Bogotá, 18 de agosto de 2024

Juan Diego Muñoz Angulo

Sistemas Operativos

Documentación: Gestión de Memoria y Multiplicación de Matrices

Este código multiplica dos matrices con valores predefinidos, pero con un tamaño proporcional al que desee proveer el usuario.

1. **imprimir**: Imprime los elementos de una matriz.
2. **insertar**: Rellena la matriz con un valor x.
3. **multiplicar**: Realiza la multiplicación de dos matrices cuadradas de tamaño n x n. Se realiza una multiplicación matricial estándar y el resultado se guarda en una nueva matriz C, cuya memoria se asigna de forma dinámica, y luego se libera al salir del programa.

El programa está diseñado para ser ejecutado desde la línea de comandos Shell o de Linux, con un argumento que representa el número de elementos en la matriz, ejemplo, ***“.\'matrizFuncionalFinal.exe' 16”***. Si no se proporciona el argumento, el programa señala el error que sale al inicio de la función main, que faltan o sobran argumentos. Puede ser compilado perfectamente con GCC.

```c
/*****************************************
*                                        *
* Autor: JD                              *
* Materia: Sistemas Operativos           *
* Fecha: 18-08-24                        *
* Tema: Gestion de Memoria               *
*  Entrega Tarea Gestion de Memoria      *
*                                        *
*****************************************/

#include <stdio.h>  //Bibliotecas
#include <stdlib.h>
#include <math.h>

// Función para imprimir la matriz.
void imprimir(int size, int *matriz) {
    int n = (int)sqrt(size); // Se calcula la dimensión de la matriz cuadrada.
    for(int i = 0; i < size; i++) { .
        if(i % n == 0) { // Si el indice es múltiplo de n, se imprime una nueva línea. Esto para
separar corretamente la matriz.
            printf("\n");
        }
        printf("%d ", matriz[i]); //Imprime.
    }
    printf("\n");
}

// Función para llenar la matriz con un valor específico.
void insertar(int size, int dato, int *matriz) {
    for(int i = 0; i < size; i++) {
        matriz[i] = dato;
    }
}

// Función para multiplicar dos matrices cuadradas.
int* multiplicar(int size, int *mA, int *mB) {
    int n = (int)sqrt(size);
    int *mC = (int*)malloc(size * sizeof(int)); //Reservamos memoria para la matriz que nos va a
quedar..

    for(int i = 0; i < n; i++) { //Pasamos sobre las filas de la primera matriz mA
        for(int j = 0; j < n; j++) { //Pasamos por las columnas de la segunda amtriz mB
            mC[i*n + j] = 0; //Ponemos el valor a cambiar en 0 para evitar problemas.
            for(int k = 0; k < n; k++) {
                mC[i*n + j] += mA[i*n + k] * mB[k*n + j]; //Realizamos la operaciones y añadimos al
resultado
            }
        }
    }
    return mC; //Devuelve la matriz que contiene el resultado
}

int main(int argc, char *argv[]) {
    if(argc != 2) { // Verifica si se ha pasado el número correcto de argumentos en consola.
        printf("Error de argumentos.\n");
        printf("Debe tener 1 argumento, el num de espacios en la Matriz.\n");
        return 0;
    }

    int size = atoi(argv[1]);
    int mA[size];
    int mB[size];

    insertar(size, 1, mA);
    insertar(size, 2, mB); //llenamos con dos valores predefinidos

    printf("Matriz A:");
    imprimir(size, mA);
    printf("Matriz B:");
    imprimir(size, mB);

    int *mC = multiplicar(size, mA, mB);

    printf("Matriz C Resultado:");
    imprimir(size, mC);

    free(mC); //Libera la memoria asignada a la matriz resultante en malloc

    printf("\nEso es todo amigos.\n");
    return 0;
}
```