UnB/FGA/ Eng. Software / Programação para Sistemas Paralelos e Distribuídos
Turma A Prof. Fernando W Cruz

Exercícios de fixação - OpenMP

1. O programa abaixo está correto? Caso esteja errado, o que precisa ser modificado para que ele consiga processar corretamente?

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

int main(void) {
        int myid, nthreads;

        #pragma omp parallel private(myid) private(nthreads)
        {
             myid = omp_get_thread_num();

             #pragma omp single
             nthreads = omp_get_num_threads();
             printf("%d de %d - Hello, World!\n", myid, nthreads);
        }
        return 0;
} /* fim-main */
```

2. Supondo OMP_NUM_THREADS=4, a execução do código abaixo produzirá impressão da variável i com um valor sempre maior do que 10. Faça testes para conferir se esta afirmação está correta ou não.

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

int main(int argc, char *argv[]) {
        int i=0;

        #pragma omp parallel shared(i)
        {
                        i=i+10;
        }

        printf("i=%d\n", i);
        return 0;
}
```

3. Elaborar um programa para descobrir o número de iterações que cada thread consegue processar em uma aplicação OpenMP. Por exemplo, imagine uma operação qualquer sobre os elementos de um vetor e descubra quantos elementos cada thread manipulou.

4. O programa a seguir contabiliza a quantidade de números primos dentro de uma faixa de valores. Faça a medição do tempo de resposta deste programa (tempo de processamento) desta versão não-paralela. Em seguida, promova as alterações necessárias para que este programa funcione em paralelo e descubra qual quantidade de threads produz o melhor desempenho para essa aplicação (speedup).

```c
#include <stdio.h>
#include <omp.h>
#define N 300000

int main(void) {
    int i;
    int primos=0;
    for (i=N; i>1; i--) {
        int j=2;
        while (j<i) {
            if (i%j == 0)
                break;
            j++;
        } /* fim-while */
        if (j==i)
            primos++;

    } /* fim-for */
    printf("%d primos!\n", primos);
    return 0;
} /*fim-main */
```