ST0254 – Organización de computadores
Práctica 1: Programación en Assembler

I.S. José Luis Montoya Pareja
Profesor Universidad EAFIT
Medellín, Colombia, Suramérica
email jmontoya@eafit.edu.co

Resumen

Esta práctica busca aumentar la comprensión de los estudiantes sobre el funcionamiento de un procesador mediante la programación de bajo nivel; son las instrucciones de máquina el punto de entrada para llegar a entender cómo funciona el procesador.

Palabras Clave
Ensamblador, assembler, Intel, compilador, programa, instrucciones.

INTRODUCCIÓN

"The Greek philosopher Zeno considered the problem of summing an infinite series to achieve a finite result, but rejected it as an impossibility: the result was Zeno's paradox. Later, Aristotle proposed a philosophical resolution of the paradox, but the mathematical content was apparently unresolved until taken up by Democritus and then Archimedes. It was through Archimedes's method of exhaustion that an infinite number of progressive subdivisions could be performed to achieve a finite result. Liu Hui independently employed a similar method a few centuries later.

In the 14th century, the earliest examples of the use of Taylor series and closely-related methods were given by Madhava of Sangamagrama. Though no record of his work survives, writings of later Indian mathematicians suggest that he found a number of special cases of the Taylor series, including those for the trigonometric functions of sine, cosine, tangent, and arctangent. The Kerala school of astronomy and mathematics further expanded his works with various series expansions and rational approximations until the 16th century.

In the 17th century, James Gregory also worked in this area and published several Maclaurin series. It was not until 1715 however that a general method for constructing these series for all functions for which they exist was finally provided by Brook Taylor, after whom the series are now named.

The Maclaurin series was named after Colin Maclaurin, a professor in Edinburgh, who published the special case of the Taylor result in the 18th century." [1]

Funciones trigonométricas

Como todos sabemos, la serie de Maclaurin para el cálculo del Seno y el Coseno nos permite obtener un resultado mediante una sumatoria infinita de términos que al final nos deja el resultado de la función trigonométrica.

$$\sin x = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1}$$

$$\cos x = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} x^{2n}$$

El coprocesador matemático de Intel tiene implícitas las instrucciones fsin y fcos para realizar dichos cálculos, sin embargo lo que se quiere con la práctica es implementar las series en assembler y que esto pueda ser usado por cualquier programa.

OBJETIVOS

1. Objetivo General

Realizar el proceso de construcción de una DLL en lenguaje ensamblador donde se encuentre implementado las funciónes trigonométricas y que sea capaz de ser invocada por un programa escrito en Visual Basic 6.

2. Objetivos Específicos

a. Realizar el programa en lenguaje ensamblador que realice el cálculo del Seno mediante series.
b. Realizar el programa en lenguaje ensamblador que realice el cálculo del Coseno mediante series.
c. Implementar la función de la tangente mediante el cálculo del seno y el coseno.
d. Crear una DLL en assembler con dicho cálculo

ESPECIFICACIONES DE LA DLL

La DLL deberá poseer tres funciones públicas: Seno, Coseno y Tangente. Se adjuntan los prototipos.

1. single seno(int numAngulo)
2. single coseno(int numAngulo)
3. single tangente(int numAngulo)

Todas las funciones tienen como parámetro el número entero del ángulo que se va a calcular. El ángulo puede ser entre -360 y 360 grados sin decimales. El ángulo deberá estar en grados.

Visual Basic soporta el tipo de datos single para flotantes de 32 bits, lo cual significa que se utilizarán números de punto flotante de 32 bits.

La DLL deberá tener el código en ASM para invocar las 3 funciones y debe estar en capacidad de retornar los resultados con el método estándar definido de retorno de datos de tipo flotante.

ANOTACIONES GENERALES DE ENTREGA E HITOS A CUMPLIR

Anotaciones generales de la práctica

1. La práctica se puede hacer individual o en grupos de hasta máximo tres personas. Los nombres de las personas que conforman cada grupo deben ser informados al profesor antes del martes 28 de julio.
2. La totalidad de la práctica se realizará en fasm [2].
3. La práctica se hará en Windows.
4. Los informes finales de la práctica se entregarán en un archivo siguiendo el formato de la IEEE para la publicación de artículos, el cual se puede descargar desde EAFIT Interactiva.
5. La entrega se realizará por EAFIT Interactiva (no se admiten por correo electrónico) adjuntando dos archivos:
    a. El archivo con extensión .dll generado por cada grupo
    b. Los fuentes de assembler de dicho archivo.
    c. Un archivo tipo documento Word o PDF con los siguientes elementos:
        i. Los detalles de diseño encontrados durante la construcción del módulo.
        ii. Las especificaciones del módulo entregado, es decir: los criterios de diseño del trabajo.
        iii. Dificultades encontradas durante el diseño de la práctica
        iv. Opinión personal sobre la práctica.
6. Sustentación: El profesor tendrá instalado Visual Basic 6 con una aplicación que invocará la DLL de cada grupo y compilará y ejecutará la aplicación.
7. Los avances se empezarán a enviar a partir del día 31 de Julio.
8. Criterio de calificación:
    a. 0.0: No se entrega nada
    b. 1.0: Se entrega una de las funciones trabajando en Assembler
    c. 1.5: Se entrega dos de las funciones trabajando en Assembler
    d. 2.0: Las tres funciones se entregan funcionando pero no hay DLL
    e. 3.0: Las tres funciones se entregan funcionando en la DLL
    f. 4.0: Se puede compilar la DLL con VB pero no funciona el programa
    g. 5.0: El programa funciona correctamente.
9. Nota: Se podría utilizar fasm para Linux, caso en el cual el grupo deberá entregar la aplicación en Linux adicional al trabajo sobre la librería. En este caso si todo funciona y la aplicación se verifica se tendrá una bonificación de 1.0 en la nota mas mala durante el semestre.

REFERENCIAS

http://en.wikipedia.org/wiki/Taylor_series#History visitado Sábado 18 de julio de 2009

http://flatassembler.net/ visitado el sábado 18 de julio de 2009