

## Práctico 1 – Memoria compartida

1. Para el siguiente programa concurrente suponga:

- Que las instrucciones del siguiente código no son atómicas (la ejecución puede ser interrumpida).
- Todas las variables están inicializadas en 0 antes de empezar.

Indique cual/es de las siguientes opciones son verdaderas:

- a) En algún caso el valor de x al terminar el programa es 188.
- b) En algún caso el valor de x al terminar el programa es 95.
- c) En algún caso el valor de x al terminar el programa es 942.

P1::	P2::	P3::
If (x = 0) then y:= 4*23; x:= y + 2;	If (x > 0) then x:= x + 1;	x:= (x*8) + x*2;

Todas las opciones resultan verdaderas: "a" (188), "b" (95) y "c" (942).

**P1::**

If (x = 0) then  
y:= 4\*23; /P11  
x:= y + 2; /P12

**P2::**

If (x > 0) then  
x:= x + 1; /P21

**P3::**

x:= (x\*8) /P31 + x\*2 /P32;

Intercalados de instrucciones atómicas:

P31, P21, P11, P12, P32 = 188

P31, P32, P11, P12, P21 = 95

P11, P12, P31, P21, P32 = 942

2. Suponga ahora, el mismo ejercicio anterior pero las instrucciones son atómicas. Indique cual/es de las siguientes opciones son verdaderas

- a) El valor de x al terminar el programa es 20.
- b) El valor de x al terminar el programa es 94.
- c) Es posible calcular todos los resultados posibles para las variables x e y.

La opción "b" resulta verdadera debido a:

P2, P3, P1 = 94 (valor de resultado mínimo que se puede obtener)

La opción "c" es verdadera debido a que son instrucciones atómicas, con lo cual el intercalado de instrucciones dará como resultado todas los posibles valores de x e y

3. A partir del código multiplicacion.c, pruebe ejecutar el algoritmo con diferente cantidad de hilos (2, 4, y 8) para tamaños de matriz de 32 x 32, y 64 x 64 posiciones. Complete la tabla con los tiempos de ejecución obtenidos. Para compilar:

```
$ gcc -o nombreEjecutable -fopenmp -lm multiplicacion.c
```

Cantidad de hilos/ Tamaño de la matriz	32 X 32	64 X 64
2	0.000988	0.003532
4	0.001268	0.003834
8	0.001620	0.004691

4. Implemente una versión secuencial del algoritmo y realice el cálculo de speedup y eficiencia. Tenga en cuenta que la versión secuencial no debe ser el algoritmo paralelo con un solo hilo dado que la existencia de hilos agrega overhead en el procesamiento y se debe utilizar la mejor versión secuencial para calcular correctamente la performance del algoritmo.

Tiempo de ejecución de versión secuencial del algoritmo: 0.000686

Speedup es cuanto mejora la aplicación paralela respecto de la aplicación secuencial.

Speedup = Tiempo Secuencial / Tiempo Paralelo

Eficiencia representa el porcentaje de tiempo en el cual los procesadores estuvieron útiles (es decir estuvieron siendo utilizados)

Cantidad de hilos/ Tamaño de la matriz	32 X 32	Speedup	Eficiencia
2	0.000988	0.68	0.34
4	0.001268	0.54	0.13
8	0.001620	0.42	0.05

Speedup = 0.000686 / 0.000988 = 0.694332

Speedup = 0.000686 / 0.001268 = 0.541009

Speedup = 0.000686 / 0.001620 = 0.423456

Los valores de eficiencia se encuentran entre 0 y 1. La eficiencia máxima ( $E=1$ ) se obtiene cuando todos los procesadores están siendo completamente utilizados durante todo el periodo de ejecución.

Eficiencia = Speedup / procesadores

Eficiencia =  $0.684332 / 2 = 0.342166$

Eficiencia =  $0.541009 / 4 = 0.135252$

Eficiencia =  $0.423456 / 8 = 0.052932$

Cantidad de hilos/ Tamaño de la matriz	64 X 64	Speedup	Eficiencia
2	0.003532	0.19	0.09
4	0.003834	0.17	0.04
8	0.004691	0.14	0.01

Speedup =  $0.000686 / 0.003532 = 0.194224$

Speedup =  $0.000686 / 0.003834 = 0.178925$

Speedup =  $0.000686 / 0.004691 = 0.146237$

Eficiencia =  $0.194224 / 2 = 0.097112$

Eficiencia =  $0.178925 / 4 = 0.044731$

Eficiencia =  $0.146237 / 8 = 0.018279$