Práctica 2

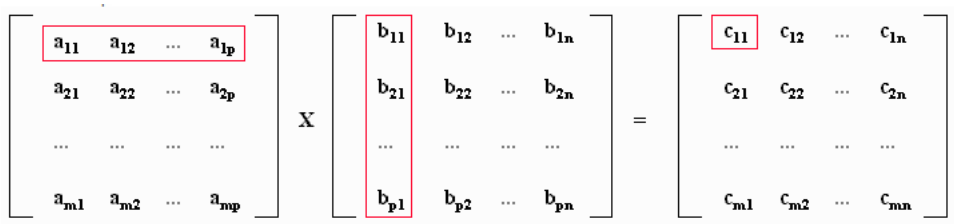
Oscar Daniel Juárez Cruz

April 13, 2018

1 Introducción

Los procesos nos pueden ayudar a repartir varias tareas entre un número asinado de procesos. En esta ocasión usaremos procesos para multiplicar dos matrices de manera secuencial.
La multiplicación de matrices debe cumplir la siguiente condición
$A_{axm} \cdot B_{mxb} = M_{axb}$ *el número de columnas de la matriz A es el número de filas de la matriz B y la matriz resultante M es de dimensiones axb*

Multiplicación de matrices



$$c_{11} = a_{11}b_{11} + a_{12}b_{21} + \ldots + a_{1p}b_{p1}$$

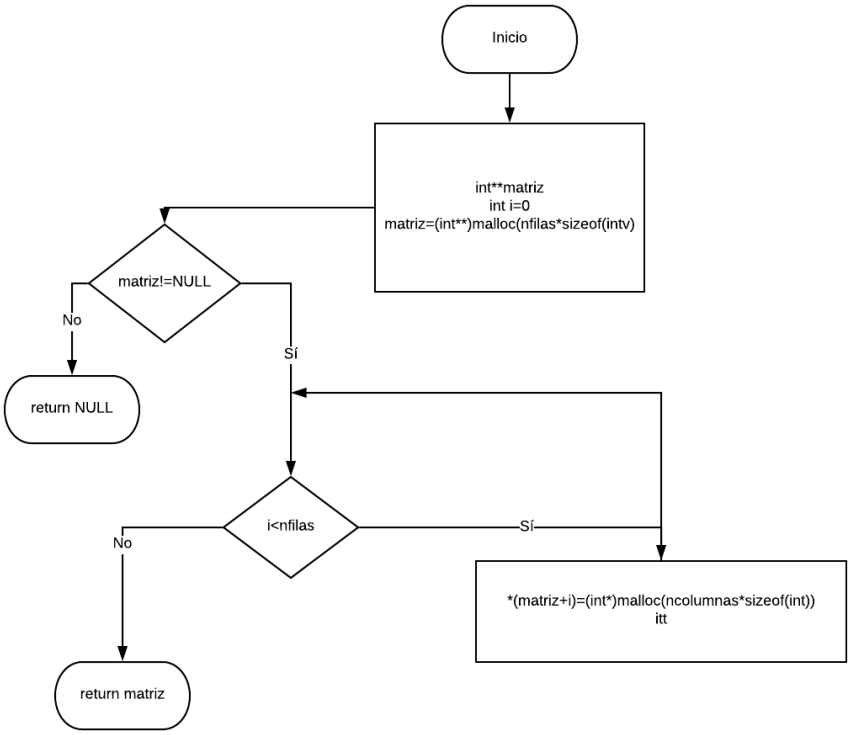
2 Desarrollo

Reutilizamos parte del código anterior pero ahora en vez de que los hijos crearan más hijos iban a multiplicar dos matrices de manera distribuida, quiere decir que a cada proceso le corresponde multiplicar cierto número de filas e imprimirlas

Para resolver el problema lo primero que debemos hacer es crear la matriz.

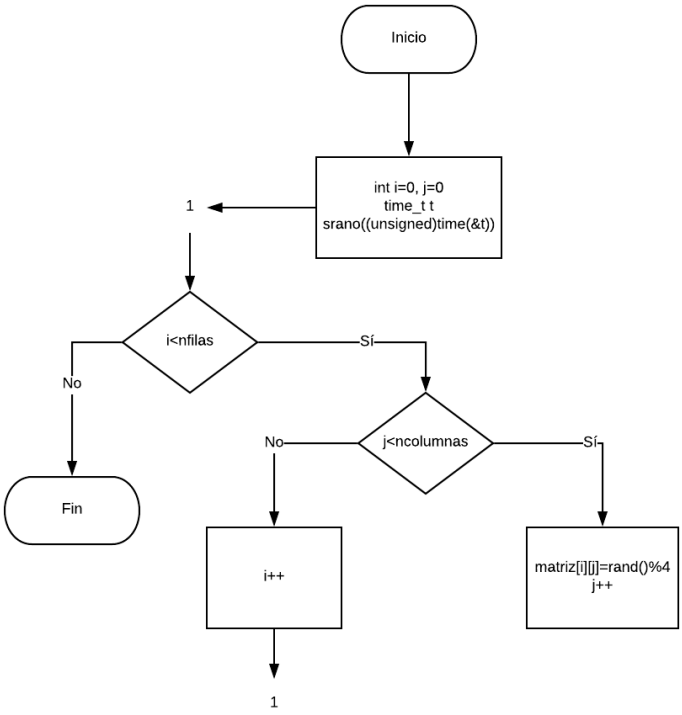
Las matrices se crearon con memoria dinámica a traveś de la función crearMatriz(...).

int **crearMatriz(int nfilas, int ncolumnas)



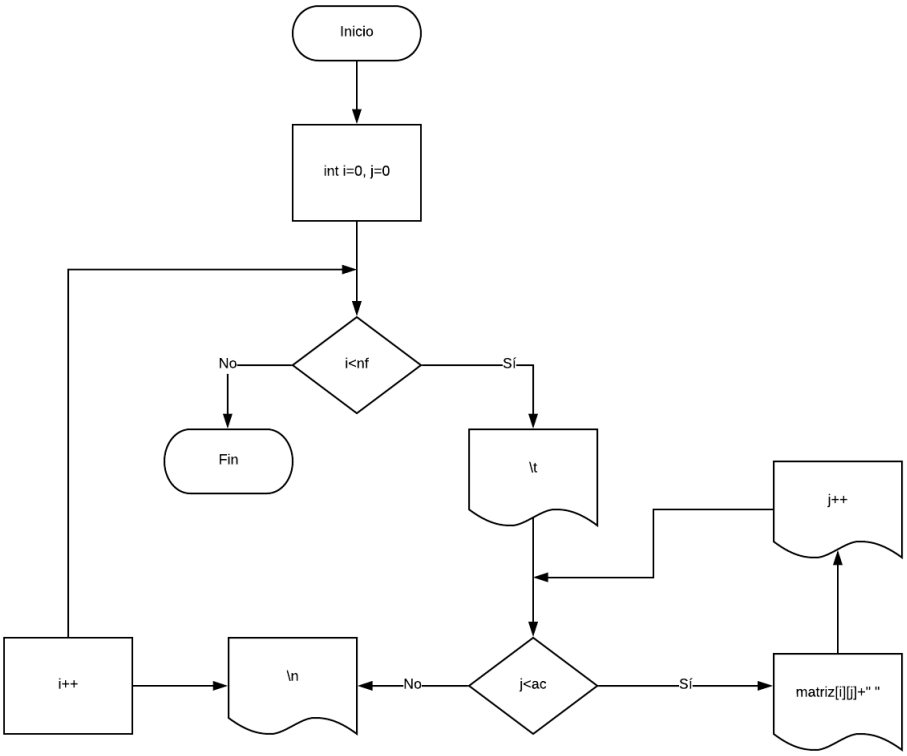
Después llenamos las matrices con la función llenarMatriz(...)

void llenarMatriz (int **matriz, int nfilas, int ncolumnas)



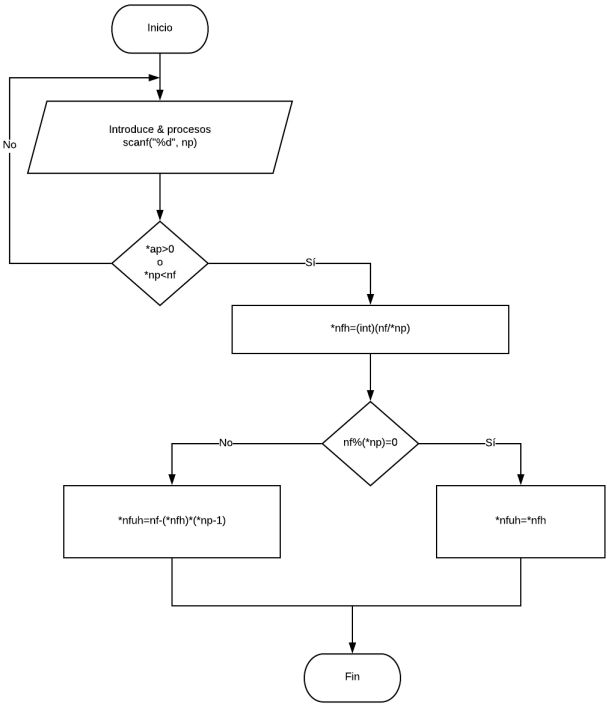
Después mostramos las matrices creadas con la función imprimirMatriz(...)

void imprimirMatriz (int **m, int nf, int nc)



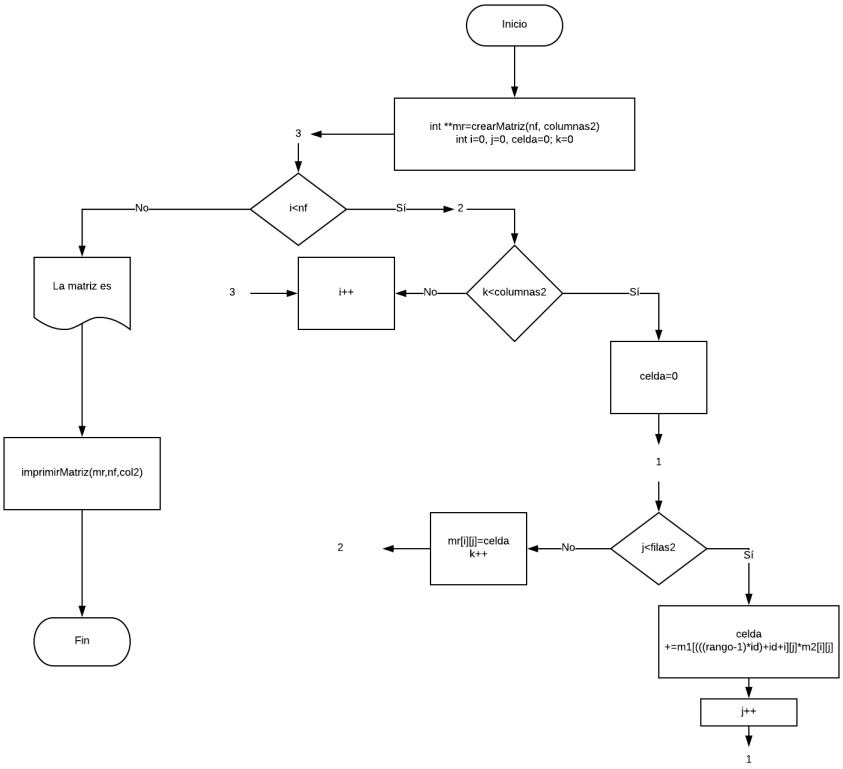
Elegimos el número de procesos que necesitamos y usamos la función dividirProcesos(...) para saber cuantas filas corresponden a cada hijo

void dividirProcesos(int *np, int *nfh, int *nfuh, int nf)



Ahora creamos un for para poder iterar con los hijos y cada uno tiene la responsabilidad de multiplicar cierto número de filas con la función multiplicarMatrices(...)

void multiplicarMatrices (int **m1, int **m2, int rango, int id, int nf, int filas2, int columnas2)



Cada que se multiplica se va imprimiendo el número de proceso que la está haciendo y las filas correspondientes como se puede ver en el siguiente output

```
afront@afront-HP-Notebook:~/Documentos/SO/C/Practica2/Código$ ./p
Ingresa el numero de filas de la primer matriz:4
Ingresa el numero de columnas de la primer matriz:4
Ingresa el numero de filas de la segunda matriz:4
Ingresa el numero de columnas de la segunda matriz:4
Las matrices generadas son:
Matriz 1:
         3 1 3 2
         3 0 3 1
         1 1 0 1
         2 1 3 0

Matriz 2:
         3 1 3 2
         3 0 3 1
         1 1 0 1
         2 1 3 0

Introduce el numero de procesos:
3
La matriz resultante del proceso 1 es:
         19 8 18 10

La matriz resultante del proceso 2 es:
         14 7 12 9

La matriz resultante del proceso 3 es:
         8 2 9 3
         12 5 9 8

afront@afront-HP-Notebook:~/Documentos/SO/C/Practica2/Código$
```

Solución:

$$\mathbf{C} = \mathbf{A} \cdot \mathbf{B} = \begin{pmatrix} 3 & 1 & 3 & 2 \\ 3 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 2 & 1 & 3 & 0 \end{pmatrix} \cdot \begin{pmatrix} 3 & 1 & 3 & 2 \\ 3 & 0 & 3 & 1 \\ 1 & 1 & 0 & 1 \\ 2 & 1 & 3 & 0 \end{pmatrix} =$$

$$= \begin{pmatrix} 19 & 8 & 18 & 10 \\ 12 & 5 & 12 & 7 \\ 8 & 2 & 9 & 3 \\ 12 & 5 & 9 & 8 \end{pmatrix}$$

3 Conclusión

Los procesos nos pueden ayudar a optimizar una tarea repartiendola en varias, con lo que podemos generar un control sobre lo procesos cuando estos se ejecutan.

Lo más complicado de la práctica fue hacer que la multiplicación se repartiera en diferentes procesos, pero al copiarse la información del padre a lo hijos ellos tenían el acceso a la matriz, pero aún así fue complicado generar la multiplicación y que cada hijo aportara una fila diferente