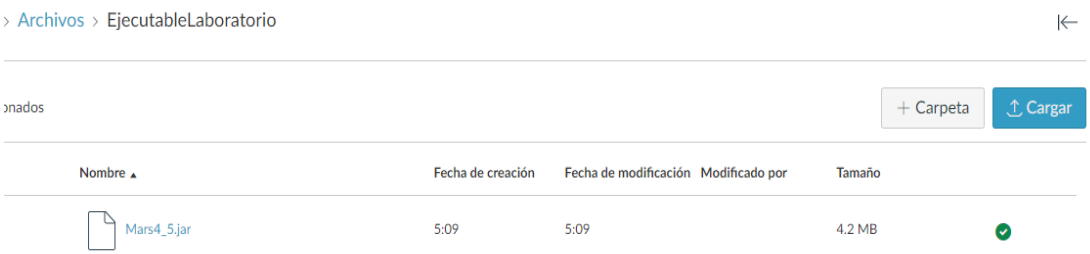
Asignatura	Datos del alumno	Fecha
Estructura de Computadores	Apellidos: Ferro Marin	9/09/2024
	Nombre: Geovanny Alfonso	

Laboratorio #1: Simulación y optimización de un programa en un procesador escalar segmentado

Preparación para el laboratorio

Para poder realizar la práctica de laboratorio deberá usarse la aplicación MARS (MIPS Assembler and Runtime Simulator), que básicamente es un IDE para MIPS. El archivo con el cual podrá ejecutar la actividad es un archivo Executable Jar File (.jar) que tiene como nombre: Mars4_5 y que se encuentra en la siguiente carpeta:



Dentro de las características de esta aplicación encontramos:

- Interfaz gráfica de usuario con control "apuntar y hacer clic" y editor integrado.
- Valores de registro y memoria fácilmente editables, similares a los de una hoja de cálculo.
- Visualización de valores en hexadecimal o decimal.
- Modo de línea de comandos para que los instructores prueben y evalúen fácilmente muchos programas.
- Registros de coma flotante, coprocesador1 y coprocesador2. Herramienta estándar: visualización y edición a nivel de bits de los registros de coma flotante de 32 bits (captura de pantalla).
- Ejecución en un solo paso a velocidad variable.

© Universidad Internacional de La Rioja (UNIR)

- Utilidad "Tool" para el control MIPS de dispositivos simulados. Herramienta estándar: Caché.

Conceptos Previos

Assembler: es un lenguaje de programación de bajo nivel (también conocido como ensamblador) que se utiliza para escribir programas que se ejecutan directamente en un computador o en otros dispositivos electrónicos. Los programas escritos en ensamblador son traducidos a lenguaje de máquina, que es el lenguaje que entiende el procesador del computador.

A diferencia de los lenguajes de programación de alto nivel, como C++ o Java, que se enfocan en la abstracción y la simplificación del proceso de programación, el ensamblador es un lenguaje de programación muy cercano a la arquitectura del procesador. Esto significa que los programas escritos en ensamblador son muy eficientes y rápidos, ya que se aprovechan al máximo las capacidades del procesador y se pueden controlar todos los detalles del hardware. Los archivos de este lenguaje tienen la extensión *.asm

Descripción del laboratorio

En esta práctica vamos a trabajar la ejecución de código Assembler que deberá pedir o mostrar por consola ciertos datos que permita la ejecución de los scripts propuestos. Esta actividad puede realizar de manera individual o grupal (máximo 4 estudiantes). Los scripts que se deben desarrollar son los siguientes:

1. Número mayor (mínimo 3 números). Se debe pedir al usuario por consola cuantos números desea comparar (mínimo 3 y máximo 5 números a comparar), posteriormente, por consola debe mostrar un mensaje que indique cuál es el mayor de los números digitados. El nombre del archivo

© Universidad Internacional de La Rioja (UNIR)

deberá guardarse con la extensión asm y utilizando la siguiente nomenclatura: ApellidoNombre_Mayor

2. Número menor (mínimo 3 números y máximo 5 números). Al igual que en el script anterior, se debe pedir al usuario por consola los números a comparar y mostrar por consolar el mensaje donde se indique cuál es el menor de los números digitados. El nombre del archivo deberá guardarse con la extensión asm y utilizando la siguiente nomenclatura: ApellidoNombre_Menor

3. Serie Fibonacci. Se debe pedir al usuario por consola cuántos número de la serie Fibonacci desea generar, posteriormente, y con base en el número introducido generé la serie y muestre por consola el resultado. Por ejemplo, la serie Fibonacci si el usuario decide ingresar los 5 primeros números son: 0, 1, 1, 2, 3. Además, de imprimirlos por consola deberá realizar la suma de todos los números de la serie que en este ejemplo sería: 7. El nombre del archivo deberá guardarse con la extensión asm y utilizando la siguiente nomenclatura: ApellidoNombre_Fibonacci

Entrega del laboratorio

Todos los lineamientos de entrega deben cumplirse para la revisión y evaluación de la actividad. Lo primero que se debe realizar antes de hacer el envío de la actividad es cargar cada uno de los scripts generados (tres en total) para la actividad al repositorio de GitHub, y compartir el enlace del repositorio en el documento del informe para validar el código de la actividad. El informe debe cumplir las siguientes indicaciones:

a. Un solo archivo en formato PDF con un informe en el que aparezcan los siguientes puntos:

© Universidad Internacional de La Rioja (UNIR)

Asignatura	Datos del alumno	Fecha
Estructura de Computadores	Apellidos: Ferro Marin	9/09/2024
	Nombre: Geovanny Alfonso	

- Introducción
- Desarrollo de la actividad
- Para cada uno de los scripts propuestos para la actividad se debe realizar 3 capturas de pantalla:

  a. Antes de compilar

  b. Después de compilar

  c. Después de ejecutar
- Conclusiones

b. El enlace al repositorio con los scripts generados en assembler (*.asm) siguiendo la nomenclatura propuesta en la descripción de la actividad. Cada uno de los scripts debe ser presentado en archivos independientes

- El código Assembler utilizado para la generación en cada uno de los scripts debe estar cargado en GitHub (https://github.com/). Guardar cada uno de los scripts de forma individual (no se deben cargar archivos formatos RAR en Github.
- Nota: si la actividad la realizan de forma grupal todos los integrantes debe publicar el enlace a su respectivo perfil en GitHub e incluir los enlaces al final del informe.
- Cada una de las líneas debe estar comentada con la respectiva descripción de que realiza cada instrucción.

Solución.

Introducción.

Para realizar el laboratorio hago uso del simulador MARS, en el ejecuto tres programas de lenguaje assembler, con el objetivo de comprender mejor la sintaxis y la lógica de programación en assembler, los scripts desarrollados permiten comparar números ingresados para encontrar el mayor, menor y también una serie Fibonacci que calcula la suma de los elementos.

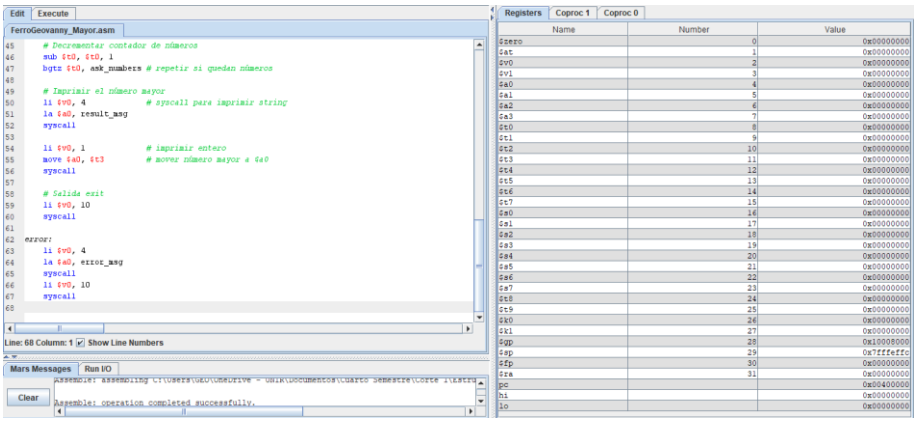
© Universidad Internacional de La Rioja (UNIR)

Asignatura	Datos del alumno	Fecha
Estructura de Computadores	Apellidos: Ferro Marin	9/09/2024
	Nombre: Geovanny Alfonso	

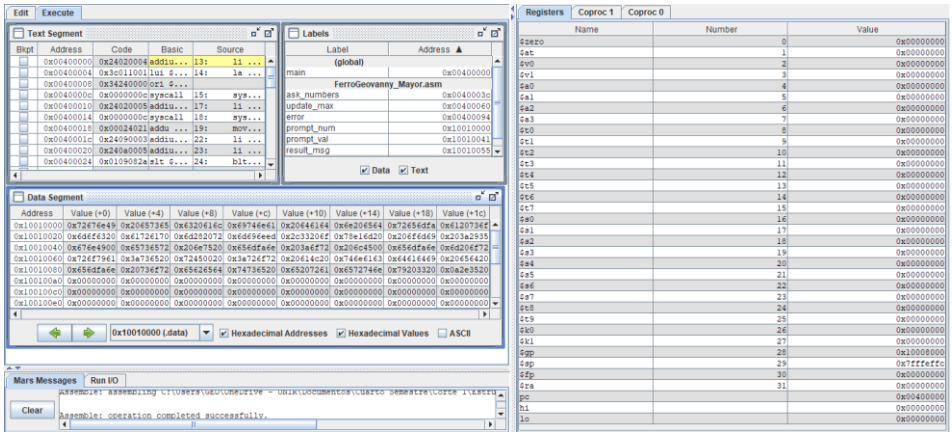
Número mayor.

Este algoritmo funciona sirve para pedir los números, compararlos y determinar el mayor.

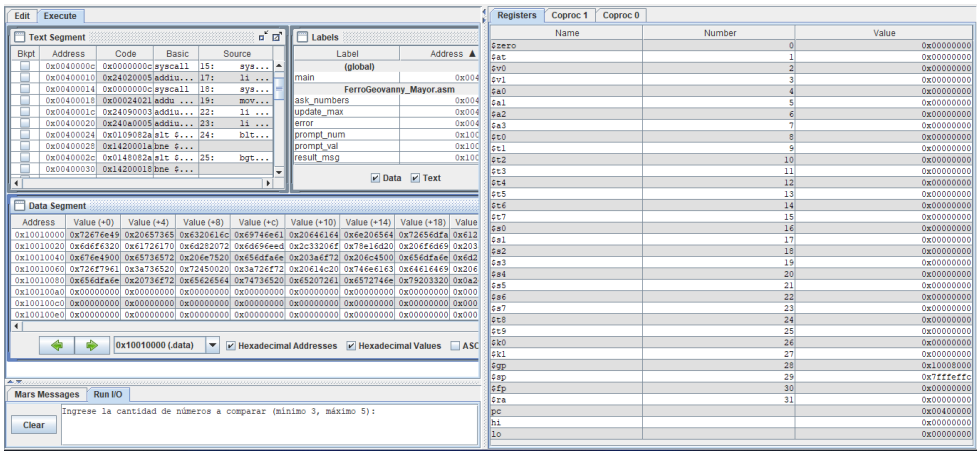
Antes de compilar



Después de compilar

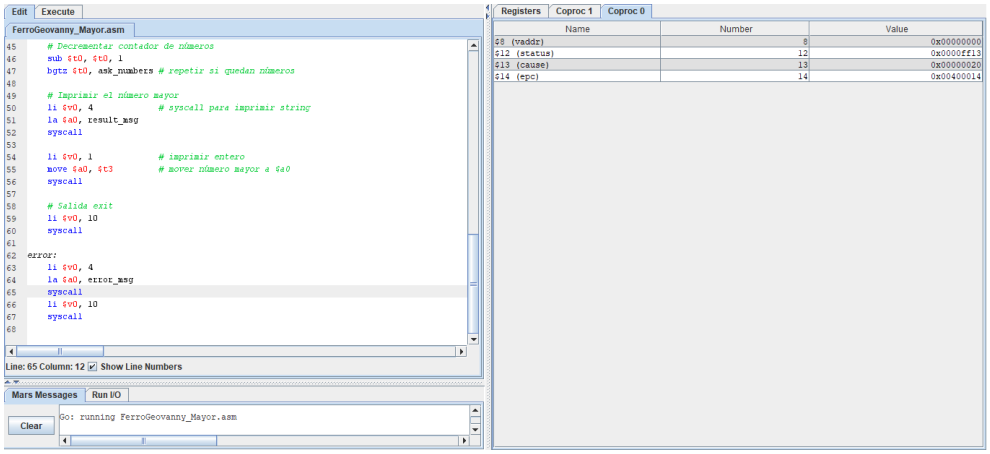


Después de ejecutar



© Universidad Internacional de La Rioja (UNIR)

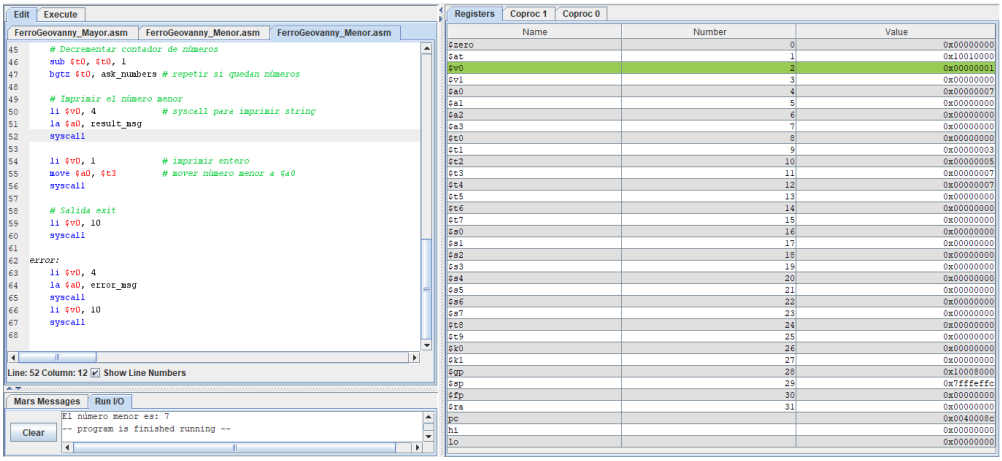
Asignatura	Datos del alumno	Fecha
Estructura de Computadores	Apellidos: Ferro Marin	9/09/2024
	Nombre: Geovanny Alfonso	



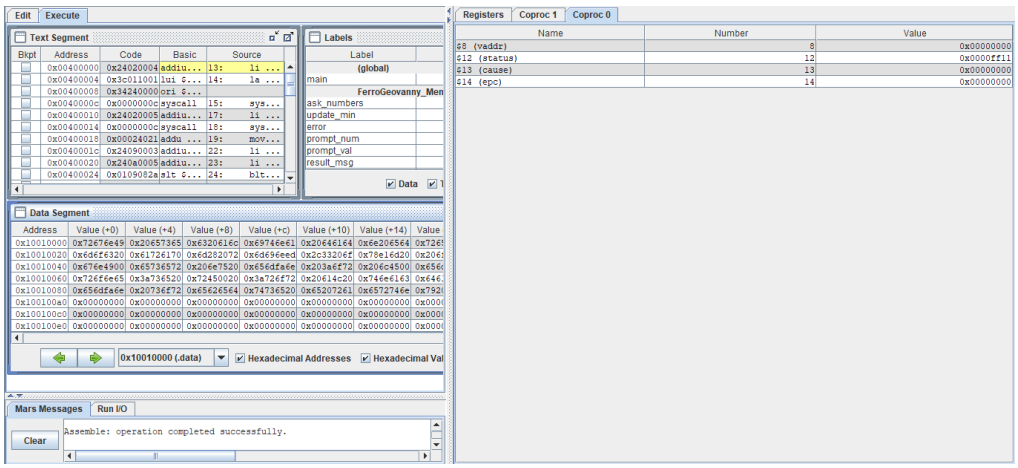
Número menor.

Este algoritmo funciona como el anterior, pero para encontrar el número menor

Antes de compilar



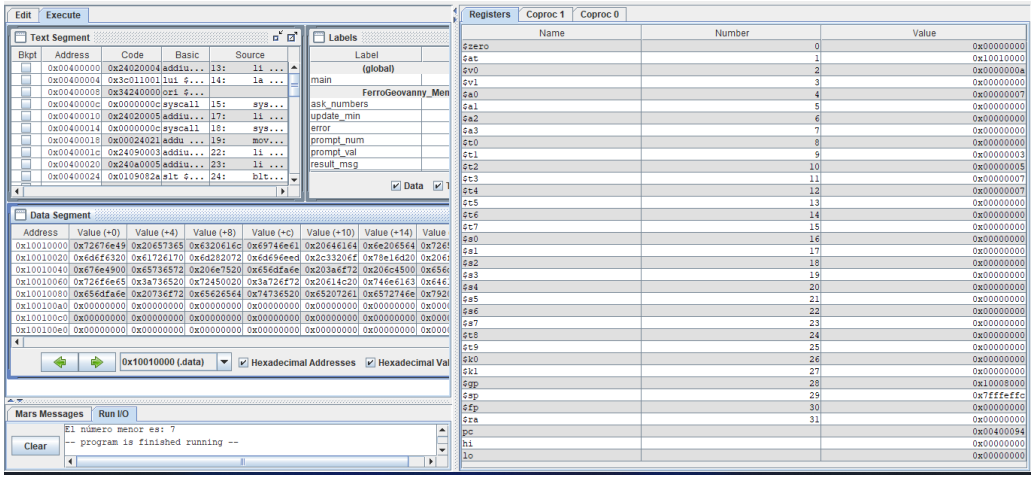
Después de compilar



© Universidad Internacional de La Rioja (UNIR)

Asignatura	Datos del alumno	Fecha
Estructura de Computadores	Apellidos: Ferro Marin	9/09/2024
	Nombre: Geovanny Alfonso	

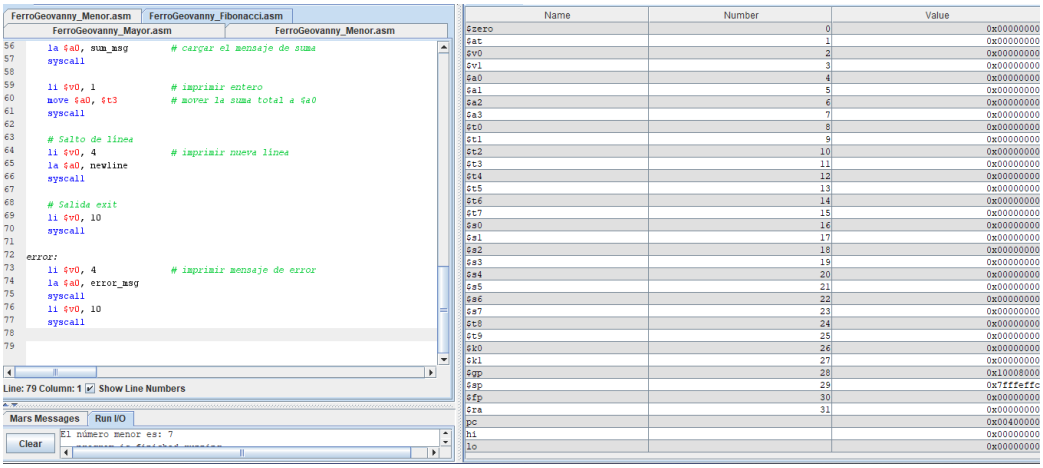
Después de ejecutar



Serie Fibonacci.

Este algoritmo explica como se genera la serie Fibonacci y como se realiza la suma de todos los términos generados.

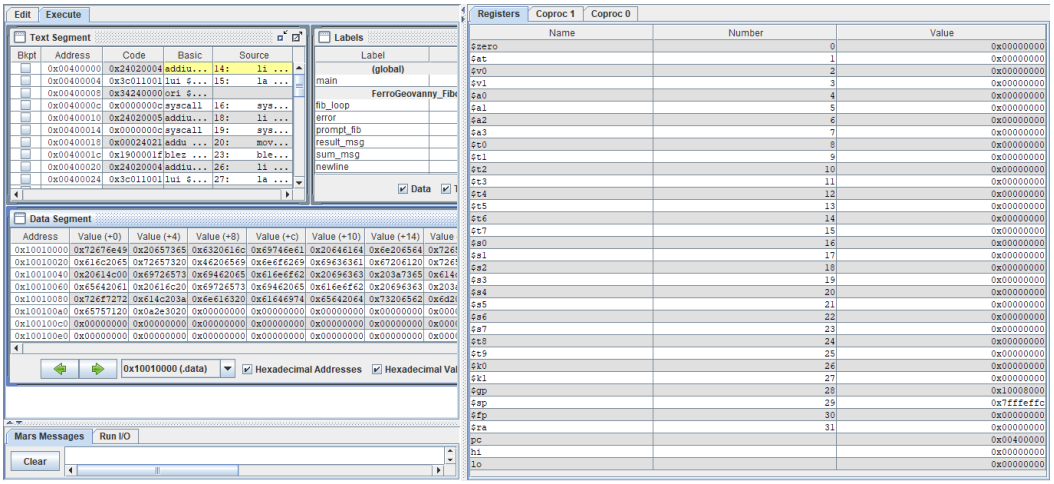
Antes de compilar



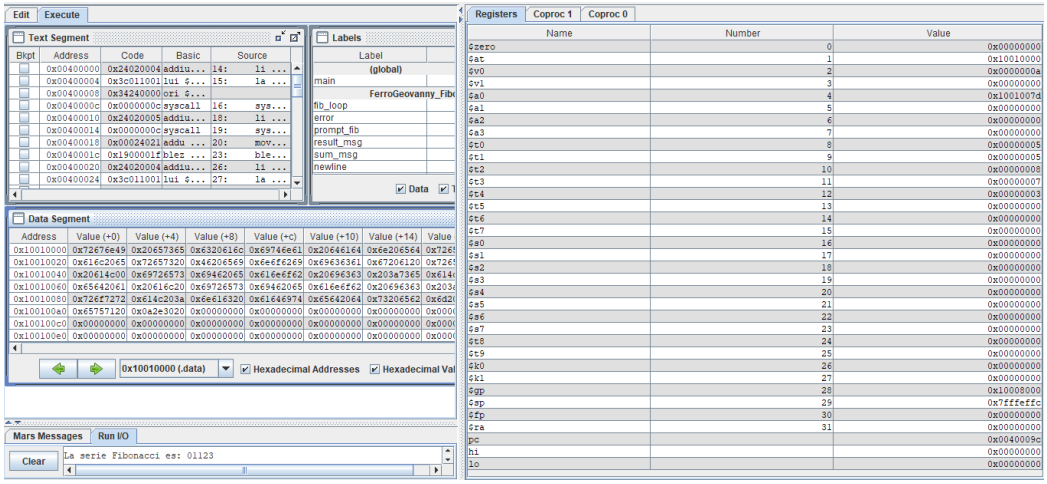
© Universidad Internacional de La Rioja (UNIR)

Asignatura	Datos del alumno	Fecha
Estructura de Computadores	Apellidos: Ferro Marin	9/09/2024
	Nombre: Geovanny Alfonso	

Después de compilar



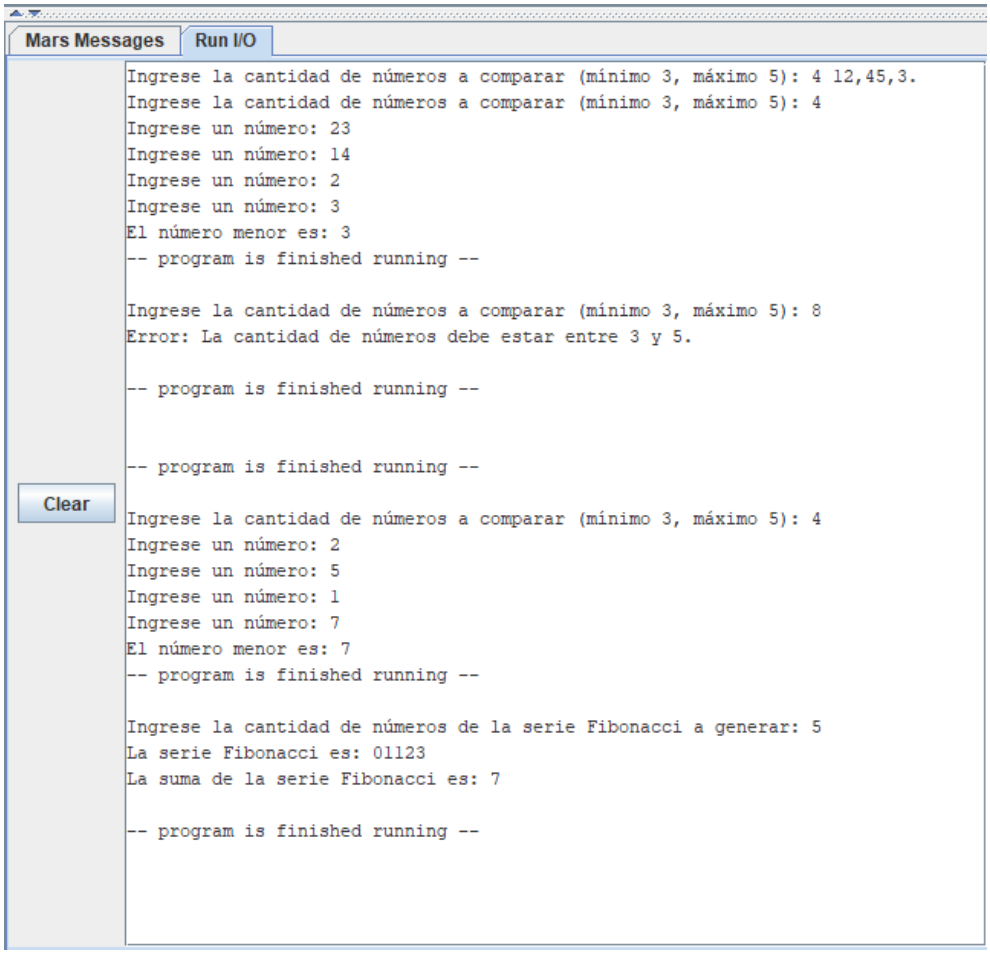
Después de ejecutar



Resumen de los scripts

© Universidad Internacional de La Rioja (UNIR)

Asignatura	Datos del alumno	Fecha
Estructura de Computadores	Apellidos: Ferro Marin	9/09/2024
	Nombre: Geovanny Alfonso	



Ingrese la cantidad de números a comparar (mínimo 3, máximo 5): 4 12,45,3.

Ingrese la cantidad de números a comparar (mínimo 3, máximo 5): 4

Ingrese un número: 23

Ingrese un número: 14

Ingrese un número: 2

Ingrese un número: 3

El número menor es: 3

-- program is finished running --

Ingrese la cantidad de números a comparar (mínimo 3, máximo 5): 8

Error: La cantidad de números debe estar entre 3 y 5.

-- program is finished running --

© Universidad Internacional de La Rioja (UNIR)

Asignatura	Datos del alumno	Fecha
Estructura de Computadores	Apellidos: Ferro Marin	9/09/2024
	Nombre: Geovanny Alfonso	

-- program is finished running --

Ingrese la cantidad de números a comparar (mínimo 3, máximo 5): 4

Ingrese un número: 2

Ingrese un número: 5

Ingrese un número: 1

Ingrese un número: 7

El número menor es: 7

-- program is finished running --

Ingrese la cantidad de números de la serie Fibonacci a generar: 5

La serie Fibonacci es: 01123

La suma de la serie Fibonacci es: 7

-- program is finished running –

Conclusiones

Con el desarrollo de este laboratorio logro mejorar mis habilidades en el manejo y manipulación de registros, también logro hacer la implementación de bucles y las llamadas al sistema para gestionar la entrada y salida. También refuerzo los conceptos básicos de programación en bajo nivel y estructura de computadores.

Se dejan los scripts en Github, se adjuntó el enlace para ingresar

link https://github.com/geoferro/LaboratorioEstructuraComputadores

© Universidad Internacional de La Rioja (UNIR)

Asignatura	Datos del alumno	Fecha
Estructura de Computadores	Apellidos: Ferro Marin	9/09/2024
	Nombre: Geovanny Alfonso	

(Computadores, s.f.) https://www.youtube.com/watch?v=wm8J7W8W1N4

(DevTek, s.f.) https://www.youtube.com/watch?v=Wrcy5zQxkVs

(www.youtube.com, s.f.)

Criterio de Evaluación	Descripción	Porcentaje
Compilación - Ejecución de los scripts	El código Assembler generado debe cumplir con los requisitos específicos de cada script (número mayor, número menor, serie Fibonacci) y funcionar correctamente. Cada línea del código debe estar correctamente comentada, explicando de manera clara y precisa lo que realiza cada instrucción.	60%
Organización Informe	El informe debe estar bien organizado, siguiendo las indicaciones proporcionadas (PDF, estructura, inclusión de enlaces al repositorio de GitHub). Inclusión de capturas de pantalla que muestren el estado del código antes de compilar, después de compilar, y después de ejecutar, para cada script.	30%

© Universidad Internacional de La Rioja (UNIR)

Asignatura	Datos del alumno	Fecha
Estructura de Computadores	Apellidos: Ferro Marin Nombre: Geovanny Alfonso	9/09/2024

Uso de GitHub y Presentación del Código	Los scripts deben estar correctamente cargados en GitHub, con un enlace al repositorio incluido en el informe. La presentación del código en GitHub debe ser clara y seguir la nomenclatura indicada.	10%

© Universidad Internacional de La Rioja (UNIR)