

## Práctica Nro. 2

Programación con Pthreads

---

***Se seleccionará un ejercicio que los alumnos deberán entregar en grupos de dos personas.***

*Pautas:*

*Compilar en Linux gcc:*

*`gcc -pthread -o salidaEjecutable archivoFuente`*

*De ser posible calcular el speedup y la eficiencia comparando con la implementación secuencial de los algoritmos.*

*Implementar utilizando sincronización mediante semáforos y mediante variables condición.*

1. Implementar la multiplicación de matrices cuadradas de  $N \times N$  por medio de  $T$  threads. Tomar el tiempo al ejecutarla con diferentes valores de  $N=512, 1024$  y  $2048$ . Para cada valor de  $N$  usar  $T=2, 4, 8$  y  $16$ .
2. Realizar un algoritmo que dado un elemento  $X$  indique la cantidad de veces que aparece dentro de un vector de enteros de  $N$  elementos. Se cuenta con una cantidad  $T$  de threads buscadores. Al final la cantidad de veces que aparece el elemento  $X$  debe quedar en una variable llamada *ocurrencias*. Probar con los siguientes valores de  $T=2, 4, 8$  y  $16$ .
3. Realizar la búsqueda del mínimo y el máximo valor en un vector de  $N$  elementos. Paralelizar con  $2, 4$  y  $8$  Threads.
4. Realizar un algoritmo paralelo que ordene un vector de  $N$  elementos por mezcla. Paralelizar con  $2, 4$  y  $8$  Threads.
5. Realizar un algoritmo paralelo que calcule el promedio de  $N$  elementos almacenados en un vector. Paralelizar con  $2, 4$  y  $8$  Threads.

6. Realizar un algoritmo paralelo con 4 threads que resuelva la siguiente operación:

$$C = (b * A) * (a * B)$$

Donde A y B son matrices de NxN, a es el valor promedio de los elementos de la matriz A y b es el valor promedio de los elementos de la matriz B.

Probar para N= 512, 1024 y 2048.

Calcular el speedup y la eficiencia comparando con la implementación secuencial del algoritmo.