

**LENGUAJE  
ENSAMBLADOR**

**SUBROUTINAS**

# SUBROUTINAS - CONCEPTO

- Es un programa auto-contenido
- Se puede invocar desde cualquier punto de un programa. En nuestro caso se utiliza la instrucción CALL
- Reutilización de código
- Programas modulares
- Requieren de comunicación para el pasaje de parámetros
- Dos tipos de pasaje de parámetros:
  - Por valor (por copia de una variable)
  - Pro referencia ( dirección de memoria de una variable).
- Las subrutinas tienen la desventaja de que reducen la velocidad de ejecución de los programas, esto se debe a la forma en que las subrutinas se ejecutan.



# SUBROUTINAS EJECUCIÓN

## • ¿Cómo se ejecuta una subrutina?

- 1.- Se encuentra la llamada Call
- 2.- El microprocesador almacena en la Pila el contenido del IP
- 3.- Se coloca en el IP el valor del desplazamiento correspondiente a la subrutina
- 4.- El microprocesador ejecuta las instrucciones de la subrutina
- 5.- La subrutina termina cuando se encuentra la instrucción Ret
- 6.- Se saca de la pila el valor original del IP y se continua el flujo del programa



# DEFINICIÓN DE UNA SUBROUTINA



Nombre Proc

...

...

...

Ret

Nombre Endp



Cuerpo del procedimiento

# LLAMADA A LA SUBRUTINA VIA CALL

En programa principal

...

Push Parametro 1

Push Parametro 2

Call Nombre

...

...



# IMPLEMENTACIÓN DEL PASAJE DE PARÁMETROS

- Vía registros
  - El número de registros es la principal limitación
  - Es importante documentar que registros se usan
- Vía memoria
  - Se usa un área definida de memoria (RAM).
  - Difícil de estandarizar

# IMPLEMENTACIÓN DEL PASAJE DE PARÁMETROS

- Vía pila (stack)
  - Es el método más ampliamente usado.
  - El verdadero "pasaje de parámetros".
  - Independiente de memoria y registros.
  - Hay que comprender bien como funciona porque la pila (stack) es usada por el usuario y por el sistema.

En x86, SP apunta al último lugar usado

# FUNCIONAMIENTO DE LA PILA

- El operando está (de forma implícita) en la cabeza de la pila
- Se requiere un registro Puntero de Pila (SP)
  - Contiene la dirección de la cabeza de la pila
- Operaciones sobre la pila
  - **PUSH** ; operación de Apilar
  - **POP** ; operación de Desapilar
  - Son inversas entre sí

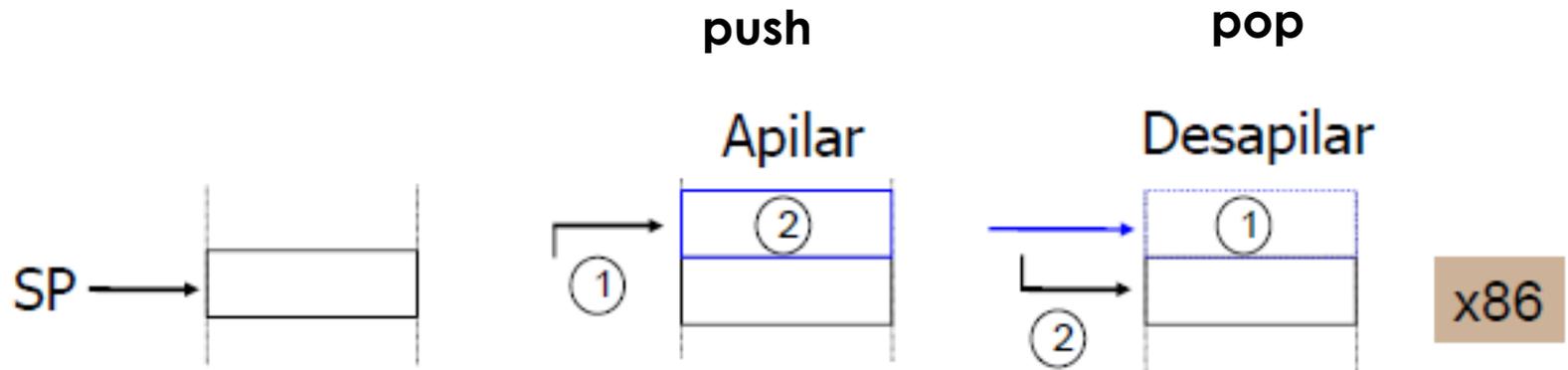


# OPERACIONES DE APILAR/DESAPILAR



- Secuencia de dos acciones:
  - 1- Movimiento de datos Reg-Mem ó Mem- Reg
  - 2- Modificación del puntero antes/después de la anterior
- Tener en cuenta:
  - dónde apunta el puntero
  - cómo crece la pila

# FUNCIONAMIENTO DE LA PILA MSX88



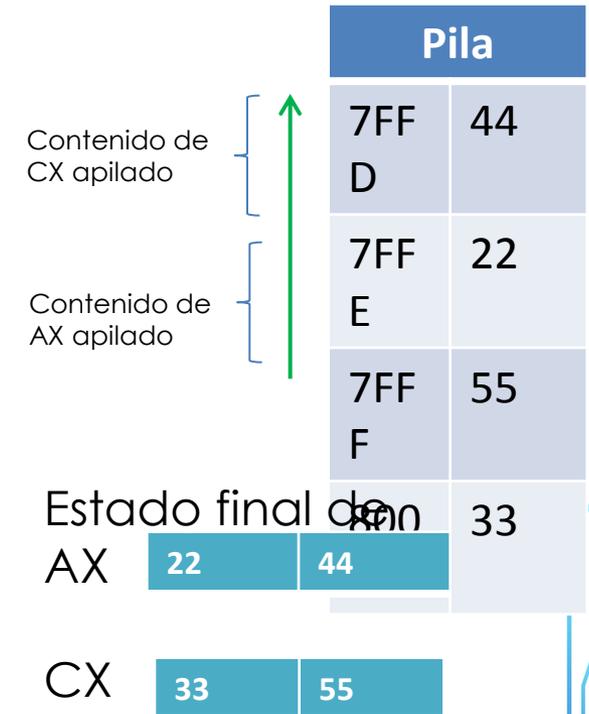
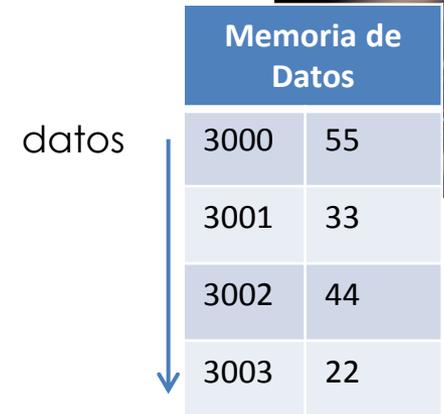
PUSH y POP realizan las operaciones de apilado y desapilado en la pila del procesador respectivamente, admiten todos los tipos de direccionamiento (excepto inmediato).

Los operandos deben ser siempre de 16 bits

# EJEMPLO DE PUSH Y POP

```

Ejemplo:   ORG   2000H
           MOV   BX, 3000H
           MOV   AX, [BX]
           PUSH  AX
           MOV   BX, 3002H
           MOV   CX, [BX]
           PUSH  CX
           POP   AX
           POP   CX
           HLT
           ORG   3000H
datos DB   55h, 33h, 44h, 22h
           END
    
```



# EJEMPLO CON SUBRUTINA

**ORG 1000H**

subrutina: NEG AX  
RET

**ORG 2000H**

MOV BX, 0  
MOV AX, dato  
PUSH AX  
CALL subrutina  
POP BX  
HLT

**ORG 3000H**

dato: DB 55H  
**END**

Analizar la pila y los valores  
finales de AX y BX

