

Mini EP 5: Control-F

Alfredo Goldman, Elisa Silva e Luciana Marques
MAC 0219-5742 – Programação Concorrente e Paralela 2021

Entrega até 31 de maio de 2021

1. Introdução

Buscar a ocorrência de um certo trecho dentro de um texto é uma atividade muito usual. Como tal, uma boa implementação deve ser capaz de explorar as capacidades de processamento paralelo dos computadores modernos para garantir uma boa experiência ao usuário.

Neste quinto mini EP iremos implementar um buscador de substrings paralelo usando Pthreads em C. Vocês devem partir de um código base que implementa o algoritmo Knuth-Morris-Pratt e também traz funcionalidades de medição e verificação.

Seguindo o caminho do último mini EP, a especificação será menos específica para incentivar a experimentação e descoberta. Mas não deixe de buscar os monitores para tirar eventuais dúvidas.

2. Tarefas

Baixe do eDisciplinas o arquivo “MiniEP5.zip”. Ao descompactá-lo, você irá encontrar uma pasta com o nome “MiniEP5” contendo os arquivos: “ep.c”, “huge.txt”, “large.txt” e “medium.txt”.

Durante o desenvolvimento, certifique-se que o programa gerado roda no mesmo diretório dos arquivos de textos para ele funcionar.

Primeiro, edite o arquivo “ep.c” para que as variáveis NAME e NUSP reflitam seu nome e número USP respectivamente.

Depois, você deve editar a função **task** que devolve o número de ocorrências da substring *search* dentro da string *data* de tamanho *len* (*data* é uma string que termina com o caractere ‘\0’).

Ela deve usar o valor da variável *threads* para criar *threads* que computam partes diferentes da string *data* a fim de reduzir o tempo de execução. Não se preocupe com o fato de já existir a *thread* principal, se o valor de *threads* for 1, crie uma *thread* e execute a contagem nela. Antes de devolver o número de ocorrências, a função **task** deve encerrar todas as *threads* criadas.

Uma característica do algoritmo usado neste mini EP é a que se duas ocorrências se sobrepõem, apenas a primeira é considerada. Segue o exemplo: ocorrências de ABA em ABABA, apesar de ocorrer duas vezes a substring ABA dentro de ABABA, só a primeira será considerada, pois a segunda sobrepõe a primeira.

O programa executa 6 *queries* diferentes, cada uma com {1,2,4 e 8} threads, 5 vezes cada. Os tempos são apresentados em milissegundos.

É esperado que uma boa implementação além de correta, observe um ganho de pelo menos 20% para a última *query* usando várias *threads* em relação à versão sequencial.

3. Entrega

Salve a saída do programa em um arquivo “ep.out” e envie ambos os arquivos “ep.c” e “ep.out” em um arquivo .ZIP com o seu número USP como nome.

Não deixe de preencher o formulário de feedback sobre a atividade:
<https://forms.gle/nMx6T7wcH4Ny72Xy8>. Seu preenchimento não exige identificação.

Entrega até 31 de maio de 2021.

4. Critério de Avaliação

Os Mini EPs usam um critério de avaliação binária (ou 1 ou 0). Para tirar 1 envie .ZIP conforme especificado no eDisciplinas. Não será avaliada a qualidade do código, mas tenha zelo.

Vale reforçar parágrafo II do artigo 23 do **Código de Ética da USP**:

Artigo 23 - É vedado aos membros do corpo docente e demais alunos da Universidade:

[...]

II. lançar mão de meios e artifícios que possam fraudar a avaliação do desempenho, seu ou de outrem, em atividades acadêmicas, culturais, artísticas, desportivas e sociais, no âmbito da Universidade, e acobertar a eventual utilização desses meios.

Mini EPs plagiados receberão nota 0.

Se tiver dúvidas, envie uma mensagem no fórum do curso ou envie e-mails para elisa@silva.moe, lucianadacostamarques@gmail.com ou gold@ime.usp.br com *[miniEP5]* no assunto do e-mail. Divirta-se!