

Reporte Final: Práctica 2 de Organización de computadores
Pérez Pérez, Juan David.
jperezp2@eafit.edu.co
Universidad EAFIT

Resumen Este documento pretende dar a conocer todos los aspectos involucrados en la práctica número 2 presentada en la asignatura: Organización de Computadores.

Índice de Términos

Matriz, posiciones de memoria, contenido, contador, operaciones, C++.

I. INTRODUCCIÓN

En este informe muestra todo el planteamiento e instrucciones que se desarrollaron para el debido funcionamiento de este proyecto, con sus respectivas características funcionales y no funcionales.

II. MARCO TEÓRICO

A. Fase de revisión

En la primera parte del desarrollo de la práctica se comenzó por comprender el funcionamiento de esta, para poder llevar a cabo la debida lógica para desarrollar el proyecto.

Esto teniendo en cuenta que el entorno o lenguaje en el cual se hace este proyecto a pesar de ser conocido, en las pautas del proyecto se pide hacer funciones que manejan lenguaje ensamblador; a pesar de haber sido manejado previamente, se presenta un cambio de lógica para determinados procesos.

Por lo cual se requiere una búsqueda exhaustiva sobre la sintaxis que este maneja.

B. Fase de desarrollo

Se da continuación a la lógica propuesta de cómo realizar las acciones en las matrices 4x4, en esta lógica están propuestas varias ideas; una de ellas es inicializar dos matrices con 16 posiciones en las

cuales se pide al usuario que ingrese los coeficientes que desea en cada matriz, especificando previamente la fila y columna en la cual se asigna el valor.

Además de esto se requieren 4 métodos que procesan la operación que desee ejecutar el usuario, ya sea: suma, resta, multiplicación o división.

Esto se hace con el fin de tener todos los valores almacenados para poder realizar las debidas operaciones.

Antes de realizar esta serie operaciones se valida previamente que sea una de las opciones disponibles, también se tiene un proceso por cada acción las cuales son:

- (1) Sumar matrices
- (2) Restar matrices
- (3) Multiplicar matrices
- (4) División de matrices

Para todo esto se establecen una serie de condicionales que especifican que operación se realiza según la opción seleccionada, en cada condicional se determina qué valor se va a operar de la matriz a y b, todo ello se empieza con un contador i (filas) y un contador j (columnas).

C. Fase Final

Se realizan cambios sobre el cómo deben cambiar los contadores o variables de la fila y columna de la matriz (i,j) para recorrer la matriz y obtener sus respectivos valores, también se hace algunas pequeñas modificaciones respecto a los valores que se reciben en el coprocesador matemático.

Además de un análisis o prueba de escritorio previa para verificar el correcto funcionamiento del programa y una mejora en cuanto a las líneas de código usadas, además de comentarlo para comprender de mejor manera que realiza cada parte del proyecto.

III. INDICACIONES ÚTILES

A. Código de algunas funciones

Método de suma

```
float pSuma(float numA,float numB){
    float result;
    __asm {
        FLD DWORD PTR numA // cargo
        el primer valor a analizar
        FLD DWORD PTR numB // cargo
        el segundo valor a analizar
        FADD //realizo la operación
        indicada
        FSTP DWORD PTR result
        // almaceno y extraigo el resultado
        FSTP ST(0) //limpio el stack
    }
    return result;
}
```

Llenado de matrices

```
// Matriz A
cout << "Valores Matriz A" << endl;
for (int i = 0; i < 4;i++) {
    for (int j = 0; j < 4;j++) {
        cout << "Ingresa el valor
en la fila: " << i + 1 << " columna: " << j + 1 <<
endl;

        cin >> entrada;
        matrizA[i][j] = entrada;
    }
}

//Matriz B
cout << "Valores Matriz B" << endl;
for (int i = 0; i < 4;i++) {
    for (int j = 0; j < 4;j++) {
        cout << "Ingresa el valor
en la fila: " << i + 1 << " columna: " << j + 1 <<
endl;

        cin >> entrada;
        matrizB[i][j] = entrada;
    }
}
```

B. Salida del programa

En el programa muestra como salida la matriz resultante de la operación seleccionada entre las matrices que digitó el usuario, con sus respectivos valores en cada casilla indicando el orden en el que visitó cada posición, esto se muestra de forma matricial.

A_{ij} i: fila j: columna

```
A11 A12 A13 A14
A21 A22 A23 A24
A31 A32 A33 A34
A41 A42 A43 A44
```

IV. DIFICULTADES

Algunas dificultades presentadas en el desarrollo del proyecto fue el obtener el valor directamente de la posición i,j de la matriz mediante instrucciones del coprocesador matemático como el FLD, también el detectar errores en tiempo de ejecución del programa lo cual se resolvió posteriormente con herramientas proporcionadas por el IDE *visual studio*.

V. CONCLUSIONES

Como resultado de toda la investigación, planteamiento y desarrollo realizado en este documento, se puede ver el cómo se llevó a cabo el proyecto realizado con sus respectivas métricas las cuales requieren un uso del coprocesador matemático, conocer más a fondo una nueva herramienta que posee la unidad central de procesamiento.

Todo esto por medio del IDE **Visual Studio** mediante lenguajes como **C++** y *assembler*.

Por otro lado se ve la lógica implementada para este proyecto aprovechando la programación mediante posiciones de memoria para llevar a cabo el recorrido de la matriz.

REFERENCIAS

- [1] Sitio web ***visual studio***
“<https://www.visualstudio.com/en-us/downloads/download-visual-studio-vs.aspx>”.
- [2] Intel x86 ***instruction set reference***
“<http://www.felixcloutier.com/x86/>”.
- [3] University of Virginia
“<http://www.cs.virginia.edu/~evans/cs216/guides/x86.html>”.
- [4] Microsoft page “<https://msdn.microsoft.com/es-co/library/ms235629.aspx>”.