

## Tema 5: La pila, las macros y los procedimientos

---

- La pila
- Las macros
- Definición de procedimientos
- Tipos de procedimientos: *NEAR* y *FAR*
- Paso de parámetros a un procedimiento
- Mediante registros
- Mediante la pila
- Estructuras y paso de parámetros
- Macros y procedimientos

Estructura de Computadores

## Bibliografía básica

---

- 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-Biblioteca tecnológica
- Lenguajes ensambladores  
R. Martínez Tomás. Ed. Paraninfo
- Lenguaje ensamblador de los 80x86  
Jon Beltrán de Heredia  
Editorial Anaya-Multimedia. 1996



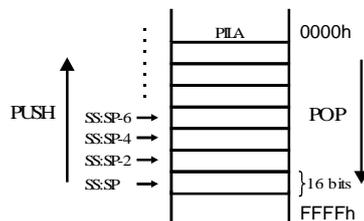
Área de Arquitectura y Tecnología de Computadores  
Departamento de Automática  
Universidad de Alcalá



Tema 5: La pila, las macros y los procedimientos  
Estructura de Computadores

## La pila

- Es una parte de la memoria encargada de gestionar de forma adecuada los procedimientos y las interrupciones
- Es una estructura LIFO (last in first out)
- La gestión se realiza a través del puntero de pila SP y de su registro de segmento asociado SS
- El tamaño de cada posiciones de 16 bits
- La pila crece hacia abajo (direcciones decrecientes de memoria) con PUSH
- La pila decrece hacia arriba (direcciones crecientes de memoria) con POP



## Macros

- Una macro es un conjunto de sentencias de ensamblador. Esta formada por:
  - Cabecera (el nombre de la macro y los parámetros)
  - Cuerpo (el conjunto de sentencias que definen el comportamiento de la macro)
  - Terminador
- Al llamar a una macro puede que no se pasen todos los parámetros, en cuyo caso se ignoran las sentencias que los referencian. Si se le pasan de más ignorará esos parámetros

### Definición de la macro

```
SUMAR MACRO xx, yy, zz  
    MOV AX, xx  
    ADD AX, yy  
    MOV zz, AX  
ENDM
```

### Llamada a la macro desde nuestro programa

```
SUMAR Num1, Num2, CX
```



## Instrucciones de manejo de bits (I)

- **Nombre:** TEST
- **Formato:** TEST destino, contador
- **Descripción:**
  - Es igual que la operación AND ( y lógico) pero no se guarda el resultado en destino
  - Se realiza a nivel de bit y modifica los flag de estado
- **Ejemplos:**
  - ; AX = 1234, BX = 0000
  - TEST AX, BX ; AX =1234, BX = 0000, Resultado = 0000
  - JZ Es\_Cero ; Salta si es cero a la posición Es\_Cero



## Instrucciones de manejo de bits (II)

- **Nombre:** CLI
- **Formato:** CLI
- **Descripción:**
  - Desactiva las interrupciones
  - Las interrupciones no enmascarables no se pueden inhibir
- **Ejemplos:**
  - CLI \_\_\_\_\_
- **Nombre:** STI
- **Formato:** STI
- **Descripción:**
  - Permite las interrupciones
- **Ejemplos:**
  - STI



## Instrucciones de manejo de bits (III)

---

- **Nombre:** CLC
  - **Formato:** CLC
  - **Descripción:**  
Desactiva el flag de acarreo
  - **Ejemplos:**  
CLC
- 

- **Nombre:** STC
- **Formato:** STC
- **Descripción:**  
Activa el flag de acarreo
- **Ejemplos:**  
STC



## Instrucciones de manejo de bits (IV)

---

- **Nombre:** CLD
  - **Formato:** CLD
  - **Descripción:**  
Pone a cero el flag de dirección (para trabajar con las instrucciones de cadenas)
  - **Ejemplos:**  
CLD
- 

- **Nombre:** STD
- **Formato:** STD
- **Descripción:**  
Pone a uno el flag de dirección (para trabajar con las instrucciones de cadenas)
- **Ejemplos:**  
STD



## Definición de procedimientos

- Los procedimientos son un conjunto de instrucciones que se pueden ejecutar desde muchos lugares diferentes de un programa, en vez de tener que repetir muchas veces el mismo conjunto de instrucciones cada vez que se necesitan
- Los procedimientos se llaman mediante la instrucción CALL
- La última instrucción de un procedimiento es la instrucción RET
- La dirección de retorno de un procedimiento (y en ciertos casos, procedimiento tipo FAR, el segmento al que pertenece) se almacena en la pila
- Para delimitar el conjunto de instrucciones que forman un procedimiento se emplean las palabras clave: PROC (comienzo del procedimiento) y ENDP (final del procedimiento)



## Tipos de procedimientos

- A los procedimientos se les asignan dos atributos **FAR** (si se le va a llamar desde otro segmento) y **NEAR** (si se le llama desde el mismo segmento)
- El procedimiento principal de un programa siempre es **FAR**

### Ejemplo de definición de un procedimiento

ImprimeTira	PROC	FAR	; es un procedimiento que llama
MOV AH, 09h			; a la función 9h del DOS para imprimir.
INT 21h			; una cadena de caracteres dadas en DS:DX
RET			; retorno de procedimiento
ImprimeTira		ENDP	



## Paso de parámetros mediante registros a un procedimiento

- Normalmente se emplean los registros para pasar parámetros a un procedimiento y para devolver el resultado del mismo
- Ya que tenemos un número limitado de registros, puede que se modifiquen los registros dentro de un procedimiento. En cuyo caso perderemos los valores que tenían antes de la llamada
- Por ello, se deben salvar en la pila todos los registros que se empleen en el procedimiento, como el primer grupo de instrucciones
- También se deben recuperar antes de volver del procedimiento los valores originales de los registros. DE TODOS MENOS DE AQUELLOS EN LOS QUE SE DEVUELVAN LOS RESULTADOS DEL PROCEDIMIENTO



## Paso de parámetros mediante la pila a un procedimiento NEAR (I)

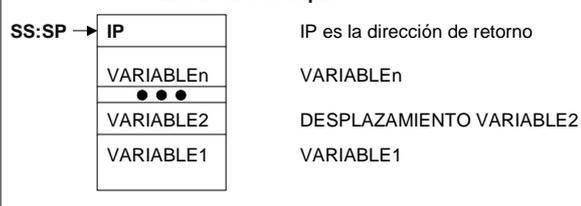
- Llamadas a procedimientos NEAR. Acciones del módulo llamador
- Los parámetros se encuentran dentro del mismo segmento. Indicaremos pues su desplazamiento (LEA) o la variable en sí

### Procedimiento llamador:

```
MOV AX, VARIABLE1
PUSH AX
LEA AX, VARIABLE2
PUSH AX

CALL PROCEDIMIENTO
```

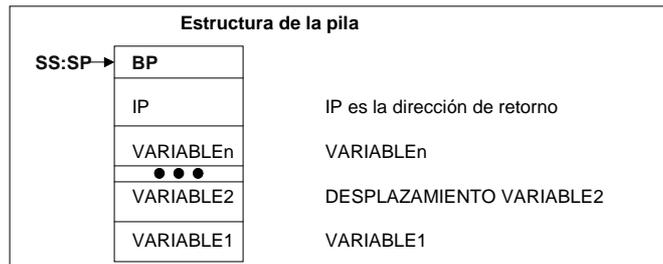
### Estructura de la pila



## Paso de parámetros mediante la pila a un procedimiento NEAR (II)

### Acciones del módulo llamado

- 1.- Almacenamos en la pila el registro BP, que es el que emplearemos para elegir el elemento que se desee de la pila (PUSH BP)



- 2.- Ponemos el valor del puntero de pila en BP (MOV BP, SP)
- 3.- Se recogen los parámetros a través de BP. En general el desplazamiento de un parámetro  $i$  será igual a:  $[BP] + 4 + 2 \times (n - i)$



## Paso de parámetros mediante la pila a un procedimiento NEAR (III)

### Acciones del módulo llamado

- Argumento de la instrucción RET desde el módulo llamado
- Indica el número de posiciones de la pila que se desean eliminar
- Será siempre el doble del número de parámetros. Si se le pasan 4 parámetros el argumento será 8. Es decir, RET 8



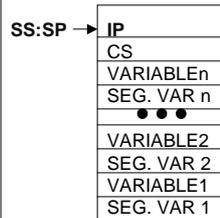
## Paso de parámetros mediante la pila a un procedimiento FAR (I)

- Llamadas a procedimientos NEAR. Acciones del módulo llamador
- Los parámetros se encuentran dentro de diferente segmento. Indicaremos pues su desplazamiento (LEA) y el segmento al que pertenece

### Procedimiento llamador:

```
LEA AX, VARIABLE1
PUSH DS
PUSH AX
LEA AX, VARIABLE2
PUSH DS
PUSH AX
CALL PROCEDIMIENTO
```

### Estructura de la pila



IP es la dirección de retorno  
 CS de la dirección de retorno  
 VARIABLEn  
 SEGMENTO VARIABLE n

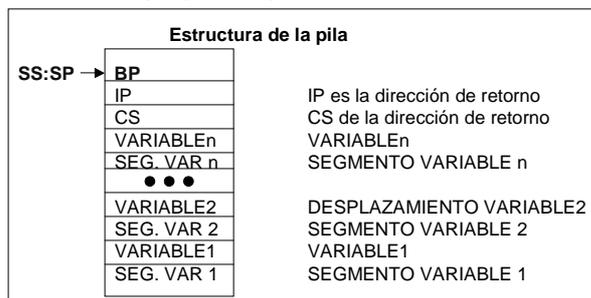
DESPLAZAMIENTO VARIABLE2  
 SEGMENTO VARIABLE 2  
 VARIABLE1  
 SEGMENTO VARIABLE 1



## Paso de parámetros mediante la pila a un procedimiento FAR (II)

### Acciones del módulo llamado

- 1.- Almacenamos en la pila el registro BP, que es el que emplearemos para elegir el elemento que se desee de la pila (PUSH BP)



- 2.- Ponemos el valor del puntero de pila en BP (MOV BP, SP)
- 3.- Se recogen los parámetros a través de BP. En general el desplazamiento de un parámetro  $i$  será igual a:  $[BP] + 6 + 4 \times (n - i)$  y el segmento al que pertenece:  $[BP] + 8 + 4 \times (n - i)$



## Paso de parámetros mediante la pila a un procedimiento FAR (III)

### Acciones del módulo llamado

- Argumento de la instrucción RET desde el módulo llamado
- Indica el número de posiciones de la pila que se desean eliminar
- Será siempre el cuádruple del número de parámetros. Si se le pasan 4 parámetros el argumento será 16. Es decir, RET 16



## Estructuras y paso de parámetros (I)

### Estructuras

- Las estructuras se definen con un identificador de comienzo de estructura STRUC y un delimitador de final ENDS
- Agrupan varios tipos de datos o campos bajo un mismo nombre
- Se puede acceder a cada campo mediante el nombre de la estructura *punto* nombre del campo
- Se suelen emplear para simplificar el paso de parámetros mediante la pila y el cálculo del argumento de RET

DNI STRUC	
Nombre_y_Apellidos	DB 50 DUP ("0")
Domicilio	DB 60 DUP ("0")
Edad	DB 0
Profesion	DB 15 DUP ("0")
ENDS	



## Estructuras y paso de parámetros (II)

### En el código

Estructura struc

```

bp0          DW ?
retorno     DW ?
p2          DW ?
P1          DW ?
NRET EQU OFFSET p1 - EQU OFFSET retorno
Estructura ends
    
```

### Campos de la estructura:

bp0	DW ⇔	2 bytes
retorno	DW ⇔	Equivale a un desplazamiento de +2 bytes (los del primer DW de la estructura) respecto al comienzo de la estructura
p2	DW ⇔	Equivale a un desplazamiento de +4 bytes (DW + retorno) respecto al comienzo de la estructura
p1	DW ⇔	Equivale a un desplazamiento de +6 bytes (DW+retorno+p2) respecto al comienzo de la estructura



## Estructuras y paso de parámetros (III)

### Acciones del módulo llamado

- 1.- Almacenamos en la pila el registro BP, que es el que emplearemos para elegir el elemento que se desee de la pila (PUSH BP)
- 2.- Ponemos el valor del puntero de pila en BP (MOV BP, SP)
- 3.- Se recogen los parámetros a través de la estructura y BP
- 4.- El argumento de RET será *nret*

### Ejemplo de la estructura definida antes

Estructura struc

```

bp0          DW ?
retorno     DW ?
p2          DW ?
P1          DW ?
NRET EQU OFFSET p1 - EQU OFFSET retorno
Estructura ends
    
```

### Módulo llamado

```

Prueba PROC NEAR
    PUSH BP
    MOV BP, SP
    PUSH AX
    MOV AX, [BP].p1
    ADD [BP].p1, DX
    POP AX
    POP BP
    RET NRET
Prueba ENDP
    
```



## Las macros y los procedimientos

---

Procedimientos	Macros
El código asociado es único	El código se repite cada vez que se llama que el ensamblador lo expande.
Más lentos	Más rápidas
Al finalizar el procedimiento se vuelve a la instrucción siguiente a la que la llamó	Después de la ejecución de la macro ejecuta la instrucción siguiente.
Menos flexible por con los parámetros	Más flexible con los parámetros

