

Laboratorio 6: MIPS Assembly Language Programming

Nicolás GÓMEZ MORGADO

Laboratorio Arquitectura de Computadores

23 de julio de 2024

1. Actividades

1.1. SPIM

Instale la aplicación PCSPIM en PC de laboratorio. Este software está disponible desde Moodle, debe descargar también los códigos assembler adjuntos.

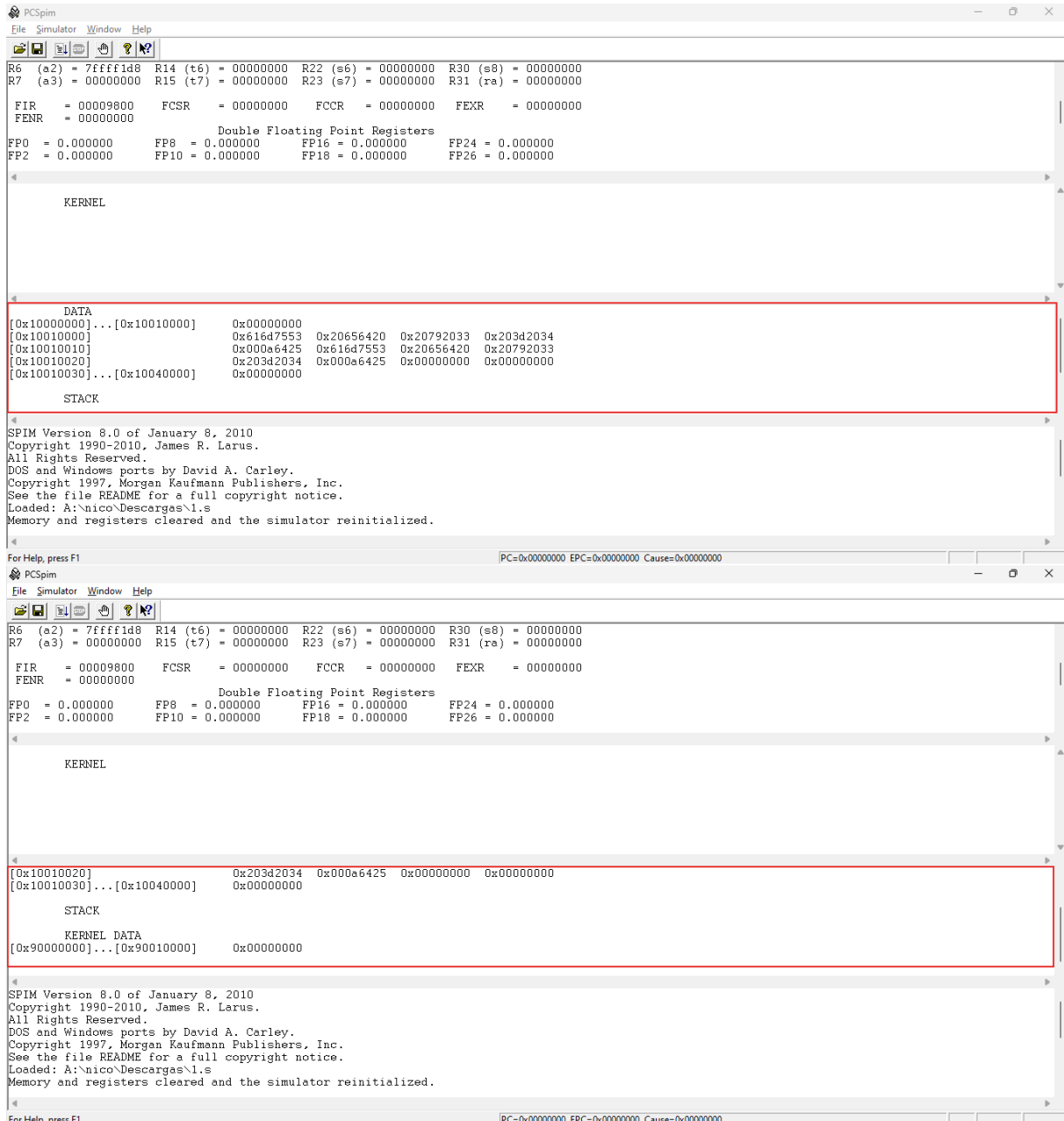
- a) Averigüe qué es una directiva y cuál es el propósito de las directivas `.data`, `.asciiz`, `.byte` y `.text`.

Una directiva es una instrucción que le dice al ensamblador que realice una tarea específica. Las directivas son instrucciones que no se traducen en instrucciones de máquina, sino que le dicen al ensamblador cómo ensamblar el código. A su vez las directivas `.data`, `.asciiz`, `.byte` y `.text` son directivas que se utilizan para definir el tipo de datos que se van a utilizar en el programa. Por ejemplo:

Directiva	Propósito
<code>.data</code>	Define el segmento de datos.
<code>.asciiz</code>	Define una cadena de caracteres.
<code>.byte</code>	Define un byte.
<code>.text</code>	Define el segmento de texto.

- b) Ejecute el simulador y cargue el programa “1.s”. Identifique el “segmento de datos” (captura del segmento de datos).

En el programa la zona del segmento de datos se puede identificar como el lugar donde aparecen los títulos **DATA**, **STACK** y **KERNEL DATA**. Como se muestra en las siguientes capturas:



PCSpin

File Simulator Window Help

R6 (a2) = 7ffff1d8 R14 (t6) = 00000000 R22 (s6) = 00000000 R30 (s8) = 00000000
R7 (a3) = 00000000 R15 (t7) = 00000000 R23 (s7) = 00000000 R31 (ra) = 00000000

FIR = 00009800 FCSR = 00000000 FCCR = 00000000 FEHR = 00000000
FENR = 00000000

Double Floating Point Registers
FP0 = 0.000000 FP8 = 0.000000 FP16 = 0.000000 FP24 = 0.000000
FP2 = 0.000000 FP10 = 0.000000 FP18 = 0.000000 FP26 = 0.000000

KERNEL

DATA

[0x10000000]...[0x10010000]	0x00000000	0x20656420	0x20792033	0x203d2034
[0x10010000]	0x616d7553	0x000a6425	0x616d7553	0x20656420
[0x10010010]	0x000a6425	0x203d2034	0x000a6425	0x00000000
[0x10010020]	0x203d2034	0x000a6425	0x00000000	0x00000000
[0x10010030]...[0x10040000]	0x00000000			

STACK

SPIM Version 8.0 of January 8, 2010
Copyright 1990-2010, James R. Larus.
All Rights Reserved.
DOS and Windows ports by David A. Carley.
Copyright 1997, Morgan Kaufmann Publishers, Inc.
See the file README for a full copyright notice.
Loaded: A:\nico\Descargas\1.s
Memory and registers cleared and the simulator reinitialized.

For Help, press F1

PC=0x00000000 EPC=0x00000000 Cause=0x00000000

PCSpin

File Simulator Window Help

R6 (a2) = 7ffff1d8 R14 (t6) = 00000000 R22 (s6) = 00000000 R30 (s8) = 00000000
R7 (a3) = 00000000 R15 (t7) = 00000000 R23 (s7) = 00000000 R31 (ra) = 00000000

FIR = 00009800 FCSR = 00000000 FCCR = 00000000 FEHR = 00000000
FENR = 00000000

Double Floating Point Registers
FP0 = 0.000000 FP8 = 0.000000 FP16 = 0.000000 FP24 = 0.000000
FP2 = 0.000000 FP10 = 0.000000 FP18 = 0.000000 FP26 = 0.000000

KERNEL

DATA

[0x10010020]	0x203d2034	0x000a6425	0x00000000	0x00000000
[0x10010030]...[0x10040000]	0x00000000			

STACK

KERNEL DATA

[0x90000000]...[0x90010000]	0x00000000
-----------------------------	------------

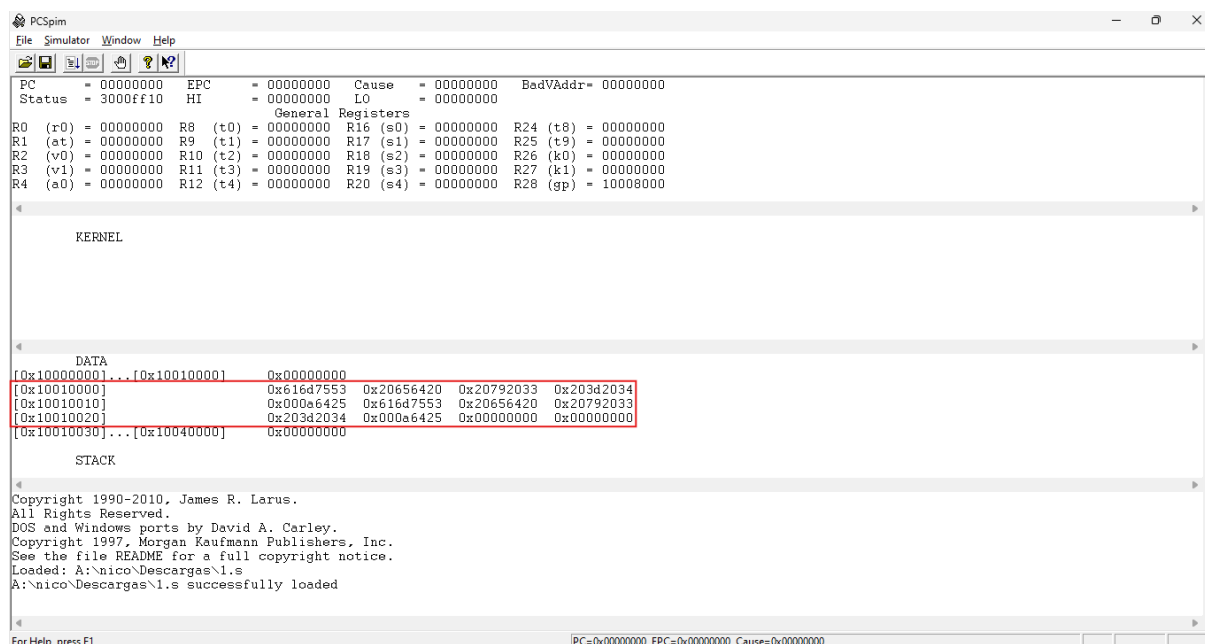
SPIM Version 8.0 of January 8, 2010
Copyright 1990-2010, James R. Larus.
All Rights Reserved.
DOS and Windows ports by David A. Carley.
Copyright 1997, Morgan Kaufmann Publishers, Inc.
See the file README for a full copyright notice.
Loaded: A:\nico\Descargas\1.s
Memory and registers cleared and the simulator reinitialized.

For Help, press F1

PC=0x00000000 EPC=0x00000000 Cause=0x00000000

- c) Identifique en qué lugar de la memoria se almacena el arreglo de bytes. Explícite la dirección de memoria de cada byte (utilice una calculadora online para la conversión de base: Suma de 3 y 4 = %d\n).

Las direcciones de memoria donde se almacena el arreglo de bytes se pueden identificar en la siguiente zona:



Siendo estas direcciones:

- 0x10010000
- 0x10010010
- 0x10010020

Mas especificamente:

- 0x10000000: 53 = 'S'
- 0x10000001: 75 = 'u'
- 0x10000002: 6d = 'm'
- 0x10000003: 61 = 'a'
- 0x10000004: 20 = ' '
- 0x10000005: 64 = 'd'
- 0x10000006: 65 = 'e'
- 0x10000007: 20 = ' '
- 0x10000008: 33 = '3'
- 0x10000009: 20 = ' '
- 0x1000000A: 79 = 'y'
- 0x1000000B: 20 = ' '
- 0x1000000C: 34 = '4'
- 0x1000000D: 20 = ' '
- 0x1000000E: 3d = '='
- 0x1000000F: 20 = ' '
- 0x10000010: 25 = '%'
- 0x10000011: 64 = 'd'
- 0x10000012: 0a = '\n'
- 0x10000013: 00 = Final cadena de caracteres.

- d) Busque online una tabla ASCII. Identifique en memoria de datos la frase “Suma de 3 y 4 = %d\n”.

<https://www.chileoffshore.com/es/toolkits/basic-conversion/ascii-to-hexa>

La frase transformada seria la siguiente:

53756D61206465203320792034203D2025645C6E.

La cual dividida en 8 caracteres por sector, se obtiene lo siguiente:

- 53 75 6D 61
- 20 64 65 20
- 33 20 79 20
- 34 20 3D 20
- 25 64 5C 6E

Estos caracteres de la frase se giran, obteniendo lo siguiente:

- 61 6D 75 53 = 0x616d7553 = 'Suma'
- 20 65 64 20 = 0x20656420 = ' de '
- 20 79 20 33 = 0x20792033 = '3 y '
- 20 3D 20 34 = 0x203D2034 = '4 = '
- 6E 5C 64 25 = 0x6E5C6425 = '%d\n'

- e) Averigüe y concluya qué ordenamiento de datos emplea SPIM (Little Endian o Big Endian).

SPIM emplea el ordenamiento de datos Little Endian, ya que en este ordenamiento los bytes menos significativos se almacenan en las direcciones de memoria más bajas, mientras que los bytes más significativos se almacenan en las direcciones de memoria más altas.

- f) Cargue el programa “2.s”. identifique en qué dirección de memoria se inicia el segmento de instrucciones.

El programa 2.s inicia el segmento de instrucciones en la dirección de memoria **0x00400000**.

- g) Identifique qué significado posee la etiqueta main.

La etiqueta main es donde se describen las instrucciones que se ejecutarán al inicio del programa. Es el punto de entrada del programa.

- h) Explique el propósito de la directiva align. Para ello cambie el valor de 2 a 4 de cualquiera de las declaradas y observe cambios (editar con procesador de texto).

La directiva .align se encarga de alinear la dirección de memoria de la siguiente instrucción a un múltiplo de 2. Si se cambia el valor de 2 a 4, la dirección de memoria de la siguiente instrucción se alineará a un múltiplo de 4.

Las primeras imagenes se tomaron sin reiniciar el programa, y por falta de tiempo se dejaron asi, pero solo son una duplicacion de la respuesta solicitada

1.2. Entrega

- a Documentar cada una de las acciones antes señaladas. Es de exclusiva responsabilidad del estudiante respetar el formato de entrega de esta guía. El formato de entrega debe ser en .pdf, capturas legibles, recortadas y centradas, el nombre del archivo debe contener su nombre y apellido (Laboratorio_6_Nombre_Apellido). Todas las actividades deben ser entregadas (subidas) a la plataforma digital en las fechas establecidas. Por cada hora de atraso, se descontará 1 pto.