

Registros y Segmentos

Daniel Araya Román

Visión General

Las estructuras de datos que se utilizan en ensamblador corresponden a las estructuras físicas de los registros de la memoria principal. Sus instrucciones tienen una relación directa y unívoca con las instrucciones del lenguaje máquina.

Ventajas

- **Velocidad:** Los registros son los dispositivos de almacenamiento directamente del procesador.
- **Flexibilidad:** Los registros pueden ser utilizados para almacenar cualquier tipo de dato.
- **Rutinas Optimizadas:** Las rutinas de ensamblador naturalmente están optimizadas para trabajar con registros.
- **Control absoluto del PC:** El programador tiene control absoluto sobre el PC, lo que permite optimizar el flujo de ejecución del programa.

Desventajas

- **Tiempo de programación:** El tiempo de programación es mayor que en lenguajes de alto nivel.
- **Código fuente extenso:** El código fuente es extenso y difícil de leer.
- **Dependencia de la arquitectura:** El código fuente depende de la arquitectura del procesador, por lo que no es portable.
- **Afectar recursos del sistema:** El programador debe tener cuidado de no afectar recursos del sistema.

Segmentos

Es un bloque de memoria designado a un programa, que puede estar ubicado en cualquier parte de la memoria principal. Sólo necesita tanto espacio como el programa requiera para su ejecución. Se puede tener cualquier número de segmentos en memoria. Los segmentos principales son, el **segmento de código**, el **segmento de datos** y el **segmento de pila**.

Segmento de Código

El *registro CS*, direcciona este segmento, por medio de la primera instrucción ejecutable del programa, que se encuentra al inicio del mismo. El segmento de código contiene las instrucciones para la ejecución del programa. El sistema operativo enlaza esa localidad, para iniciar la ejecución del programa.

Segmento de Datos

El *registro DS*, direcciona este segmento, que contiene las variables y constantes del programa. El segmento de datos es de lectura y escritura, por lo que se pueden modificar los valores de las variables y constantes.

Segmento de Pila

El **registro SS**, direcciona este segmento, almacena las direcciones y los valores de manera temporal, para el uso de las rutinas del programa. El segmento de pila es de lectura y escritura, por lo que se pueden modificar los valores de las direcciones y los valores de la pila.

Registros del Procesador

Registros de Segmento

- **CS**: *Code Segment*
- **DS**: *Data Segment*.
- **SS**: *Stack Segment*.
- **ES**: *Extra Segment (ES)*.
- **FS**: *Extra Segment (FS)*.
- **GS**: *Extra Segment (GS)*.

Registros de Propósito General

- **AX**: *Accumulator*.
- **BX**: *Base*.
- **CX**: *Counter*.
- **DX**: *Data*.

Registros apuntadores de índice

- **SI**: *Source Index*.
- **DI**: *Destination Index*.

Registros apuntadores de pila

- **SP**: *Stack Pointer*.
- **BP**: *Base Pointer*.

Registro CS

DOS almacena la dirección de la primera instrucción ejecutable del programa en el registro **CS**. Junto al registro **IP**, **registro apuntador de instrucción**, indica la dirección de la siguiente instrucción a ser buscada para su ejecución.

Registro DS

DOS almacena la dirección del segmento de datos en el registro **DS**. El registro **DS** direcciona el segmento de datos, que contiene las variables y constantes del programa. Más un valor de desplazamiento, direcciona una referencia a un *byte (8-bit)* en específico dentro del segmento de datos.

Registro SS

DOS almacena la dirección del segmento de pila en el registro **SS**. El registro **SS** direcciona el segmento de pila, más un valor de desplazamiento del registro **SP**, direcciona una referencia a un *word (16-bit)* en específico dentro del segmento de pila que está siendo direccionado.

Registro IP

DOS almacena la dirección de la siguiente instrucción a ser ejecutada en el registro **IP**. Está asociado al registro **CS**, el **IP** indica la dirección de la instrucción actual en el segmento de código, que se está ejecutando.

Registro apuntadores de índice

Los registros **SI** y **DI** son registros de índice de 16 bits, que se utilizan para direccionamiento indexado. Ambos se pueden realizar operaciones de cadenas de caracteres, el registro **SI** está relacionado con el registro **DS**, mientras que el registro **DI** está relacionado con el registro **ES**.

Registro Generales

Los registros **AX**, **BX**, **CX** y **DX** son registros de propósito general de 16 bits. Estos registros se pueden utilizar para almacenar datos, realizar operaciones aritméticas y lógicas, y direccionamiento de memoria.

Registro AX

El registro **AX** es el registro acumulador de 16 bits. Se utiliza para operaciones de entrada y salida, y para operaciones aritméticas y lógicas.

Registro BX

El registro **BX** es el registro base de 16 bits. Es el único registro de propósito general que puede ser utilizado para direccionamiento indexado. Se utiliza para operaciones aritméticas y lógicas.

Registro CX

El registro **CX** es el registro contador de 16 bits. Se utiliza para contar iteraciones en ciclos, y para operaciones aritméticas y lógicas.

Registro DX

El registro **DX** es el registro de datos de 16 bits. Se utiliza para operaciones de entrada y salida, y para operaciones aritméticas y lógicas.

Registro apuntadores de pila

Los registros **SP** y **BP** son registros de propósito general de 16 bits. Estos registros se utilizan para direccionamiento de memoria, y para direccionamiento de pila.

Registro SP y BP

Los registros **SP** y **BP** son registros de propósito general de 16 bits. Estos registros se utilizan para direccionamiento de memoria, y para direccionamiento de pila.

Registro de *banderas*

El registro **FLAGS** es un registro de 16 bits, que contiene los *bits* de estado del procesador. Estos *bits* de estado se utilizan para indicar el resultado de una operación aritmética o lógica, y para controlar el funcionamiento del procesador.

La siguiente tabla muestra los *bits* de estado del registro **FLAGS**.

Bit	Nombre	Descripción	Bit	Nombre	Descripción
0	CF	<i>Carry Flag</i>	8	OF	<i>Overflow Flag</i>
1			9	DF	<i>Direction Flag</i>
2	PF	<i>Parity Flag</i>	10	IF	<i>Interrupt Enable Flag</i>
3			11	TF	<i>Trap Flag</i>
4	AF	<i>Auxiliary Carry Flag</i>	12	SF	<i>Sign Flag</i>
5			13	ZF	<i>Zero Flag</i>
6	ZF	<i>Zero Flag</i>	14		
7	SF	<i>Sign Flag</i>	15		