

Procedimientos y Funciones

Procedimiento

- ▶ Cambia el flujo de ejecución de los programas.
- ▶ Encapsula código que se ejecuta y posteriormente se regresa a la siguiente línea de donde se ejecuto.



Nos ayuda a:

- ▶ Reutilizar código
- ▶ Reducir la cantidad de código
- ▶ La organización del programa
- ▶ Que el mantenimiento sea sencillo

Tipos de procedimiento:

- ▶ NEAR: es el procedimiento que está dentro del mismo segmento de código donde está la llamada,
- ▶ FAR: el procedimiento y la llamada no están en el mismo segmento de código.

Partes de los procedimientos

- ▶ Declaración del procedimiento
- ▶ Código del procedimiento
- ▶ Directiva de retorno
- ▶ Fin del procedimiento

Sintaxis:

- ▶ PROC: palabra reservada para la definición de procedimientos.
- ▶ ENDP: palabra reservada para indicar el final del procedimiento
- ▶ RET: palabra reservada para indicar al ensamblador el momento de salto al punto de llamada al procedimiento.
- ▶ CALL: palabra reservada para hacer la llamada a los procedimientos
 - ▶ CALL "Nombre del procedimiento"

Sintaxis:

“Nombre del procedimiento” proc NEAR | FAR

“Contenido del procedimiento”

RET

“Nombre del procedimiento” ENDP

Procedimiento1 PROC NEAR

MOV AX, BX

RET

Procedimiento1 ENDP

Cadenas de caracteres

Instrucciones

- ▶ LEA: transfiere la dirección efectiva es decir, el desplazamiento del operando fuente al destino.
- ▶ OFFSET: asigna el desplazamiento de un operando o variable.
- ▶ Con ambas, cargamos las direcciones de memoria:
- ▶ LEA destino, fuente: El operando fuente debe estar ubicado en memoria, y se coloca su desplazamiento en el registro índice o apuntador especificado en destino.

Ejemplo:

LEA DX,TEXTO ;Cargamos en DX la dirección efectiva del texto

Cadenas de Caracteres

► Asignar Valores:

LEA DX, Variable

MOV dx,OFFSET Variable

Indices

- ▶ SI
- ▶ DI