

CI-0117 Laboratorio Semana 5 - Programación Paralela y Concurrente

Jennifer Villalobos Fernández | B67751

1. - Construir una hoja electrónica para realizar los cálculos dada una distribución de tiempos de las tareas.

- Debe entregar la hoja electrónica
- Utilizar cuatro trabajadores
- Utilizar los mismos esquemas de mapeo: Estático por bloque, Estático cíclico y Dinámico
- Utilizar los siguientes datos para los tiempos de las tareas: [12, 60, 70, 20, 40, 60, 40, 13, 80, 60, 13, 11, 30, 12], total = 521
- Calcular el "speedup" y "efficiency"

El documento con todos los requerimientos se encuentra en la carpeta del laboratorio:

Con cuatro trabajadores (procesos o hilos)

Tareas(15)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	Total 521 ⇒ Tiempo serial
Unidades de tiempo por tarea	12	60	70	20	40	60	40	13	80	60	13	11	30	12	

Mapeo de tareas a trabajadores (A, B, C, D)														
A	A	A	A	B	B	B	B	C	C	C	C	D	D	
A	B	C	D	A	B	C	D	A	B	C	D	A	B	Mapeo estatico por bloque (cuatro tareas por trabajador o las que alcance)
A	B	C	D	A	D	B	C	A	D	C	B	A	D	Mapeo estatico ciclico (una tarea para cada trabajador)
														Mapeo dinamico (una a cada trabajador, conforme terminen toman la siguiente)

Tiempos totales por trabajador					Trabajador
A	B	C	D	MAX	
162	153	164	42	164	MapEpB (SpeedUp 521/164 = 3.17, Efficiency 3.17/4 = 0.79)
162	192	96	152	192	MapEc (SpeedUp 521/192 = 2.71, Efficiency 2.71/4 = 0.67)
162	111	96	152	162	MapD (SpeedUp 521/162 = 3.21, Efficiency 3.21/4 = 0.80)

(a) Asignación de tareas a trabajadores.

2. Convertir el programa "PiPorSeriesConMensajes.c" para que utilice su clase "Buzon".

- Entregar otro programa que utilice su clase para paso de mensajes: se realizó el programa con el cambio solicitado. El método "Enviar" de la clase Buzón presentada anteriormente no presenta problemas, sin embargo, el método "Recibir" lanzó un error al cuál no le encontré solución:

```

jennifer@jennifer-VirtualBox: ~/Documentos/C++/PiPorSeriesConMensajes
Archivo Editar Ver Buscar Terminal Ayuda
jennifer@jennifer-VirtualBox:~$ cd ~/Documentos/C++/PiPorSeriesConMensajes
jennifer@jennifer-VirtualBox:~/Documentos/C++/PiPorSeriesConMensajes$ make
g++ -c -o Buzon.o Buzon.cpp
g++ -c -g Buzon.cpp PiPorSeriesConMensajes.cpp
g++ -g Buzon.o PiPorSeriesConMensajes.o -o pi
jennifer@jennifer-VirtualBox:~/Documentos/C++/PiPorSeriesConMensajes$ ./pi
Creando el proceso 8040: termino inicial 0, termino final 100000
Creando el proceso 8041: termino inicial 100000, termino final 200000
Creando el proceso 8042: termino inicial 200000, termino final 300000
Creando el proceso 8043: termino inicial 300000, termino final 400000
Creando el proceso 8044: termino inicial 400000, termino final 500000
Creando el proceso 8045: termino inicial 500000, termino final 600000
Creando el proceso 8046: termino inicial 600000, termino final 700000
Creando el proceso 8047: termino inicial 700000, termino final 800000
Creando el proceso 8051: termino inicial 800000, termino final 900000
Creando el proceso 8052: termino inicial 900000, termino final 1000000
Buzon::Recibir: Argument list too long
jennifer@jennifer-VirtualBox:~/Documentos/C++/PiPorSeriesConMensajes$

```

(b) Ejecución Pi con Buzon.

- Entregar un Makefile: se encuentra en la carpeta respectiva del programa.

4. Revisar los programas "SumaUno.c" y "SumaUnoNoRaceCondition.c".

- Analizar la manera de medir el tiempo en los programas: tanto en el programa "SumaUno.c" como en "SumaUnoNoRaceCondition.c" se obtiene un tiempo mayor en la versión serial, como era de esperarse. En esta parte del código se mide el tiempo de manera correcta, ya que toma todos los segundos que dura en ejecutarse el método SerialTest, el cual toma de uno en uno todos los procesos disponibles (100). En el método ForkTestRaceCondition el tiempo es menor, debido a los múltiples procesos realizando la tarea, sin embargo el tiempo es "erróneo", ya que no calcula lo que está durando cada proceso hijo por aparte al realizar la suma. Con el método ForkTestNORaceCondition en el programa "SumaUnoNoRaceCondition.c", se controla el paso de los procesos durante la suma, por lo que se calcula el tiempo correctamente. Por esta razón se dura un poco más que el método anterior, pero no tanto como la versión serial.

```

jennifer@jennifer-VirtualBox: ~/Documentos/C++/Sumas
Archivo Editar Ver Buscar Terminal Ayuda
jennifer@jennifer-VirtualBox:~$ cd ~/Documentos/C++/Sumas
jennifer@jennifer-VirtualBox:~/Documentos/C++/Sumas$ g++ SumaUnoNoRaceCondition
.cpp -o sumaNoRaceCondition
jennifer@jennifer-VirtualBox:~/Documentos/C++/Sumas$ ./sumaNoRaceCondition
Serial version:      Valor acumulado es 100000 con 100 procesos in 0.000631 s
econds
Fork, Race Cond.:   Valor acumulado es 100000 con 100 procesos in 0.005964 s
econds
Fork, No Race Cond.: Valor acumulado es 100000 con 100 procesos in 0.005982 s
econds
jennifer@jennifer-VirtualBox:~/Documentos/C++/Sumas$

```

(c) Ejecución SumaUnoNoRaceCondition.

- Comprobar que "SumaUno.c" tiene problemas de condición de carrera: el programa presenta una secuencia de salida impredecible, lo que confirma la condición de carrera. Se modificó el código del programa para añadir unas impresiones en consola, esto con el propósito de demostrar la salida del mismo:

