

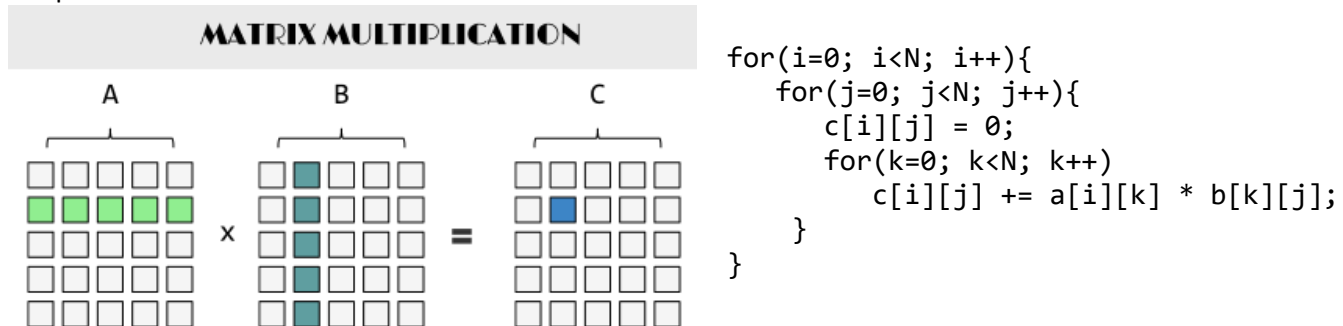
Taller II – Snail Matrix

Objetivo

Solucionar problemas computacionales que impliquen comunicación entre procesos mediante segmentos de memoria compartida (System V).

Descripción

La figura y el código siguiente ilustran el procedimiento aplicado para estimar una posición del resultado de la multiplicación de dos matrices:



Debido a la disponibilidad de hardware de múltiples núcleos y de un sistema operativo con planificación multitarea, se solicita implementar el procedimiento de multiplicación de matrices **mediante múltiples procesos**. Sin embargo, por requerimiento especial, se desea que cada proceso **hijo** procese un sector especial de la matriz. Este sector **es un anillo** que incluye dos filas y dos columnas, y son definidas por el identificador de cada proceso hijo. El esquema siguiente muestra un ejemplo del cómputo distribuido para cada proceso hijo.

Esquema de hilos para la operación con la Matriz

5	3	12	9	0	11
4	18	50	31	62	7
45	25	3	14	32	16
1	9	6	48	37	1
2	28	22	7	25	4
41	33	66	2	8	3

Hijo 1
Hijo 2
Hijo 3
Padre (imprimir)

El esquema deberá garantizar que el resultado final sea impreso por el proceso **Padre** al finalizar el computo realizado por cada uno de los procesos hijos.

Para que una matriz esté en un segmento de memoria compartida considere las siguientes funciones:

```
unsigned int sizeof_dm(int rows, int col, size_t sizeElement){
    size_t size = rows * sizeof(void *); //indexSize
    size      += (col * rows * sizeElement); //rows - Size
    return size;
}

void create_index(void **m, int rows, int cols, size_t sizeElement){
    int i;
    size_t sizeRow = cols * sizeElement;
    m[0] = m+rows;
    for(i=1; i<rows; i++){
        m[i] = (m[i-1]+sizeRow);
    }
}
```

De tal forma que el siguiente conjunto de instrucciones permitiría direccionar un vector 2-dimendional en un conjunto continuo de memoria que resida en un segmento de memoria compartida. El valor *sizeElement* deberá ser el tamaño del tipo de dato del que será la matriz.

```
size_t sizeMatrix = sizeof_dm(Rows, Cols, sizeof(double));
shmId = shmget(IPC_PRIVATE, sizeMatrix, IPC_CREAT|0600);
matrix = shmat(shmId, NULL, 0);
create_index((void*)matrix, Rows, Cols, sizeof(double));
```

Requerimientos

- La matriz debe estar en un segmento de **memoria compartida**
- Los valores iniciales de las matrices pueden ser cargadas a su criterio.
- Los tamaños de las matrices deberán ser fácilmente modificados para probar diferentes escenarios de su solución
- Los procesos hijos creados inicialmente deberán ser los mismos durante toda la ejecución.
- La ejecución de los procesos hijos deberá ser paralela y simultánea.
- Se debe imprimir el identificador de cada proceso hijo y sus posiciones de cálculo

Esquema de evaluación

- Creación generalizada de jerarquía de procesos - 10%
- Creación generalizada de los segmentos de memoria compartida – 40%
- Comunicación correcta entre los procesos – 50%

Instrucciones

El código generado debe ser enviado por correo electrónico a sanchez.gt@gmail.com como archivo adjunto, con asunto Lab04 – shared memory, en el cuerpo del correo especificar nombre, código y grupo, inmediatamente después de ser revisado.