

Guía de presentación del tema 3

Fundamentos de la programación en ensamblador

El objetivo de este tema es entender los principales aspectos relacionados con la programación en ensamblador. El lenguaje ensamblador es un lenguaje legible por un programador que constituye la representación más directa del código máquina específico de una arquitectura o computador. También se persigue que el alumno aprenda a realizar pequeños programas escritos en ensamblador. Este tema se compone de los siguientes bloques:

1. Fundamentos básicos de la programación ensamblador.
2. Ensamblador de MIPS 32, modelo de memoria y representación de datos.
3. Formato de las instrucciones y modos de direccionamiento.
4. Llamadas a procedimientos y uso de pila.

Como ejemplo de lenguaje ensamblador se utilizará el ensamblador del MIPS 32, un procesador RISC de 32 bits, que posee un juego de instrucciones muy sencillo de usar. Se trata de un lenguaje muy didáctico que no recurre a instrucciones complicadas. Además, existen diversos simuladores que se pueden utilizar para realizar programas en este lenguaje ensamblador.

1. Fundamentos básicos de la programación en ensamblador

En este primer bloque se describen cuáles son las principales características que incluye un juego de instrucciones:

- Operandos con los que trabaja y que pueden residir en registros, en la memoria o en la propia instrucción.
- Direccionamiento de la memoria. Cómo son los mecanismos que se emplean para poder acceder a los datos almacenados en la memoria a través de las instrucciones máquina.
- Los modos de direccionamiento, que especifican el lugar y la forma de acceder a los operandos a los que hacen referencia las instrucciones máquina.
- El tipo y el tamaño de los operandos.
- Distintas operaciones que realizan: aritméticas, lógicas, de transferencia, etc.
- Instrucciones de control de flujo, que permiten realizar saltos condicionales, incondicionales y llamadas a funciones y procedimientos.
- El formato y la codificación del juego de instrucciones.

A continuación, se describen los principales tipos de instrucciones, mostrando ejemplos del ensamblador del MIPS 32: instrucciones de transferencia, aritméticas, lógicas, de desplazamiento, de comparación, de control de flujo, de conversión, de entrada/salida y llamadas al sistema.

A lo largo del tema, se muestra cómo se realiza la traducción de pequeños fragmentos escritos en un lenguaje de alto nivel al ensamblador del MIPS 32. En concreto, se describen cómo se traducen a ensamblador, las principales estructuras de control de flujo de lenguajes de alto nivel como C o Java: bucles while, do-while y sentencias if-else.

2. Ensamblador de MIPS 32, modelo de memoria y representación de datos.

En esta segunda parte se presenta el modelo de memoria de un computador y las instrucciones que ofrece el MIPS 32 para el acceso a memoria a distintos tipos de datos: bytes, medias palabras y palabras. A continuación, se muestra cómo representar en ensamblador diferentes tipos de datos de los lenguajes de alto nivel: booleanos, caracteres, enteros, reales, vectores, matrices y cadenas de caracteres. También se muestran las instrucciones del MIPS 32 que permiten trabajar con números en coma flotante, tanto de simple como de doble precisión.

3. Formato de instrucciones y modos de direccionamiento

Esta parte comienza introduciendo la diferencia que hay entre una instrucción en ensamblador y una pseudoinstrucción en ensamblador. Una instrucción en ensamblador se corresponde con una instrucción máquina, mientras que una pseudoinstrucción es una instrucción que se puede utilizar en programas escritos en ensamblador, pero que no tiene una correspondencia directa con una instrucción máquina. A continuación, se presenta el concepto de modo de direccionamiento. El modo de direccionamiento es el procedimiento que permite determinar la ubicación de un operando, un resultado o una instrucción. En esta parte se describen los siguientes modos de direccionamiento:

- Implícito
- Inmediato
- Directo a registro
- Directo a memoria
- Indirecto a registro
- Indirecto a memoria
- Relativo a registro base
- Relativo a registro índice
- Relativo a PC
- Direccionamiento de Pila

En esta parte del tema, se presenta también la forma de diseñar diferentes juegos de instrucciones, la diferencia entre computadores CISC y RISC. Por último, se presentan los diferentes modelos de ejecución de un computador. El modelo de ejecución indica el número de direcciones y tipo de operandos que se pueden especificar en una instrucción máquina.

4. Llamadas a procedimientos y uso de pila

Este último bloque está dedicado a las llamadas a procedimientos y uso de la pila cuando se programa en ensamblador. Esta parte comienza ilustrando el proceso de llamadas a funciones o procedimientos en el MIPS 32 y las instrucciones máquina de que dispone para poder invocar a funciones y retornar de ellas. A continuación, describe el uso de la pila en el MIPS 32 y la importancia que tiene en todo el proceso de llamadas a funciones. También se describe el convenio en el uso de registros del MIPS 32 y el convenio de paso de parámetros en este ensamblador. Se describe el marco de pila o registro de activación, que es el mecanismo que utilizan los compiladores para activar los procedimientos o funciones en los lenguajes de alto nivel. Se ilustra de forma detallada todo el proceso necesario para crear el marco de pila asociado a una función, indicando los pasos que debe realizar la subrutina que realiza la llamada y los pasos que debe realizar la subrutina llamada. Por último, se describe todo el proceso traducción y ejecución de

programas, describiendo de forma resumida las funciones que realiza el compilador, el ensamblador, el enlazador y el cargador.

Material asociado

Como material asociado a este tema se incluye el material de teoría y una colección de ejercicios propuestos y resueltos sobre los aspectos tratados en el tema.

Lecturas recomendadas

- Capítulo 3 y 4 del libro “Problemas resueltos de estructuras de computadores” (GARCIA CARBALLEIRA, Félix et al.).
- Capítulo 2 del libro “Computer organization and design. The hardware/software interface” (PATTERSON, David, et al).
- Capítulo 12, 13, y 15 del libro “Computer Organization and Architecture” (STALLINGS, William).
- Apéndice A del libro “Computer Organization and Architecture” (STALLINGS, William).