

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

Computação Concorrente (MAB-117) ~

Prof. Silvana Rossetto

DCC/IM/UFRJ

#### Projeto:

Thread de leitura: Lê o arquivo em blocos de tamanho N e guarda em um array de tamanho M, quando esse array estiver cheio, faz a cópia para o buffer principal e marca o bloco como concluído.

Demais Threads: Percorrem o buffer principal realizando suas tarefas, pedindo acesso aos blocos para a thread de leitura. As threads são;

Thread sequencial igual: percorre o buffer guardando o último elemento lido, se o elemento atual for igual ao último incrementa o contador "seq", senão zera o contador. No momento de zerar o contador verifica se a sequência atual é maior do que a maior sequência guardada (int maior, iniciada em -1), atualizando os valores se maior.

Thread triplas: Percorre o buffer principal guardando o último elemento lido, se o último elemento lido for igual ao atual incrementa o contador "seq", senão seq = 0. Quando seq == 3 incrementa a variável "qtd" em um e zera seq, printa o valor de qtd ao final da execução.

Thread sequência crescente: Percorre o buffer principal guardando o último elemento lido, se o último elemento for igual a atual - 1, incrementa a variável "seq" senão seq = 0, se seq == 6 incrementa a variável qtd. Printa o valor de qtd ao final da execução

#### Testes:

É testado um arquivo por vez, testando com diferentes valores para N e M, começando com valores pequenos e crescendo ao final dos testes.

#### Implementação:

A solução funciona perfeitamente para arquivos pequenos, porém verificar a corretude em arquivos grandes (milhões de elementos) é mais difícil. Sempre que testado fixando uma parte do arquivo a escolha o programa age de forma esperada e encontra os padrões ali colocados, porém não a garantia de que em um arquivo com milhões de elementos ele encontre todos os padrões de forma correta.

#### Avaliação:

Os testes realizados indicam pouca variação no tempo de execução com a mudança dos valores de N e M, salvo o caso onde N = M = número de elementos, onde há um grande ganho de desempenho.