

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



Relatório OpenMP e Pthreads

INTEGRANTES:

CAMILA ELEUTÉRIO GUSMÃO
MURILO DE JESUS SANTOS SILVA
THAINÁ LOPES FIGUEIREDO

ENUNCIADO:

Elaborar duas versões do programa contador de primo (o mesmo do trabalho 1): uma versão usando pthreads e outra usando OpenMP. Nas duas versões, o programa deverá receber, por linha de comando, os limites inferior e superior e o número de threads desejadas. No caso da versão OpenMP, você deverá usar compilação condicional (com `#ifdef _OPENMP`) conforme visto nos exemplos em sala de aula. A implementação em OpenMP deve usar a diretiva `parallel for` com redução. Você deverá entregar 1 zip com os dois programas e os tempos usando oversubscription, escalonamento "static", "dynamic" e "guided".

EXPLICAÇÃO DA IMPLEMENTAÇÃO:

O programa foi implementado em um ambiente de testes composto por uma CPU Intel Core i5 2450M (quatro núcleos) 2,5 GHz, RAM de 4GB, Windows 10 Pro versão 1607 (Build 14393.321) e GCC 4.4.5. Tanto na implementação dos primos na OpenMP quanto na versão de PThreads, 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. É possível verificar que na comparação entre os tempos Reale Sys, a versão de Pthreads apresentou uma melhora no tempo, devido a utilização dos 4 threads criadas durante sua execução. Porém o tempo de Users é menor na versão OpenMP. Isso confirma que a versão de Pthreads aperfeiçoa a paralelização, e que o OpenMP é uma técnica de paralelização menos trabalhosa.

Versão\Tempo	Real	Users	Sys
OpenMP	1m46.248s	1m44.469s	0m0.109s
Pthreads	0m56.231s	2m29.250s	0m0.172s