



## TP4: Comunicación y Sincronización en Multicomputadoras

**Fecha de entrega:** lunes 8 de octubre de 2007

**Observaciones:** La resolución del Proyecto es individual. La solución de los ejercicios será discutida en clase.

**Bibliografía:** [Lester93] Bruce P. Lester. *The Art of Parallel Programming*. Prentice Hall, 1993. Capítulos 7, 8 y 9.

1. Resuma los diferentes métodos de “broadcasting and aggregation” con diferentes topologías (7.3)
2. Implemente en Multi-Pascal el algoritmo “Insertion Sort” (8.3). Ejecute y compare con diferentes arquitecturas. Utilice la directiva “DELAY” (ver apéndice B.10.2) para aumentar el retardo en los mensajes.
3. Modifique la implementación del productor–consumidor del proyecto anterior para trabajar sobre arquitecturas multicomputadoras. Realice una versión con un canal único para cada consumidor. Los productores producen ítems de datos y los van repartiendo entre los canales. El programa debe preguntar por la cantidad de productores y consumidores. Compare los resultados que arrojan las diferentes arquitecturas.
4. Dada una matriz de  $n \times n$ , A, se debe construir una matriz resultado C tal que:

$$C[i, j] = \sum_{k=1}^n A[k, j] + \sum_{k=1}^{j-1} A[i, k] + C[i, j-1]$$

Por ejemplo, para  $C[1,1] = (4+6+5+1) = 16$ ;  $C[1,2] = (2+3+9+0)+4+16 = 34$

4	2	0	8
6	3	2	1
5	9	6	5
1	0	4	7

⇒

16	34		
		57	
		44	

A
C

Para una **arquitectura con topología LINE** escriba un programa en Multi-Pascal que, utilizando un algoritmo con paralelismo, calcule la matriz C a partir de A. En base a su solución, indique en función de la dimensión de la matriz (n) cuál es el número mínimo de procesadores que necesita.

**Cómo entregar el proyecto:** La entrega debe realizarse por email antes del horario de práctica, o en un disquete durante el horario de práctica. El formato será un archivo **ApellidoTPnro.ZIP** el cual contenga un directorio **TPnro** con los archivos a entregar. Ejemplo: la entrega del proyecto 3 del alumno García consiste del archivo GarciaTP3.zip que contiene al directorio TP3



5. Implemente el algoritmo de Jacobi en multicomputers (Sección 9.4) para una matriz de 7x7 y una cantidad fija de iteraciones. Por ejemplo, para la matriz A, después de 10 iteraciones se obtiene la matriz B (verifique el funcionamiento de su programa con estos valores):

**A**

1.00	1.00	1.00	1.00	1.00	1.00	1.00
1.00	0.00	0.00	0.00	0.00	0.00	1.00
1.00	0.00	0.00	0.00	0.00	0.00	1.00
1.00	0.00	0.00	0.00	0.00	0.00	1.00
1.00	0.00	0.00	0.00	0.00	0.00	1.00
1.00	0.00	0.00	0.00	0.00	0.00	1.00
1.00	1.00	1.00	1.00	1.00	1.00	1.00

**B**

1.00	1.00	1.00	1.00	1.00	1.00	1.00
1.00	0.91	0.84	0.82	0.84	0.91	1.00
1.00	0.84	0.72	0.68	0.72	0.84	1.00
1.00	0.82	0.68	0.63	0.68	0.82	1.00
1.00	0.84	0.72	0.68	0.72	0.84	1.00
1.00	0.91	0.84	0.82	0.84	0.91	1.00
1.00	1.00	1.00	1.00	1.00	1.00	1.00

→ **Proyecto 4:** Entregue los ejercicios 2 y 3 del TP4.

**Cómo entregar el proyecto:** La entrega debe realizarse por email antes del horario de práctica, o en un disquete durante el horario de práctica. El formato será un archivo **ApellidoTPnro.ZIP** el cual contenga un directorio **TPnro** con los archivos a entregar. Ejemplo: la entrega del proyecto 3 del alumno García consiste del archivo GarciaTP3.zip que contiene al directorio TP3