

Arquitectura de computadores

Laboratorio 2

Benjamín Moya Monsalve
Departamento de Ingeniería Informática
Universidad de Santiago de Chile, Santiago, Chile
 benjamin.moya.m@usach.cl

Resumen—En cursos pasados de matemática aplicada, se nos expusieron las distintas metodologías que en antaño se utilizaban para describir fenómenos y cuerpos complejos, como los son las aclamadas funciones geométricas y trigonométricas, vista en más de una ocasión y en cientos de aplicaciones, donde una y otra vez estas se ven de una u otra forma envueltas, por lo que su importancia radica mucho más allá de lo complejo que puede ser describir su comportamiento en ciertas situaciones, si no en como estas son un aporte al entendimiento de aquello que aún no conocemos o que está aún en modelación.

Durante el desarrollo de la actividad ayudaremos a Benjamín a descubrir los secretos detrás de un misterioso oráculo que dice revelar el futuro, apoyando mediante la aplicación de funciones matemáticas conocidas y los conceptos revisados en cátedra acerca del lenguaje ensamblador MIPS, para la familiarización dentro del entorno del trabajo con registros CPU y las funciones primitivas de la computación, con tal de obtener nuestra ansiada visión hacia el futuro.

Palabras claves—Función, Geométrica, Trigonométrica, MIPS y Registros CPU.

I. INTRODUCCIÓN

A lo largo de la vida de la humanidad, se han creado distintos métodos de medición de distancia, dependiendo de los datos que se nos son entregados, uno de los más reconocidos son las fórmulas euclidianas, donde gracias a la diferencia de puntos, y el módulo de estos, podemos comparar vectores con tal de describir cuan cerca o lejos nos ubicamos de determinada posición. Así mismo, las funciones trigonométricas nos han apoyado en una mirada mucho más compleja, ya que son partícipes de miles de fórmulas que nos ayudan a describir el comportamiento de todo lo que nos rodea.

Hoy Benjamín necesita de nuestra ayuda, con mediciones relacionadas con estos conceptos, que de ser logrados se nos concederá la facultad de ver nuestro futuro gracias a un oráculo descubierto por el mismo, en la biblioteca de la universidad. Por lo que, nuestra misión será programar en MIPS un código que mediante subrutinas, consigamos recrear las funciones seno y coseno, según la representación que Brook Taylor y otros, nos han entregado de ellas con tal de poder aproximar su resultado al más fidedigno y superar la primera etapa del laberinto matemático que se le fue propuesto para alcanzar el dichoso premio, donde, por

último, se nos entregaran dos vectores en 3 coordenadas dispuestas a representar una ubicación en el mismo sistema, el cual, gracias a lo descrito hace miles de años por Euclides, dispondremos de un segundo código para operar estos dos elementos con la distancia euclidiana, previamente descrita, para así entablar un mejor entendimiento de los procesos con los registros CPU y descubrir la coordenada de la ubicación de este anhelado oráculo. Enunciado completo.

II. ANTECEDENTES

II-A. Análisis matemático

Ciertamente, uno de los apartados más importantes de frente al desarrollo de una solución mediante la implementación en el lenguaje ensamblador MIPS, es la representación que le atribuiremos a las distintas fórmulas matemáticas que envuelven el enunciado. Así mismo, en un principio se nos describe tanto la función trigonométrica del seno, como también la del coseno, en su modelo tipo serie de Taylor, donde mediante la expansión de términos cada vez más grandes, podemos aproximar ciertas funciones gracias a la suma infinita de ellos. Por lo mismo, debido a que una suma infinita puede representar un alto coste computacional, solo nos encargaremos de obtener una aproximación con los primeros 4 términos para el seno y los primeros 5 términos para el coseno, entregándonos así las siguientes fórmulas:

$$\text{sen}(x) \approx x - \frac{x^3}{3!} - \frac{x^5}{5!} - \frac{x^7}{7!} \quad (1)$$

$$\text{cos}(x) \approx 1 - \frac{x^2}{2!} - \frac{x^4}{4!} - \frac{x^6}{6!} - \frac{x^8}{8!} \quad (2)$$

Por otro lado, se nos dice que se nos entregaran 2 vectores de 3 componentes cada uno, tal que estos serán de la forma: $\{(x, y, z) | x, y, z \in \mathbb{R}\}$. Indicando según lo descrito anteriormente, al momento de usar la fórmula de distancia euclidiana para la obtención de la coordenada, esta será de 3 parámetros, los cuales se caracterizan por ser la diferencia de cada componente de los vectores, obteniendo finalmente la siguiente fórmula:

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \quad (3)$$

II-B. Lenguaje ensamblador: MIPS

MIPS, como lenguaje de programación, es uno de los más antiguos, pero también de los más poderosos computacionalmente hablando, debido a que se caracteriza principalmente por su trabajo con registros, los cuales son los accesos a memoria más rápidos para el procesador, por lo que es ciertamente veloz, pero restringido en su capacidad de almacenamiento, dejándonos ver que nuestra principal tarea al momento de trabajar en un código para la obtención de datos, es como "fluye" nuestro programa y trabaja con lo necesario en el momento y sentido oportuno, ya sean con los datos declarados en memoria o los ingresados a los registros principales o del coprocesador.

II-C. Diseño de la solución

Gracias al bajo nivel al que MIPS pertenece, un apartado importante de frente a una implementación será la transmisión del programa, donde nos preocuparemos como el usuario interactúa ya sea con la primera o segunda parte de este, por lo que se propone desplegar un menú por consola que indique el número de opciones y cuando será necesaria ingresar la opción pertinente.

Para la primera parte del programa a desarrollar nos ocuparemos de la obtención del seno o el coseno, dependiendo del ángulo ingresado por el usuario, por lo que primero se pedirá que opción quiere ocupar y luego se le indicará que debe ingresar el ángulo en cuestión, para poder comenzar con el cálculo pertinente.

Por último, para el apartado de la distancia euclidiana, como necesitamos 2 vectores, se le pedirán 6 inputs, 3 por cada vector, donde mediante la consola se le indicará que coordenada ingresar y en que momento para que el resultado sea el esperado.

III. MATERIALES Y MÉTODOS

En esta sección se describirá en profundidad el apartado técnico y la implementación realizada para la obtención del producto final.

III-A. Materiales

Para tanto la escritura del código como la compilación del mismo se ocupó el ejecutable MARS en su versión 4.5 de agosto del 2014. Por lo tanto, con base en lo descrito anteriormente, nuestra principal herramienta, además del dispositivo que se utilizara para la creación de todo el entorno (Computadora de escritorio con un procesador Intel I3 Décima Generación, 16 GB en RAM a 2666 MHz y 1 SSD donde será almacenado el ejecutable y el código), serán los registros del CPU que el mismo MARS y el lenguaje ensamblador nos facilita, aparte de las distintas cápsulas de información útil encontradas en línea y uvirtual.

III-B. Métodos

Describe el procedimiento desarrollado. Puede contener subsecciones.

III-B1. Seno y Coseno: Para la primera, el primero código, se decidió implementar el enunciado de la siguiente manera

1. Iniciamos pidiendo el valor del ángulo a evaluar y la opción entre 1, para el coseno, y 2, para el seno, a calcular.
2. Luego, ya sea para el coseno como para el seno, cargamos los factoriales que necesitemos en t1,t2,t3 o t4, como corresponda, para poder iniciar con la obtención de las potencias, donde al llamar a la primera subrutina exponencial, se cargan 2 contadores en s4 y s5 con tal de tener uno para la suma y otra para la multiplicación, donde con cada loop se sumara la cantidad ingresada y se restara 1, para finalmente comparar si nuestro contador de multiplicación está en 0 y así retornar la cantidad calculada, y así mismo para cada exponente que necesitemos.
3. Por otro lado, para la división, dedicamos la primera parte a la entera, donde nuevamente asignamos a s5 como contador e iremos realizando restas hasta que en el momento en que el divisor supere al dividendo, este nos llevara al bucle para la parte decimal de nuestro resultado.
4. Para la parte decimal primero nos aseguramos que no exista un residuo en s0, sino que nos llevara a un bucle donde calculamos mediante la resta del residuo los primeros 4 decimales.
5. Finalmente mostramos el resultado en consola mediante la composición de la parte entera, un punto y la parte decimal, terminando así la ejecución.

III-B2. Distancia Euclidiana: El proceder en la implementación para el segundo código es el siguiente

1. Nuestro primer objetivo dentro del procedimiento es la declaración en memoria (.data) de mensajes predeterminados para la obtención de datos (declaradas bajo el .ascii), donde se especifican cuando se deben ingresar las coordenadas de determinado vector, así como la cantidad de espacio a reservar para los flotantes de precisión simple (4 Bits).
2. Luego se desplegarán por consola los mensajes pertinentes del primer y segundo vector para la obtención de sus componentes, mediante las subrutinas de GetFirstVector y GetSecondVector, solicitando por cada mensaje un input entero y guardándolas en el registro f0, para luego almacenarlas en el coprocesador, mediante lwc1 en f12,f13,f14 y f4,f5,f6 respectivamente.
3. Procedemos a calcular la distancias euclidiana con los respectivos inputs, donde por coordenada obtenemos el cuadrado de la diferencia, operando a la par sub.s y mul.s, para luego sumar los resultados de cada cuadrado con add.s y obtener la raíz mediante la aproximación del método de Newton-Raphson, donde comparando una aproximación inicial y una con la media de los puntos iniciales de comparación, realizamos un bucle en donde se compararan los registros f3 y f1, que almacenan estas aproximaciones, hasta que estas sean iguales donde mediante una flag

saldremos del bucle en el momento en que esta sea 0, para luego almacenar el resultado final en f12.

4. Por último, ajustamos el puntero del stack mediante la restauración de la dirección de retorno con lw en el comando ra y addiu con el comando sp, para luego poder mostrar el mensaje en consola e invocar el procedimiento End que da por finalizado el programa.

IV. RESULTADOS

En la siguiente sección mostraremos los resultados obtenidos por MARS luego de ejecutado el código con las distintas operaciones. Figura 1,2,3,4,5,6 y 7.

Tabla I: Funciones Trigonometricas

Input	Seno	Coseno
2	1.337	-1.1436
14	-16877.1094	45361.1268
21	-324846.217	No calculable
6	-21.342	34.1938

Tabla II: Funciones Trigonometricas

(X1,Y1,Z1)	(X2,Y2,Z3)	Resultado
(2,3,4)	(21,2,3)	19.052559
(34,2,56)	(7,8,9)	54.534393
(45,2,18)	(2,3,4)	44.01136
(23,45,667)	(32,334,556)	313.43738

V. CONCLUSIONES

Como pudimos ver a través del desarrollo del producto a entregar, ciertos apartados de MIPS resultan fundamentales a la hora de programar, por lo que uno de los recursos principales que destacan son el uso de subrutinas y la segmentación misma del código, ya que aunque parezca que MIPS se programa con base en lo imperativo, o el "cómo" de los procedimientos, si tenemos en cuenta las bases en como este trabaja en conjunto con los registros y las distintas fases del ciclo de reloj que conlleva realizar cada instrucción, para todo desarrollo dentro de MIPS, es de suma importancia hacer de esta cualidad un apoyo más que una dificultad. Bajo el mismo marco las mayores dificultades se han presentado en el primer código, donde debimos simplemente cargar el factorial ya calculado, limitarnos en temas de inputs, ya que como MIPS soporta hasta 32 bits, para el seno el mayor ángulo es 21 y para el coseno es 14 y principalmente ya una vez implementada ciertas subrutinas, se tiende a entrelazar sin tener en cuenta que se están usando los mismos registros ocasionando que los datos no sean los que requerimos en el momento, por lo que se fueron resolviendo, ya sea reasignando los registros o reordenando las instrucciones para que estas adquieran los datos necesarios antes de que sean necesariamente actualizados. Así mismo, el producto final fue uno de calidad, donde pudimos cumplir los objetivos inicialmente propuestos y aquellos que ayudaran a Benjamín a conseguir la visión acerca del anhelado futuro.

VI. ANEXOS

```
Ingrese el valor del angulo:
2
Que operacion desea realizar?
1)Seno
2)Coseno
2
La estimacion es del resultado es de:-1.1436
```

Figura 1: Pruebas Funciones Trigonometricas 1

```
Ingrese el valor del angulo:
2
Que operacion desea realizar?
1)Seno
2)Coseno
1
La estimacion es del resultado es de:1.337
```

Figura 2: Pruebas Funciones Trigonometricas 2

```
Ingrese el valor del angulo:
14
Que operacion desea realizar?
1)Seno
2)Coseno
1
La estimacion es del resultado es de:-16877.1094
```

Figura 3: Pruebas Funciones Trigonometricas 3

```
Ingrese el valor del angulo:
14
Que operacion desea realizar?
1)Seno
2)Coseno
2
La estimacion es del resultado es de:45361.1268
```

Figura 4: Pruebas Funciones Trigonometricas 4

```
Introduzca las coordenadas del primer vector:
X:2
Y:3
Z:4
Introduzca las coordenadas del segundo vector:
X:21
Y:2
Z:3
La coordenada es: 19.052559
```

Figura 5: Pruebas Distancia Ecludiana 1

```
Introduzca las coordenadas del primer vector:
X:34
Y:2
Z:56
Introduzca las coordenadas del segundo vector:
X:7
Y:8
Z:9
La coordenada es: 54.534393
```

Figura 6: Pruebas Distancia Ecludiana 2

REFERENCIAS

- [1] D. L. Matematicas, “Serie de taylor,” *Avanzado*, vol. 1, p. 1, 2020. [Online]. Available: <https://www.disfrutalasmaticas.com/algebra/taylor-serie.html>
 - [2] P. Anderson and K. Vollmar, “Mars,” *Missouri State University*, vol. 1, p. 1, August 2014. [Online]. Available: <https://courses.missouristate.edu/kenvollmar/mars/download.htm>
 - [3] U. de Almeria, “Practica 2: Ejercicios de ensamblador en mips,” *cartagena99*, vol. 1, pp. 1–12, 2010. [Online]. Available: <http://www.hpca.ual.es/~jroca/docencia/pr/fao/mips/mips>
 - [4] U. U. P. de Catalunya, “Ensamblador MIPS y tipos de datos básicos,” *jarnau*, vol. 1, pp. 1–33, Marzo 2019. [Online]. Available: http://jarnau.site.ac.upc.edu/EC/sesion04_Tema2_Vectores.pdf
 - [5] C. U. U. de Santiago de Chile, “Clase 11: Direccionamiento en MIPS,” *Uvirtual*, vol. 1, pp. 1–15, 2023. [Online]. Available: <https://uvirtual.usach.cl/moodle/course/view.php?id=21415§ion=10>
- [1] [2] [3] [4] [5]