

Fecha de asignación: 26 de octubre de 2020

Fecha límite de entrega: 23:59 horas del 15 de noviembre de 2020

1. Objetivo

- Estudiar la arquitectura del conjunto de instrucciones MIPS de 32 bits.
- Diseñar, codificar, ensamblar, simular y depurar programas escritos en lenguaje ensamblador MIPS.
- Familiarizarse con el uso de un entorno de desarrollo de software de bajo nivel.

2. Descripción

En esta práctica, cada equipo de trabajo debe diseñar, programar, simular y verificar un programa escrito en lenguaje ensamblador MIPS, que debe estar en capacidad de determinar la frecuencia de aparición de diferentes cadenas de caracteres en un archivo de entrada.

3. Procedimiento

Cada equipo de trabajo debe desarrollar el programa en lenguaje ensamblador MIPS empleando la herramienta MARS¹. Las condiciones a las que se debe ajustar el desarrollo son las siguientes:

- El archivo de entrada por analizar debe ser un archivo de texto plano.
- Las cadenas de caracteres por identificar y contar en el archivo de entrada deben ser por lo menos tres diferentes, y estas serán consultadas al usuario del programa en el momento de su ejecución.
- Los resultados generados por el programa consistirán en ilustrar cada una de las cadenas de caracteres identificadas y su número de apariciones en el archivo de entrada. Estos resultados deben ser escritos en un archivo de salida, también en texto plano.

¹ Disponible en <http://courses.missouristate.edu/KenVollmar/MARS/>

- La programación en bajo nivel debe ser estructurada, es decir, debe estar basada en el uso de procedimientos, de manera que la organización modular del programa facilite su comprensión y favorezca la reutilización de código a lo largo del mismo.
- El código desarrollado debe seguir la convención para el uso de registros MIPS.
- El código fuente en ensamblador debe estar comentado de manera asertiva.

Son diversos los aspectos específicos de la implementación de la solución al problema que tendrán que ser definidos por el equipo de trabajo. En todos los casos, las decisiones de diseño deben estar ampliamente explicadas y justificadas. La calidad de estas decisiones será valorada durante la sustentación del trabajo.

También serán valoradas la creatividad y elegancia en la programación, así como la calidad de la interacción con el usuario mediante el uso de los servicios SYSCALL. El rendimiento del programa también será tenido en cuenta, entendiéndolo como el número de instrucciones ejecutadas para completarlo.

4. Informe

Cada equipo de trabajo debe realizar un informe escrito que incluya una descripción de alto nivel de la aplicación (mediante pseudocódigo, diagrama de flujo, etc.), una relación de los procedimientos empleados (con la descripción de sus entradas, salidas y funcionalidad implementada), la explicación y justificación de las decisiones de diseño tomadas durante el desarrollo de la práctica, y los resultados de ejecución en los diversos escenarios soportados. El informe debe cerrarse con las observaciones y conclusiones pertinentes. Este documento debe estar en formato **PDF** y ser subido a la plataforma antes del cierre del plazo de entrega, con el código fuente en lenguaje ensamblador como adjunto. El informe tiene un peso del 30% en la calificación global de la práctica.

5. Sustentación

Ambos miembros del equipo de trabajo deben demostrar un dominio completo del desarrollo de la práctica. El profesor o el monitor plantearán preguntas para evaluar los conocimientos adquiridos (algoritmos empleados e implementación del programa; fundamentos conceptuales de la arquitectura del conjunto de instrucciones MIPS: formatos de instrucción, modos de direccionamiento, convención de uso de registros MIPS, uso del *stack pointer*, programación basada en procedimientos; y manejo del simulador MARS). La sustentación tiene un peso del 70% en la calificación global de la práctica.