

Ejercicios de clase sobre OpenMP
Metodología de la Programación Paralela
Curso 2019/20

Ejercicio 1. Modifica el programa siguiente para que calcule la suma de dos vectores en paralelo.

```
#include <stdio.h>
#define N 102400

int main(void)
{
    double vecA[N], vecB[N], vecC[N];
    double sum;
    int i;

    /* Inicialización de los vectores */
    /* Estos valores podrían leerse de fichero */
    /* en un escenario más realista */
    for (i = 0; i < N; i++) {
        vecA[i] = (double) i * N;
        vecB[i] = (double) i * i;
    }

    /* TODO:
     *   Implementa aquí la version paralela de:
     *   vecC = vecA + vecB
     */
    return 0;
}
```

Ejercicio 2. Paraleliza este programa que realiza el producto escalar de dos vectores. Utiliza una o más construcciones de OpenMP.

```
#include <stdio.h>
#define N 102400

int main(void)
{
    double vecA[N], vecB[N];
    double suma;
    int i;

    /* Inicialización de los vectores */
    /* Estos valores podrían leerse de fichero */
    /* en un escenario más realista */
    for (i = 0; i < N; i++) {
        vecA[i] = 1.0/((double) (N-i));
        vecB[i] = vecA[i] * vecA[i];
    }

    suma = 0.0;
    for (i = 0; i < N; i++) {
        suma += vecA[i] * vecB[i];
    }
    printf("Suma: %18.16f\n", suma);

    return 0;
}
```

Ejercicio 3. Paraleliza la siguiente función utilizando una sola región paralela (directiva parallel).

```
double ej(double x[M], double y[N], double A[M][N])
{
    int i,j;
    double aux,s=0.0;
    for (i=0; i<M; i++)
        x[i] = x[i]*x[i];
    for (i=0; i<N; i++)
        y[i] = 1.0+y[i];
    for (i=0; i<M; i++)
        for (j=0; j<N; j++) {
            aux = x[i]-y[j];
            A[i][j] = aux;
            s += aux;
        }
    return s;
}
```

Ejercicio 4. Escribe una versión paralela del siguiente código. Las funciones fun1, fun2 y fun3 son independientes entre sí.

```
int n=...;
double a,b[3];

a = -1.8;
fun1(n,&a);
b[0] = a;
a = 3.2;
fun2(n,&a);
b[1] = a;
a = 0.25;
fun3(n,&a);
b[2] = a;
```

Ejercicio 5. ¿Cuál es el error en este código?

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

int main (int argc, char *argv[]) {
    int nthreads, i, tid;
    float total;

    #pragma omp parallel
    {
        tid = omp_get_thread_num();
        if (tid == 0) {
            nthreads = omp_get_num_threads();
            printf("Number of threads = %d\n", nthreads);
        }
        printf("Thread %d is starting...\n",tid);

        #pragma omp barrier

        total = 0.0;
        #pragma omp for schedule(dynamic,10)
        for (i=0; i<1000000; i++)
            total = total + i*1.0;

        printf ("Thread %d is done! Total= %e\n",tid,total);
    }
}
```