

mov dx, offset mens

call puts

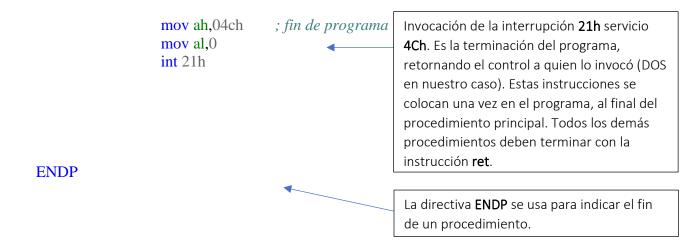
call getch

La directiva **offset** se usa para obtener la dirección en memoria (desplazamiento) de un símbolo definido en el programa. Puede ser una variable como en este caso, o un procedimiento.

Invocación del procedimiento externo **puts**. Este despliega en pantalla una cadena de caracteres direccionada por **DS:DX**. Imprime carácter por carácter hasta llegar a un 0 *crudo*.

Invocación del procedimiento externo **getch**. Este detiene la ejecución hasta que ocurra el evento de recibir un carácter del buffer del teclado. El carácter se almacena en **AL** y no se despliega en pantalla.

Coloquialmente se llama detener pantalla cuando se coloca un **getch** al final del programa y no se hace uso del carácter recibido.



```
; incluir procedimientos
; ejemplo:
; funcionX PROC ; < -- Indica a TASM el inicio de un procedimiento
; ;
; ; < --- contenido del procedimiento
; ret
; ENDP; < -- Indica a TASM el fin del procedimiento

La directiva END se usa para indicar el fin del programa.
```

## Desarrollando programas en ensamblador con TASM

## Generalidades:

- a) Por defecto, en **TASM** los números están en formato decimal. Para hexadecimales, se coloca una h al final del número.
- b) Todos los números deben iniciar con un dígito 0 9, incluyendo los hexadecimales. Si un número hexadecimal inicia con una letra, por ejemplo, FFh, se debe anteceder un 0, quedando 0FFh.
- c) Para reservar espacio en memoria para datos y variables, coloque en la sección DATA:
  - una etiqueta (nombre) del espacio en memoria,
  - el tamaño del dato,
  - el valor inicial.

## Ejemplos:

```
suma db 0; dato de 1 byte inicializado en 0. lado dw 8; dato de 2 bytes inicializado en 8. area dw ?; dato de 2 bytes no inicializado.
```

```
cadena db 32 dup(?); 32 posiciones de memoria de 1 byte reservadas, no inicializadas. promedios dw 10 dup(0); 10 posiciones de memoria de 2 bytes reservadas, todas inicializadas en 0. precios dw 451, 110, 56, 439, 1180, 900, 1270; 7 posiciones de memoria de 2 bytes reservadas, cada una inicializada.
```

Para trabajar con TASM y TLINK es necesario usar el emulador DOSBox. Realice los siguientes pasos:

- 1. Cree una carpeta de trabajo, por ejemplo, C:\OCLE
- 2. Descargue de Moodle los archivos **TASM**, **TLINK**, **PCLIB06**, **procs.inc** y **formato.asm** y guárdelos en la carpeta de trabajo.
- 3. Virtualice en **DOSBox** la carpeta, por medio de los comandos:

Mount X C:\OCLE

X:

Puede sustituir X por otra denominación de disco disponible en su sistema.

4. Ensamble la plantilla por medio de TASM, ingrese en DOSBox:

TASM formato.asm

Se generará el archivo:

- formato.obj: Traducción del código fuente.
- 5. Encadene la plantilla por medio de **TLINK**, ingrese en **DOSBox**:

TLINK formato.obj,,,PCLIB06.lib

Se generarán los archivos:

- formato.exe: Ejecutable del programa, contiene también los procedimientos externos.
- formato.map: Describe las direcciones lógicas donde se encuentran los segmentos del programa y sus tamaños.
- 6. Ejecute el programa, ingrese en **DOSBox**:

formato.exe

## Biblioteca de funciones PCLIB06

Contiene las siguientes funciones. En el archivo **procs.inc** se puede ver también la descripción de los procedimientos y sus entradas y salidas.

- 1. **clrscr**: Borra el contenido actual de la pantalla.
- 2. putchar: Despliega en pantalla el carácter ASCII en AL en la posición actual del cursor.
- 3. **puts**: Despliega en pantalla una cadena de caracteres ASCII terminada en 0 (null). El registro **DX** contiene el apuntador a la cadena a imprimir.
- 4. **getchar**: Recibe un carácter del teclado, el cual se retorna en **AL**. La ejecución se pausa en espera del carácter. Una vez recibido, se despliega en pantalla.
- 5. **getch**: Similar al anterior, pero el carácter no tiene eco en pantalla.
- 6. **gotoxy**: Coloca el cursor en la posición dada por **BX**, donde **BH** es la posición **X** (columna) y **BL** la posición **Y** (renglón).