

Relatório do 2º Trabalho Prático da Disciplina Sistemas Operacionais 1

Aluno: Hiago Antonio Costa Silva

Professor: Francisco José da Silva e Silva

SUMÁRIO

| | |
|-----------------------------|---|
| Especificações: | 3 |
| Relatório de Bugs: | 3 |
| Dificuldades: | 3 |
| Execução do Programa | 4 |
| Testes Feitos | 4 |
| Considerações Finais | 4 |

Especificações:

O Trabalho foi feito e executado em:

Compilador: Code::Blocks 20.03

Linguagem de Programação: C

Sistema Operacional: Windows 10 Pro x64

Computador: Notebook Dell Inspiron 15 5000

Processador: Intel(R) Core(TM) i7-5500U CPU @ 2.39GHz

Memória RAM: 8GB

Núcleos: 2

Processadores Lógicos: 4

Relatório de Bugs:

Foi encontrado 1 Bug:

A divisão dos macroblocos está funcionando perfeitamente, porém na função das threads, se o tamanho da matriz não for proporcional ao tamanho das threads, ocorrerão os seguintes erros:

Se o número de linhas da matriz não for proporcional ao número de linhas dos macroblocos, não ocorrerá a contagem de primos.

Se o número de colunas da matriz não for proporcional ao número de colunas dos macroblocos, a contagem dos números ocorrerá de forma errada, contabilizando um número a mais de primos.

Dificuldades:

Algumas dificuldades foram encontradas nas em instanciar por `#define` no `function.h`, exemplos destes são o número de threads e a seed utilizada, então para resolver esse problema, foram criadas variáveis globais no `function.c`, assim possibilitando o andamento do trabalho.

Outra dificuldade foi na divisão dos macroblocos, a divisão não estava acontecendo com um número fixo de macroblocos, então foi utilizado outro tipo de parâmetro, sendo ele o tamanho dos macroblocos, ainda que apresentem alguns bugs, assim como relatado anteriormente nos Bugs.

A última dificuldade aconteceu em contabilizar o tempo de execução do programa, não consegui implementar o sistema por `Profile`, então busquei outra alternativa utilizando a biblioteca `<time.h>`, assim conseguindo obter os resultados.

Execução do Programa

Para executar o programa basta iniciar o arquivo main.c, ele pedirá para colocar as dimensões da matriz, as dimensões dos macroblocos e quantas threads serão criadas. Para não ocorrer nenhum erro, coloque medidas proporcionais nas dimensões em matriz X macroblocos.

Testes Feitos

Os seguintes testes foram feitos com a Seed padrão sendo 50

| Matrizes | Macroblocos | Quant. Threads | Tempo Sequencial | Tempo Paralelo | Quant. Primos Seq. | Quant. Primos Paralelo |
|-------------|-------------|----------------|------------------|----------------|--------------------|------------------------|
| 2500x2500 | 25x25 | 2 | 2.007 | 1.025 | 695946 | 695946 |
| | | 3 | 2.012 | 0.771 | //// | //// |
| | | 4 | 2.013 | 0.657 | //// | //// |
| 10000x2500 | 50x50 | 2 | 8.070 | 4.070 | 27873331 | 27873331 |
| | | 3 | 8.046 | 3.146 | //// | //// |
| | | 4 | 8.037 | 2.641 | //// | //// |
| 20000x20000 | 20x10 | 2 | 129.420 | 70.375 | 44541214 | 44541214 |
| | | 3 | 134.718 | 59.400 | //// | //// |
| | | 4 | 129.470 | 47.256 | //// | //// |

Acima de 4 threads o resultado seguiu sendo o mesmo por limitação do meu computador

Considerações Finais

Ainda que o trabalho não tenha sido feito fielmente como pedido, acredito que me ensinou bastante na prática sobre a implementação e utilização das threads além de suas vantagens no tempo de processamento de um programa. Os resultados foram bastante satisfatórios, pois como pode se notar, o tempo de processamento foi otimizado até quase 4x com a utilização do máximo de threads que meu computador proporciona.