Trabalho de Implementação (2020/1 REMOTO)

Computação Concorrente (MAB-117)

Prof. Silvana Rossetto

¹DCC/IM/UFRJ

22 de fevereiro de 2021

Descrição do problema

Para este trabalho, vamos implementar um **programa concorrente para leitura e processamento de arquivos**. Os arquivos para processamento conterão uma sequência de números inteiros de 0 a 5. O processamento consistirá na busca dos seguintes padrões nessa sequência de números:

- (a) Maior sequência de valores idênticos;
- (b) Quantidade de ocorrências de sequências contínuas de tamanho 3 do mesmo valor;
- (c) Quantidade de ocorrências da sequência <012345>.

Restrições da solução concorrente

O arquivo de entrada deverá ser lido em **blocos** de tamanho N e armazenado em um **buffer** com M entradas. Uma thread em separado deverá implementar essa funcionalidade, de leitura do arquivo e escrita no buffer. As demais threads do programa deverão implementar as buscas pelos padrões desejados, uma thread para cada padrão. Para isso, essas threads deverão ler os blocos de dados armazenados no buffer.

Os blocos deverão ser inseridos no buffer na mesma ordem em que aparecem no arquivo. Os blocos do buffer deverão ser lidos e processados na mesma ordem em que foram inseridos no buffer. Um bloco só poderá ser retirado do buffer depois que todas as threads que fazem a busca por padrões tiverem terminado o seu processamento naquele bloco (as threads poderão ler os dados do bloco ao mesmo tempo).

Com essa organização, novos padrões de busca podem ser acrescentados e os atuais excluídos do programa com mínima interferência no restante do código.

A solução proposta deverá ser implementada em C.

Para avaliar o programa desenvolvido: (i) verificar a sua corretude, considerando diferentes casos de arquivos de entrada; (ii) avaliar quais pares de valores $[N, M]$ resultam no menor tempo de execução do programa.

Formato dos dados de entrada e de saída

Os arquivos de entrada terão formato **binário** e conterão uma **sequência de números inteiros**. O primeiro valor dessa sequência (do tipo `long` `long` `int`) indicará a quantidade de valores contidos no restante do arquivo. (Para gerar um arquivo binário de teste, use a função `fwrite`, e para ler os dados do arquivo binário use a função `fread`.)

Os dados de saída deverão ser impressos na tela, na seguinte ordem:

- Maior sequência de valores idênticos: <posição inicial> <tamanho da sequência> <valor da sequência>
- Quantidade de ocorrências de sequências contínuas de tamanho 3 do mesmo valor: <quantidade>
- Quantidade de ocorrências da sequência <012345>: <quantidade>

Exemplo de arquivo de entrada (traduzido para forma textual):

30 1 2 3 1 1 1 1 2 3 4 5 5 5 5 5 5 5 0 0 0 3 3 3 0 1 2 3 4 5 0

Saída esperada:

Maior sequência de valores idênticos: 11 7 5

Quantidade de triplas: 5

Quantidade de ocorrências da sequência <012345>: 1

Nesse exemplo, a maior sequência de valores idênticos é a sequência de 5 que começa na posição 11 e tem tamanho 7. As triplas são: uma de 1 (início na posição 4), duas de 5 (início na posição 11 e 14), uma de 0 (início na posição 18) e uma de 3 (início na posição 21). A sequência <012345> aparece uma única vez, a partir da posição 24.

Etapas do trabalho

A execução do trabalho deverá ser organizada nas seguintes etapas:

1. Projetar a solução concorrente para o problema e as estruturas de dados que serão usadas;
2. Construir um conjunto de casos de teste para avaliação da solução proposta;
3. Implementar a solução projetada, avaliar a sua corretude, refinar a implementação e refazer os testes;
4. Avaliar o ganho de desempenho obtido, considerando diferentes dimensões para: (i) o tamanho dos blocos de dados lidos do arquivo; e (ii) o tamanho do *buffer* de armazenamento desses blocos.
5. Redigir o relatório.

Artefatos que deverão ser entregues

- **Relatório:** projeto da solução concorrente, projeto dos testes e avaliação geral da solução;
- **Código fonte:** link para o repositório do código desenvolvido.

O trabalho poderá ser feito em **dupla** e deverá ser entregue até o dia **1 de março de 2021**, via formulário.

Critérios de avaliação

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

- Projeto da solução concorrente (no relatório): **2 pontos**
- Interface de uso, organização e documentação do código fonte: **1 ponto**
- Projeto dos testes e avaliação geral da aplicação (no relatório) **3 pontos**
- Execução correta da solução desenvolvida: **4 pontos**