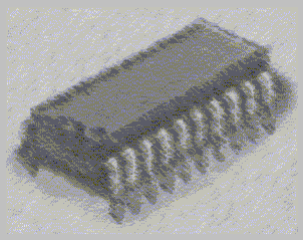
Tema 8. Directivas y programas complejos



Laboratorio de Estructura de Computadores

I. T. Informática de Gestión / Sistemas

Curso 2008-2009

Índice

- Instrucciones de manejo de cadenas
- Directivas
  - De definición de símbolos
  - De definición de datos
  - De definición de segmentos
  - De definición de procedimientos
  - De referencias externas
  - De definición de macros
  - De definición de bloques
  - De control del ensamblador
- Ensamblado de programas complejos

Instrucciones de manejo de cadenas (I)

- **Nombre:** MOVS/MOVSB/MOVSW (MOV BYTE/WORD STRING)
- **Formato:** MOVS destino, fuente
  MOVSB/MOSW
- **Descripción:** Transfiere el byte o la palabra apuntada por DS:SI al byte o palabra apuntada por ES:DI. Después de la transferencia, SI y DI se incrementan o decrementan, según el valor del indicador DF (DF=0 → incremento; DF=1 → decremento), el número de bytes transferidos. Para la repetición puede ir precedida por REP.
- **Operandos:** destino, fuente se emplean como referencia, pues por defecto MOVS emplea ES:DI como destino y DS:SI como fuente.

- **Ejemplos:**
  - movsb
  - movs tabla1, tabla2

Instrucciones de manejo de cadenas (II)

- **Nombre:** LODS/LODSB/LODSW (LOAD BYTE/WORD TO AL/AX)
- **Formato:** LODS fuente
  LODSB/LODSW
- **Descripción:** carga de un byte o de una palabra de una cadena. Transfiere un byte o una palabra de una cadena apuntada por DS:SI al acumulador AL o AX. Después incrementa o disminuye SI, según el estado del indicador DF, en 1 (byte) o en 2 (palabra). Para la repetición puede ir precedida por REP.
- **Operandos:** fuente se emplean como referencia, pues por defecto LODS emplea DS:SI como destino.

- **Ejemplos:**
  - lods cadena
  - lodsb

Instrucciones de manejo de cadenas (III)

- **Nombre:** STOS/STOSB/STOSW (STORE BYTE/WORD FROM AL/AX)
- **Formato:** STOS destino
  STOSB/STOSW
- **Descripción:** almacenamiento de un byte o de una palabra en una cadena. Transfiere un byte (AL) o una palabra (AX) a una cadena apuntada por ES:DI. Después de la ejecución se incrementa o disminuye DI, según el estado del indicador DF, en 1 (byte) o en 2 (palabra). Para la repetición puede ir precedida por REP.
- **Operandos:** destino se emplean como referencia, pues por defecto STOS emplea ES:DI como destino.
- **Ejemplos:**
  - stos cadena
  - stosw

Instrucciones de manejo de cadenas (IV)

- **Nombre:** CMPS/CMPSB/CMPSW (COMPARE BYTE/WORD)
- **Formato:** CMPS destino, fuente
  CMPSB/CMPSW
- **Descripción:** Comparación de bytes o palabras de cadenas. Actúan como CMP pero con cadenas apuntadas por DS:SI para la fuente y ES:DI para el destino. Después de la comparación, SI y DI son incrementados (DF=0) o decrementados (DF=1). Puede ir precedida por REP.
- **Operandos:** destino, fuente se emplean como referencia, pues por defecto CMPS emplea ES:DI como destino y DS:SI como fuente.
- **Ejemplos:**
  - cmps cadena1, cadena2
  - cmpsb
  - cmpsw

Instrucciones de manejo de cadenas (V)

- **Nombre:** SCAS/SCASB/SCASW (SCAN BYTE/WORD)
- **Formato:** SCAS destino
  SCASB/SCASW
- **Descripción:** análisis de un byte o de una palabra de una cadena. Ejecuta la substracción AL-destino para un byte, o AX-destino para una palabra. No almacena el resultado, pero posiciona los flags. Esto permite comparar los bytes o palabras sucesivas de una cadena. Puede ir precedida de REP para la repetición. ES:DI apunta a la cadena destino.
- **Operandos:** destino se emplean como referencia, pues por defecto SCAS emplea ES:DI como destino.
- **Ejemplos:**
  - scas cadena
  - sacsb

Instrucciones de manejo de cadenas (y VI)

- **Nombre:** REP/REPZ/REPE/REPNZ/REPNE (REPEAT)
- **Formato:** REP/REPZ/REPE/REPNZ/REPNE instrucción de cadena
- **Descripción:** Prefijo que impone la repetición de una instrucción de cadena mientras que el contenido del contador CX, disminuido en cada pasada, no llegue a cero. Esta repetición puede ser incondicional (REP) o condicional (REPZ/REPE o REPNZ/REPNE).
- **Operandos:** no tiene
- **Restricciones:** solo aplicable a instrucciones de cadena
- **Ejemplos:**
  - rep movsb ; rep como prefijo
  - rep ;primero rep
  - movs ; y luego instrucción de cadena

Las directivas (I)

- También llamadas pseudo-instrucciones o pseudo-operaciones
- No son como las sentencias “ejecutables” de los lenguajes de alto nivel, sino que sirven para reservar espacio de almacenamiento, asignar nombres a constantes, formar estructuras de datos, etc.
- No son órdenes destinadas al microprocesador, sino al programa ensamblador.
- Se pueden dividir en cuatro grupos funcionales:
  - Directivas de datos
  - Directivas condicionales
  - Directivas de listado
  - Directivas de macros

Las directivas (II)
Directivas de datos (I) Definir de símbolos (I)

- **Nombre:** EQU (EQUIVALENT)
- **Formato:** nombre EQU expresión
- **Descripción:** asigna un nombre simbólico al valor de una expresión. El “nombre” no puede redefinirse a lo largo del programa.
- **Ejemplo:**
  - columnas EQU 80

Las directivas (III)
Directivas de datos (II) Definir símbolos (y II)

- **Nombre:** =
- **Formato:** nombre = expresión
- **Descripción:** asigna un nombre simbólico al valor de una expresión. El "nombre" puede redefinirse a lo largo del módulo fuente. Útil en macros.
- **Ejemplos:**
  - valor = 10
  - valor = valor + 1

Las directivas (IV)
Directivas de datos (III) Definir datos (I)

- **Nombre:** DB (DEFINE BYTE)
- **Formato:** [nombre_variable] DB expresión
- **Descripción:** reserva memoria para una variable de tipo byte (8 bits) y los posteriores. "nombre_variable" es opcional y es el nombre asignado al primer byte.
- **Operandos:** "expresión" es el valor inicial de la variable y puede ser:
  - Una constante positiva o negativa o expresión de ellas (-128 ≤ expresión ≤127 con signo).
  - Un signo "?" que indica indefinición de valor.
  - Una cadena de caracteres delimitada por comillas simples o dobles.
  - n1 DUP (n2) que indica repetición n1 veces de la expresión n2.
- **Ejemplos:**
  - valores DB 30, -15, 20
  - DB 12*3
  - cadena DB "Hola mundo"

Las directivas (V)
Directivas de datos (IV) Definir datos (II)

- **Nombre:** DD (DEFINE DOUBLE)
- **Formato:** [nombre_variable] DD expresión
- **Descripción:** reserva memoria para una variable de tipo doble palabra (32 bits) y los posteriores. "nombre_variable" es opcional y es el nombre asignado a la primera doble palabra.

- **Operandos:** "expresión" es el valor inicial de la variable y puede ser:
  - Una constante positiva o negativa o expresión de ellas.
  - Un signo "?" que indica indefinición de valor.
  - Una dirección completa de memoria (segmento y desplazamiento).
  - n1 DUP (n2) que indica repetición n1 veces de la expresión n2.
- **Ejemplos:**
  - valores DD 300, -150, 2000
  - DD 120*3
  - direc DD tabla ; donde tabla equivale a su dirección
  - ;completa (segmento:desplazamiento)

Las directivas (VI)
Directivas de datos (V) Definir datos (III)

- **Nombre:** DQ (DEFINE QUADWORD)
- **Formato:** [nombre_variable] DQ expresión
- **Descripción:** reserva memoria para una variable de tipo cuádruple palabra (64 bits) y los posteriores. "nombre_variable" es opcional y es el nombre asignado a la primera cuádruple palabra.

- **Operandos:** "expresión" es el valor inicial de la variable y puede ser:
  - Una constante positiva o negativa o expresión de ellas .
  - Un signo "?" que indica indefinición de valor.
  - n1 DUP (n2) que indica repetición n1 veces de la expresión n2.

- **Ejemplos:**
  - valores DQ 300, -150, 2000
  - DQ 120*3
  - datos DQ 4 DUP (0) ; Equivale a DQ 0, 0, 0 , 0

Las directivas (VII)
Directivas de datos (VI) Definir datos (IV)

- **Nombre:** DT (DEFINE TENBYTE)
- **Formato:** [nombre_variable] DT expresión
- **Descripción:** reserva diez bytes de memoria para almacenar dígitos decimales empaquetados (dos dígitos por byte). El primer byte se reserva para el signo y los otros 9 para almacenar 18 dígitos decimales. El signo se almacena como 00h (si positivo) o 80h (si negativo). "nombre_variable" es opcional y es el nombre asignado al primer byte.

- **Operandos:** "expresión" es el valor inicial de la variable y puede ser:
  - Una constante positiva o negativa o expresión de ellas .
  - Un signo "?" que indica indefinición de valor.
  - n1 DUP (n2) que indica repetición n1 veces de la expresión n2.

- **Ejemplos:**
  - valores DT 0123456789
  - negat DT -0123456789

Las directivas (VIII)
Directivas de datos (VII) Definir datos (y V)

- **Nombre:** DW (DEFINE WORD)
- **Formato:** [nombre_variable] DW expresión
- **Descripción:** reserva memoria para una variable de tipo palabra (16 bits) inicializando o no esa palabra y las posteriores. "nombre_variable" es opcional y es el nombre asignado a la primera palabra.

- **Operandos:** "expresión" es el valor inicial de la variable y puede ser:
  - Una constante positiva o negativa o expresión de ellas.
  - Un signo "?" que indica indefinición de valor.
  - El desplazamiento de una variable
  - n1 DUP (n2) que indica repetición n1 veces de la expresión n2.

- **Ejemplos:**
  - valores DW 300, -150, 2000
  - DW 120*3

Las directivas (IX)
Directivas de datos (VIII) Referencias externas(I)

- **Nombre:** 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

Las directivas (X)
Directivas de datos (IX) Referencias externas(yII)

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

Las directivas (XI)
Directivas de datos (X) Definir segmentos (I)

- **Nombre:** SEGMENT
- **Formato:** nombre SEGMENT [alineamiento] [combinación]
- **Descripción:** indica el comienzo del segmento “nombre”. El final del segmento se indica mediante la directiva ENDS. El nombre de ambas directivas debe ser el mismo.

- **Alineamiento (opcional):**
  - BYTE
  - WORD
  - PARA
  - PAGE

- **Combinación(opcional):**
  - PUBLIC
  - COMMON
  - AT
  - STACK
  - MEMORY

- **Ejemplo:**
  - datos SEGMENT
  - datos ENDS

Las directivas (XII)
Directivas de datos (XI) Definir segmentos (II)

- **Nombre:** ENDS (END SEGMENT)
- **Formato:** nombre_segmento ENDS
- **Descripción:** indica el final del segmento “nombre_segmento” o bien indica el final de la estructura “nombre_estructura”.
- **Operandos:** nombre_segmento es obligatorio

Las directivas (XIII)
Directivas de datos (XII) Definir segmentos (y III)

- **Nombre:** ASSUME
- **Formato:** ASSUME registro_segmento:nombre_segmento
- **Descripción:** indica al ensamblador el registro de segmento que va a utilizar para direccionar cada segmento dentro del módulo.
- **Operandos:** "registro_segmento": DS, CS, SS o ES "nombre_segmento": generalmente el nombre asignado mediante la directiva SEGMENT.
- **Ejemplo:**
  - ASSUME CS:codigo

Las directivas (XIV)
Directivas de datos (XIII) Definir procedimientos(I)

- **Nombre:** 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

Las directivas (XV)
Directivas de datos (XIV) Definir procedimientos(II)

- **Nombre:** ENDP (END PROCEDURE)
- **Formato:** nombre_procedimiento ENDP
- **Descripción:** indica el final de un procedimiento
- **Operandos:** nombre_procedimiento es obligatorio

Las directivas (XVI)
Directivas de datos (XV) Definir macros (I)

- **Nombre:** 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

Las directivas (XVII)
Directivas de datos (XVI) Definir macros (II)

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

Las directivas (XVIII)
Directivas de datos (XVII) Definir macros (y III)

- **Nombre:** LOCAL
- **Formato:** LOCAL etiqueta
- **Descripción:** indica al ensamblador las etiquetas que debe cambiar cada vez que se expande la macro. Con ello se evita las definiciones múltiples de estas etiquetas. LOCAL sólo puede usarse dentro de la definición de una macro.
- **Ejemplo:**
  - retardo MACRO numero
  - local seguir
  - mov cx, numero
  - seguir: loop seguir
  - ENDM

Las directivas (XIX)
Directivas de datos (XVIII) Definir bloques

- **Nombre:** STRUC (STRUCTURE)
- **Formato:** nombre_estructura STRUC
- **Descripción:** define una plantilla de campos a nivel de bytes. Es una forma de definir nuevos tipos de datos. La definición de una estructura comienza con la directiva STRUC y acaba con ENDS. En la definición de la estructura no se reserva memoria. La forma de acceder a cada campo de la estrutura es: nombre_estructura.campo.
- **Operandos:** las directivas estándar de definición de datos son: DB, DW, DD, DQ y DT.

- **Ejemplo:**
  - parametros STRUC
  - p1 dw ?
  - p2 db ?
  - parametros ENDS

Las directivas (XX)
Directivas de datos (XIX) Control MASM (I)

- **Nombre:** END
- **Formato:** END [expresión]
- **Descripción:** indica el final del programa fuente. El operando “expresión” indica la dirección de comienzo del programa fuente. En el caso de varios módulos fuentes que se enlazan juntos para formar un solo programa ejecutable, sólo el módulo principal puede especificar expresión.
- **Ejemplo:** END inicio

Las directivas (XXI)
Directivas de datos (y XX) Control MASM (y II)

- **Nombre:** .RADIX
- **Formato:** .RADIX expresión
- **Descripción:** sirve para cambiar la base de numeración por defecto. Por defecto, los números sin sufijo se considera que están en base 10. Expresión está siempre en base 10.
- **Ejemplo:** .RADIX 16 ; base 16 (hexadecimal) por defecto

Las directivas (XXII)
Directivas condicionales

- Sirven para que el ensamblador incluya o ignore porciones de programa fuente según una condición en tiempo de ensamblaje.
  IFxxx [condición]
  ...
  ELSE
  ...
  ENDIF
- Puede incluirse o no la sentencia ELSE
- **Ejemplo:**
  - PRUEBA = 0
  - IF PRUEBA EQ 0
  - ... ; *Instrucciones que sólo se utilizan en pruebas*
  - ...
  - ENDIF

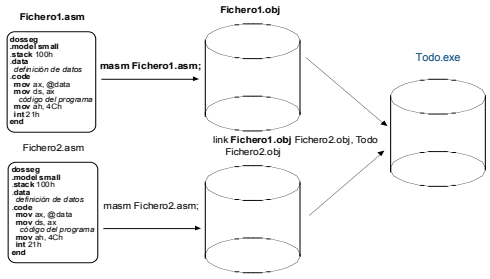
Las directivas (y XXIII)
Directivas de listado

- Indican al ensamblador la información a obtener en el listado de salida y el formato de esa información.
- Se clasifican a su vez en:
  - **De formato del listado:**
    - PAGE, TITLE, SUBTTL
  - **De listado de macros:**
    - .LALL: lista macros y expansiones
    - .SALL: suprime listado de macros y expansiones
  - **Comentarios:**
    - COMMENT
  - **Mensajes:**
    - %OUT: emite un mensaje por pantalla durante el ensamblado

Ensamblado de programas complejos (I)

- Para enlazar programas formados por diferentes módulos fuente, se deben emplear las directivas EXTRN y PUBLIC en los módulos para las etiquetas compartidas
- Una vez creados los módulos objetos la sintaxis del link será:

LINK [opciones] módulos_objeto, [ejecutable];



Ensamblado de programas complejos (y II)

Programa 1

```
Extrn texto2: byte
Extrn leer: far
Public variable
Dosseg
.model small
.stack 100h
.data
 texto db 'Pulsa tecla$'
 variable db ?
.code
  inicio: mov ax, @data
  mov ds, ax
  lea dx, texto
  mov ah, 9
```

Programa 1 (continuación)

```
int 21h
call leer
lea dx, texto2,
mov ah, 9
int 21h
Mov dl, variable
Mov ah, 2
Int 21ih
End inicio
```

Programa 2

```
Extrn variable: byte
Public texto2
Dosseg
.model small
.stack 100h
.data
 texto2db 'Has pulsado$'
.code
Leer proc
  mov ax, @data
  mov ds, ax
  mov ah, 1
  int 21h
  mov variable, al
```

Programa 2 (continuación)

```
ret
Leer endp
End
```

Para ensamblar el programa

Masm prg1;
Masm prg2
Link prg1 prg2, todo

Bibliografía

- 8088-8086/8087 programación ensamblador en entorno MS-DOS
  Miguel Angel Roselló.
  Ed. Anaya Multimedia
- Microprocesadores: el 8088 / 86
  Fernando Remiro Domínguez, Agustín Martín García
  Ed. Akal-Bibiloteca tecnológica
- Lenguajes ensambladores
  R. Martínez Tomás.
  Ed. Paraninfo
- Lenguaje ensamblador de los 80x86
  Jon Beltrán de Heredia
  Ed. Anaya-Multimedia. 1996