|  |  |
| --- | --- |
| PROGRAMA ACADÉMICO: **Ingeniería Informática** | |
| Módulo: **Laboratorio de Arquitectura de Hardware** | |
| Área : **Infraestructura** | Docentes: **Carlos Andrés Madrigal** |

1. **Laboratorio N. 5. Lenguaje Ensamblador**
2. **Objetivo de Aprendizaje**

* Comprender el conjunto de instrucciones del procesador.
* Desarrollar habilidades en la programación en lenguaje ensamblador.

1. **Fundamentos Teóricos**

**DEFINICIONES**

1. **LENGUAJE ENSAMBLADOR**

El lenguaje de maquina es un tipo de codificación a partir de secuencias de 0’s y 1’s que es comprendido por procesador del computador con el objetivo de ejecutar un conjunto de instrucciones, programar de esta manera resulta demasiado difícil. Cuando se quiere enviar instrucciones precisas al computador se hace a través del **lenguaje ensamblador**, que codifica las instrucciones de forma textual y un ensamblador se encarga de convertir el lenguaje ensamblador a lenguaje de maquina.

1. **INSTRUCTION SET ARCHITECTURE (ISA).**

Es el conjunto de instrucciones soportado por el procesador, en el caso de la Raspberry Pi 2 es el ARMv8. Algunas de las instrucciones son:

REGISTROS:

|  |  |  |
| --- | --- | --- |
| NOMBRE | NOMBRE ALTERNATIVO | DESCRIPCIÓN |
| r0-r3 |  | Usado para mantener argumentos en procedimientos. Estos registros son de almacenamiento temporal. r0 es usado para retornar el resultado de una función. |
| r4-r9 | v1 - v6 | Son usados para el almacenamiento de variables. |
| r10 | sl, v7 | Apuntador limite del Stack |
| r11 | ip | Es usado con r0-r3 para almacenamiento temporal |
| r13 | sp | Apuntador al Stack |
| r14 | lr | Registro Link, mantiene la dirección de retorno de una función |
| r15 | pc | Contador del programa, mantiene la dirección de la siguiente instrucción |

Operaciones Aritméticas

Operaciones:

ADD operand1 + operand2

ADC operand1 + operand2 + carry

SUB operand1 - operand2

SBC operand1 - operand2 + carry -1

RSB operand2 - operand1

RSC operand2 - operand1 + carry – 1

Sintaxis: <operation>{<cond>}{S} Rd, Rn, Operand2

Ejemplos

ADD r0, r1, r2

SUBGT r3, r3, #1

RSBLES r4, r5, #5

1. **Recursos**

Computador con acceso a internet.

Single Board Computer Raspberry pi 3

Cable MiniUSB de alimentación de 3A.

Tarjeta SD de 16GHz con el sistema operativo raspbian instalado.

Cable conversor HDMI-VGA

Pulsador

Pinzas, cortafrío y cables.

Resistencias

Protoboard

1. **Guía de desarrollo**

**PROCEDIMIENTO**

1. Instalar CodeBlocks y seguir los ejemplos del tutorial desde el punto 9 hasta el 9.9. “ARM\_Assembly\_Programming\_Using\_Raspberry\_Pi\_GUI.pdf”
2. Abrir el tutorial [“Assembly\_Language\_with\_the\_Raspberry\_Pi](http://www.science.smith.edu/dftwiki/index.php/Tutorial:_Assembly_Language_with_the_Raspberry_Pi)” http://www.science.smith.edu/dftwiki/index.php/Tutorial:\_Assembly\_Language\_with\_the\_Raspberry\_Pi y seguirlo completamente paso a paso, programando cada ejemplo en CodeBlocks. Analizar cada ejemplo y comprender cada instrucción.
3. Realizar un programa en código ensamblador sobre la Raspberry Pi que le pida al usuario máximo 50 valores numéricos los cuales deben almacenarse. Si el usuario ingresa el numero 0 indicará que no se quiere ingresar más datos. El programa debe obtener los siguientes resultados.

1. Ordenar y visualizar los números de forma ascendente.

2. Visualizar el número menor

3. Visualizar la moda

4. Calcular y visualizar el promedio de los valores.

Ayuda: Para almacenar información en lenguaje ensamblador. https://raspberrypiassembly.wordpress.com/

1. **Entregables**

**Informe**

* Este debe ser desarrollado de acuerdo al formato IEEE para la redacción de artículos científicos. "www.**ieee**.org/documents/MSW\_USltr\_**format**.doc"
* Se debe adjuntar archivo \*.s de los programas.

1. **Referencias**

<http://www.science.smith.edu/dftwiki/index.php/Tutorial:_Assembly_Language_with_the_Raspberry_Pi>

<http://www.peter-cockerell.net/aalp/html/frames.html>

<http://www.toves.org/books/arm/>

<http://arantxa.ii.uam.es/~gdrivera/sed/docs/ARMBook.pdf>

<http://www.cl.cam.ac.uk/projects/raspberrypi/tutorials/os/ok01.html>

<http://www.science.smith.edu/dftwiki/index.php/Tutorial:_Assembly_Language_with_the_Raspberry_Pi>

<http://thinkingeek.com/2013/01/09/arm-assembler-raspberry-pi-chapter-1/>

<https://raspberrypiassembly.wordpress.com/>

<http://simplemachines.it/doc/arm_inst.pdf>

<http://infocenter.arm.com/help/topic/com.arm.doc.qrc0001m/QRC0001_UAL.pdf>

<http://www.pp4s.co.uk/main/tu-trans-asm-arm.html>

1. **Criterios de Evaluación**

|  |  |
| --- | --- |
| **Aspecto a evaluar** | **Desarrollo del laboratorio y preguntas en el desarrollo** |
| **Escala de Valoración** | **Criterios** |
| 5 | Demuestra total comprensión del problema. Todos los requerimientos de la tarea están incluidos en la desarrollo de la práctica. |
| 4 | Demuestra considerable comprensión del problema. Todos los requerimientos de la tarea están incluidos en la desarrollo de la práctica. |
| 3 | Demuestra comprensión parcial del problema. La mayor cantidad de requerimientos de la tarea están incluidos en la desarrollo de la práctica. |
| 2 | Demuestra poca comprensión del problema. Muchos de los requerimientos de la tarea faltan en el desarrollo de la práctica. |
| 1 | No comprende el problema |
| 0 | No responde. No intentó hacer la tarea. |

|  |
| --- |
| **Observaciones:** |
|  |
|  |
|  |
|  |
|  |
|  |