

Directivas

Una directiva es un comando incrustado en el código fuente, que el ensamblador reconoce y actúa en base a ésta. Las directivas no se llevan a cabo en tiempo de ejecución, mientras que las instrucciones sí.

Directiva EQU (EQUIvalence)

Sintaxis: nombre EQU expresión

- La directiva EQU asigna un nombre simbólico al valor de una expresión. El compilador, cuando encuentre en el código dicho nombre simbólico, lo sustituirá por el valor de la expresión. La expresión indicada puede ser:
 1. Una constante numérica.
 2. Una referencia de dirección (cualquier modo de direccionamiento).
 3. Cualquier combinación de símbolos y operaciones que generen un valor numérico.
 4. Otro nombre simbólico.

Ejemplos:	COLUMNAS	EQU	80
	FILAS	EQU	25
	PANTALLA	EQU	FILAS*COLUMNAS
	MOVER	EQU	MOV
	EDICION	EQU	"PULSAR TECLA"
	DIRECCION	EQU	[SI+BX]

Directiva DB (Define Byte)

Sintaxis: [nom_variable] EQU expresión

- La directiva DB reserva memoria para datos de tipo byte (8 bits), o para una variable de este tipo, inicializando o no dicho byte y los posteriores.
- El nombre de la variable es opcional, y se asigna dicho nombre para el primer byte (recordar que el lenguaje máquina no entiende de variables, si no de direcciones de memoria. Una variable es una etiqueta o referencia para el compilador, el cual sustituye cada referencia a esta variable por la dirección de memoria correspondiente).
- La expresión es el valor inicial de la variable, y puede ser:

- La expresión es el valor inicial de la variable, y puede ser:
 - Una constante numérica positiva o negativa con rango de byte (-128 a +127, ó de 0 a 255).
 - Una cadena de caracteres delimitada por comillas simples o dobles.
 - Un signo interrogante (?), que indica valor indefinido. Sólo reserva espacio en la memoria.
 - n DUP(expresión)**, que repite 'n' veces la 'expresión'.

Ejemplos:

VALOR	DB	111
TEXTO	DB	"HOLA, SOY UN TEXTO"
CALCULO	DB	15*2
RESERVA	DB	15 DUP(?)
NODEFIN	DB	?

- De modo similar a DB, se pueden definir otros tipos de variables en lenguaje ensamblador. Lo único que varía es el tamaño de los datos y el nombre de la directiva:

Operador DUP (DUPlicate)

Sintaxis: **num DUP(valor)**

- Define en memoria la repetición de 'num' veces de 'valor'.
- El parámetro 'valor' puede ser un valor entero, una cadena de caracteres entrecomillada u otro operador.
- Ejemplos:

```
DB      20 DUP(100)      ; Repetir 20 veces 100
DB      64 DUP('HOLA')   ; Repetir 64 veces 'HOLA'
DB      256 DUP(?)        ; Asignar 256 bytes indefinidos
```

DB (Define Byte)	Reservar datos de tamaño byte (8 bits)
DW (Define Word)	Reservar datos de tipo palabra (16 bits)
DD (Define Doubleword)	Reservar datos de tipo doble palabra (32 bits)
DQ (Define Quadword)	Reservar datos de tipo quádruple palabra (64 bits)
DT (Define Terabyte)	Reservar datos de tipo terabyte (80 bits o 10 bytes)

Directiva END

Sintaxis: **END [expresión]**

- Indica el final del programa fuente. El operando 'expresión' indica la dirección de comienzo del programa fuente, que es, normalmente, una etiqueta.
- Ejemplo:

```
END
END      INICIO
```

En el caso de varios módulos fuentes que se enlazan juntos para formar un solo programa ejecutable, sólo el módulo principal puede especificar expresión.

Directiva CODE

La directiva .CODE identifica el área de un programa que contiene instrucciones:

.code

Directiva ORG

ORG expresión

Establece el contador de ubicación a expresión.

Directivas PUBLIC y EXTRN

▪ **PUBLIC**

- Formato: PUBLIC símbolo
- Descripción: permite que los símbolos especificados sean accesibles por otros módulos durante un enlazado conjunto (LINK). El módulo que quiere invocar al símbolo debe contener la sentencia EXTRN.
- Ejemplo: PUBLIC dato
 dato DB 23h

▪ **EXTRN**

- Formato: EXTRN símbolo: tipo
- Descripción: identifica a los símbolos que fueron definidos (y declarados PUBLIC) en otro módulo.
- Ejemplo: EXTRN dato:byte

Directivas PROC y ENDP

▪ **PROC (PROCEDURE)**

- Formato: nombre_procedimiento PROC [atributo]
- Descripción: indica el comienzo del procedimiento "nombre_procedimiento". Los procedimientos deben finalizar con la directiva ENDP
- Operandos: atributo = NEAR o FAR (si no se pone nada se supone NEAR)
- Ejemplo: rutina PROC
 rutina ENDP

▪ **ENDP (END PROCEDURE)**

- Formato: nombre_procedimiento ENDP
- Descripción: indica el final de un procedimiento
- Operandos: nombre_procedimiento es obligatorio

Directiva MACRO y ENDMACRO

▪ **MACRO**

- Formato: nombre_macro MACRO lista de parámetros
- Descripción: especifica el nombre y los parámetros de la macro. Los parámetros se separan por comas. El final de la macro se especifica mediante la directiva ENDM.
- Ejemplo: sumar MACRO operando1, operando2, resultado
 cuerpo de la macro
 ENDM

- **ENDM (END MACRO)**
 - Formato: ENDM
 - Descripción: indica el final de una macro

Directiva STACK

Indica que comienza un segmento de pila.

Directiva MODEL

Determina el tipo de modelo de memoria. La sintaxis es:

.MODEL modelomemoria [opcionesmodelo] donde el campo modelodememoria puede tomar los siguientes valores

Modelo	Descripción
Tiny (diminuto)	Un solo segmento, contiene código y datos. Este modelo lo utilizan los programas que tienen la extensión .com en sus nombres de archivo
Small (pequeño)	Un segmento de código y un segmento de datos. Todo el código y los datos son cercanos, de manera predeterminada
Medium (mediano)	Varios segmentos de código y un solo segmento de datos
Compact (compacto)	Un segmento de código y varios segmentos de datos
Large (grande)	Varios segmentos de código y de datos
Huge (enorme)	Igual que el modelo grande (large), pero los elementos individuales de datos pueden ser más grandes que un solo segmento
Flat (plano)	Modo protegido. Utiliza desplazamientos de 32 bits para el código y los datos. Todos los datos y el código (incluyendo los recursos del sistema) se encuentran en un solo segmento de 32 bits

Normalmente utilizamos el modelo de memoria pequeño, ya que mantiene todo el código dentro de un solo segmento, y todos los datos (incluyendo la pila) dentro de un solo segmento. Como resultado, sólo tenemos que manipular los desplazamientos de código y de datos, y los segmentos nunca cambian.