

Exercícios práticos

OpenMP

INE5410 - Programação Concorrente
Prof. Márcio Castro

1 Dicas úteis

Para realizar os exercícios a seguir, você necessitará de:

- Um editor de texto para escrever o seu código: escolha o editor de sua preferência (vim, emacs, nano, pico, gedit, ...)
- Um compilador: usaremos o gcc
- Um terminal: para compilar e executar o seu programa.

A sintaxe para compilar um programa em C com OpenMP é a seguinte:

```
$ gcc -o <nome_arquivo_binario> <nome_arquivo_contendo_o_código> -fopenmp
```

Por exemplo: para criar um programa chamado `meu_programa` a partir de um código em C chamado `meu_programa.c` faça:

```
$ gcc -o meu_programa meu_programa.c -fopenmp
```

Se tudo ocorrer bem, ao final da compilação será gerado um arquivo binário chamado `meu_programa`. Para executá-lo com XXXX threads faça:

```
$ export OMP_NUM_THREADS=XXXX  
$ ./meu_programa
```

Você deverá incluir pelo menos as seguintes bibliotecas nos seus códigos:

```
#include <stdio.h>  
#include <omp.h>
```

2 Exercícios

Exercício 1 O programa “calculo.c” realiza uma série de cálculos, armazenando os resultados em um arranjo de tamanho `SIZE`. Para compilá-lo adicione a flag de compilação `-lm`. Paralelize esse programa com OpenMP. Verifique o tempo de execução da versão sequencial e paralela com diferentes números de threads com o uso da ferramenta `time` do Linux (`time ./calculo`). **Dica:** remova todos os `printf()` para medir o tempo de execução corretamente.

Exercício 2 O programa “mm.c” calcula o produto de duas matrizes, armazenando os resultados em uma outra matriz. Paralelize esse programa com OpenMP. Verifique o tempo de execução da versão sequencial e paralela com diferentes números de threads com o uso da ferramenta `time` do Linux (`time ./calculo`). **Dica:** remova todos os `printf()` para medir o tempo de execução corretamente.