

Manual Técnico

Detalles técnicos:

1. Practica.asm: en este archivo se tiene todo el procedimiento del programa en sí, es el que ejecuta los macros.
2. Practica.exe: es el ejecutable de DOS BOX para poder lanzar el programa.
3. Metodos.asm: contiene todos los macros que se ejecutan en practica.asm por lo cual es incluida en el programa.

Descripción de procedimiento:

El procedimiento es muy sencillo, por medio de macros en el archivo métodos.asm se imprime en consola un encabezado y un menú, luego nos pide elegir una opción mediante el número, si la opción se encuentra nos dirigirá a un apartado donde se imprime el encabezado de cada proceso y luego ejecuta una macro para ejecutar el procedimiento deseado mediante la opción elegida, en caso no se elija una opción aparente, el programa solo pintara de nuevo el encabezado y el menú.

Descripción de Macros:

1. **Print:** imprime el texto que se le mande.

```
print macro cadena
    mov ah, 09h
    lea dx, cadena
    int 21h
endm
```

2. **Terminar:** da por terminado el programa.

```
terminar macro
    mov ah, 4ch
    xor al, al
    int 21h
endm
```

3. **Getchar:** pide un carácter y lo guarda en al.

```
getChar macro
    mov ah, 01h
    int 21h
endm
```

4. **limpiarPantalla:** limpia la pantalla, lo que ya ha sido escrito en consola.

```
limpiarPantalla macro
    mov ax, 0600h
    mov bh, 0fh
    mov cx, 0000h
    mov dx, 184Fh
    int 10h
endm
```

5. **imprimirNumero:** imprime en pantalla un numero de dos dígitos.

```
imprimirNumero macro n1
    mov al, n1
    AAM
    mov bx, ax
    mov ah, 02h
    mov dl, bh
    add dl, 30h
    int 21h
    mov ah, 02h
    mov dl, bl
    add dl, 30h
    int 21h
endm
```

6. **numeroDoble:** Pide al usuario ingresar dos dígitos que serán tomados como un número, o puede poner un signo menos y después los dos dígitos y este también será tomado como un numero negativo.

```
numeroDoble macro u,de,n
LOCAL _INICIO, _FIN, _NEGATIVO
_INICIO:
    getChar
    cmp al,2DH
    je _NEGATIVO
    sub al,30h
    mov de, al
    getChar
    sub al,30h
    mov u, al
    mov al,de
    mov bl,10
    mul bl
    add al,u
    mov n, al
    jmp _FIN
_NEGATIVO:
    getChar
    sub al,30h
    mov de, al
    getChar
    sub al,30h
    mov u, al
    mov al,de
    mov bl,10
    mul bl
    add al,u
    mov n, al
    neg n
    jmp _FIN
_FIN:
endm
```

7. **factorial:** realiza un ciclo donde multiplica n y n-1 hasta llegar a n=1 para poder dar como resultado la factorial de un número, este solo puede ser de 1 a 4.

```
factorial macro msg1,msg2,n1, numFactorial
local _inicio,_Multiplicacion,_Resultado
_inicio:
    print msg1
    getChar
    sub al,30h
    mov cl, al
    jmp _Multiplicacion
_Multiplicacion:
    mov al,r
    mov bl,cl
    mul bl
    mov r, al
    loop _Multiplicacion
_Resultado:
    print msg2
    imprimirNumero r
    getChar
endm

arreglo macro vec,msg1
    add vec[si],offset msg1
    inc si
endm
```

8. **aritmética:** ejecuta un ciclo de pedir un numero y un operador hasta que se le da un ; para que del resultado después de calcular la operación.

```
aritmetica macro u,de,n1,n2,msg1,msg2,msg3,msg4
LOCAL _Inicio, _Operador, _Suma, _Resta, _Multiplicacion,_Division, _Resultado
_Inicio:
    print msg1
    numeroDoble u,de,n1
    jmp _Operador
_Operador:
    print msg2
    getChar
    cmp al,2BH
    je _Suma
    cmp al,2DH
    je _Resta
    cmp al,2AH
    je _Multiplicacion
    cmp al,2FH
    je _Division
    cmp al,3BH
    je _Resultado
_Suma:
    xor al,al
    print msg1
    numeroDoble u,de,n2
    mov al, n1
    add al, n2
    mov n1, al

    jmp _Operador
_Resta:
    xor al,al
    print msg1
    numeroDoble u,de,n2
    mov al, n1
    sub al, n2
    mov n1, al

    jmp _Operador
_Multiplicacion:
    xor al,al
    print msg1
    numeroDoble u,de,n2
    mov al, n1
    mov bl, n2
    imul bl
    mov n1,al
```

9. **crearArchivos:** crea un archivo con la extensión que se le provea, dando como resultado éxito o error si esta falla.

```
crearArchivo macro nombre,error,exito
LOCAL _exito,_Error,_FIN
    mov ah,3ch
    mov cx,0
    lea dx,offset nombre
    int 21h
    jc _Error
    jmp _exito
    _exito:
        mov bx,ax
        mov ah, 3eh
        int 21h
        print exito
        jmp _FIN
    _Error:
        print salto
        jmp _FIN
    _FIN:
endm
```

10. **editarArchivo:** crea el archivos con la extensión que se le provea para luego escribir sobre el lo que se le indique.

```
editarArchivo macro nombre,error,exito,vec
    crearArchivo nombre,error,exito
    mov ah,3dh
    mov al,1h
    mov dx, offset nombre
    int 21h
    mov bx,ax
    mov cx, offset vec
    mov dx,offset vec
    mov ah,40h
    int 21h
    print exito
    mov ah,3eh
    int 21h
endm
```