

---

# High Performance Computing

## Departamento de Ingeniería en Informática

### Ejercicios

1. El siguiente código (simplificado) intenta que dos hebras compartan variables declaradas como `shared`. Asuma que se crean exactamente dos hebras:

```
1 main() {
2     int yo, vecino, flag[2];
3     #pragma omp parallel private(yo, vecino) shared(buffer, flag)
4     {
5         yo = omp_get_thread_num(); // quién soy yo
6         flag[yo] = 0;
7         #pragma barrier           // sincronizo
8         buffer[yo] = 5*yo;        // produzco un item
9         flag[yo] = 1;            // aviso que el dato esta en el buffer

10        vecino = (yo == 0 ? 1 : 0);
11        while (flag[vecino] == 0); // espero por el item

12        result = buffer[vecino]*buffer[yo]; // computo final
    }
}
```

Explique por qué este código es incorrecto y realice los cambios apropiados. Hint: el problema está asociado a la concurrencia.

2. Las siguientes líneas de códigos corresponden a la invocación de un kernel que traspone una matriz cuadrada:

```
dim3 blocksize;
dim3 gridsize;
gridsize.x = gridsize.y = N/512;
blocksize.x = blocksize.y = 512;
traspose<<<gridsize, blocksize>>>(A, N);
```

La trasposición se hace *in-situ*, es decir al retornar el kernel, *A* es la traspuesta. No puede usar una matriz adicional. Asuma que la memoria para *A* ya ha sido procurada, tanto en host como en GPU, como arreglos bidimensionales de  $N \times N$  flotantes. Escriba un kernel en C y CUDA que trasponga la matriz.

3. Suponga que le han solicitado construir un código paralelo para implementar el adversario en un juego de dominó. Diga con argumentación cuál de las tecnologías vistas en clases sería la más apropiada.
4. Sobre el paradigma SIMD
  - (a) Explique el paradigma.

- 
- (b) Comente sobre las similitudes y diferencias del paradigma SIMD implementado en registros MMX e implementado en hebras OpenMP.
5. Escriba un programa en OpenMP que utilice tareas para encontrar el valor máximo de un arreglo de largo  $N = 2^{2^k}$ . Asuma que el arreglo es de enteros y los elementos ya han sido almacenados en el arreglo. Utilice  $H$  hebras. Al finalizar el programa debe imprimir el valor máximo encontrado.
6. Demuestre que el speedup según Gustafson puede expresarse como

$$S(n) = n + (1 - n) \times s$$

donde  $s$  es la fracción del tiempo secuencial en la ejecución del programa.

7. Si la fracción secuencial de un programa paralelo es 1% y se ejecuta en un multicore de 32 núcleos, ¿Cuál es el speedup según Amdahl's?
8. En un sistema dual core, donde cada core tiene dos clusters SSE para cómputo vectorial de 128 bits, ambos para operaciones de punto flotante de precisión simple, ¿Cuál sería el máximo speedup posible que cualquier aplicación podría alcanzar?