

Trabalho

Nota: 10,0 (P1)

Data Entrega: 25/10/2024

Individual

- Todos os programas podem ser implementados na linguagem que acharem melhor.
- O programa com troca de mensagens pode ser usando MPI ou outra ferramenta/biblioteca que realize a troca de mensagens

Implementar TRÊS programas para obter a Qtd. números primos em um vetor

- . Um programa usando memória compartilhada (processos);
- . Outro programa usando troca de mensagens (processos);
- . Outro programa usando threads;
- . Executar cada programa variando o número de threads ou processos de 1 a 4 que está sendo usado para execução dos testes.
- . Variar o tamanho do vetor: 100 ou 1000 (Alocação dinâmica)

Implementar TRÊS programas para obter a Qtd. números primos em um vetor

- . Obs: Cada programa deve criar a quantidade de processos especificada e balancear a carga de cada processo. O mesmo programa que executa com 1 processo será usado para executar com 2, 3 e 4, ou seja, a implementação de cada programa deve ser genérica.
- . Não será implementado um programa para rodar com 1 processo/thread, outro para rodar com 2 processos/threads, etc. É apenas UM programa que será implementado que servirá para executar com 1, 2, 3 e 4 processos/threads;

Qtd. números primos em um vetor

- . Tirar uma média do tempo de execução. Executar 5 vezes para cada combinação (tamanho do vetor e número de processos ou threads);
 - thread, 1 thread e vetor 100
 - thread, 1 thread e vetor 1000
 - thread, 2 thread e vetor 100
 - thread, 4 thread e vetor 1000
- . Entregar relatório com 1 gráfico: um com o tempo de execução para os dois tamanhos de vetor comparando a execução com memória comp, troca de mensagens e threads;

Observações

- Para obter o tempo de execução: `time ./a.out;`
 - `time: real`
- Obs: Pode usar o Cocalc
- Remover os `printf` do programa ao obter o tempo de cada execução.

- 04/10 e 18/10 - matéria normal
- 11/10 e 25/10 - Aulas para dar andamento no trabalho e tirar dúvidas