

UNIVERSIDADE DO ESTADO DO RIO DE JANEIRO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
SISTEMAS OPERACIONAIS I
PROFESSOR: LEANDRO MARZULO
CURSO: CIÊNCIA DA COMPUTAÇÃO



Relatório Sequencial e Memória Compartilhada

INTEGRANTES:

CAMILA ELEUTÉRIO GUSMÃO
MURILO DE JESUS SANTOS SILVA
THAINÁ LOPES FIGUEIREDO

ENUNCIADO:

No site é fornecida uma versão sequencial de um programa em C que informa quantos números primos existem entre dois inteiros positivos informados pelo usuário. O trabalho consiste em paralelizar essa aplicação usando processos e memória compartilhadas, conforme visto no exemplo de somatório dos elementos de um vetor. O número de processos desejado também deve ser informado pelo usuário. O trabalho é em trio (quem não tiver trio pode fazer individualmente) e apenas um membro da dupla precisa submeter o trabalho. Um zip contendo o código C e um relatório (ou comentários no código) explicando a implementação e uma medição do tempo sequencial e o tempo para 2 processos, considerando que a faixa fornecida é de 1 a 10 milhões (desejamos saber quantos primos existem entre 1 e 10 milhões). Tanto o código fonte quanto o relatório devem conter os nomes dos membros da dupla. A versão sequencial do programa e o exemplo do somatório estão publicados no site.

EXPLICAÇÃO DA IMPLEMENTAÇÃO:

O programa foi implementado em um ambiente de testes composto por uma CPU Intel Core i52450M (quatro núcleos) 2,5 GHz, RAM de 4GB, Windows 10 Pro versão 1607 (Build 14393.321) e GCC4.4.5. Tanto na implementação dos primos na versão Sequencial quanto na memória compartilhada, encontramos um total de 664580 números primos na faixa fornecida de 1 a 10 milhões de números. Verificamos a medição do tempo durante a compilação utilizando o comando “time” para ambas as versões seus valores estão descritos na tabela abaixo. Percebe-se que o tempo Sys da Memória Compartilhada é maior que o da Sequencial porque é tempo que o CPU gasta no modo kernel para os processos. É possível verificar também que na comparação entre os tempos Real e Users a versão de Memória Compartilhada apresentou uma melhora no tempo do que a versão Sequencial, devido a utilização dos dois processos criados durante sua execução, confirmando que a utilização de processos otimiza o tempo de execução dos programas devido ao seu paralelismo.

Versão\Tempo	Real	Users	Sys
Sequencial	0m24.608s	0m24.516s	0m0.016s
Memória Compartilhada 2 processos	0m22.829s	0m22.797s	0m0.031s