

PRACTICA 1:

Dado los siguientes códigos, ensamblar cada uno de los programas y comentarlos utilizando el MASM. Tomando como referencia para organizar el programa lo siguiente:

Normalmente podemos considerar que un programa en ensamblador está formado de las siguientes secciones:

- ✓ **Código de inicio**
- ✓ **Declaración de constantes simbólicas**
- ✓ **Variables del programa**
- ✓ **Código del programa**
- ✓ **Código de terminación**

OBJETIVOS:

- Uso y aprendizaje del lenguaje ensamblador
- Presentación del programa debug
- Observar el comportamiento de los registros internos del cpu
- Interpretar la secuencia de un programa con lenguaje ensamblador
- Documentar (comentar) un programa en lenguaje de bajo nivel

Programa 1:

```
.model smal  
  
.stack 100h  
  
.data  
  
message db "Hola, mundo!",0dh,0ah,'$'  
  
.code  
  
main proc  
  
mov dx, @data
```

```
mov ds, ax
mov ah,9
mov dx, offset message
int 21h
mov ax, 4C00h
int 21h
main endp
end main
```

2. CAPTURAR Y COMENTAR EL SIGUIENTE PROGRAMA:

```
;PROGRAMA CALCULADORA
```

```
FIN_MEN EQU '$'
```

```
NUEVE EQU 9
```

```
CR EQU 13
```

```
LF EQU 10
```

```
;----SEGMENTO DE PILA
```

```
PILA SEGMENT 'STACK'
```

```
DB 100 DUP("PILA")
```

```
PILA5 EQU THIS WORD
```

```
PILA ENDS
```

DATA SEGMENT 'DATA'

LETRERO DB ' OPERACION DE DOS DIGITOS'

DB CR

DB LF

DB LF

DB ' PROPORCIONA LOS OPERANDOS DE LA FORMA SIGUIENTE '

DB CR

DB LF

DB LF

DB ' OPERANDO1 (+,-,*,/) OPERANDO2 ', 0ah,0dh, '\$'

MENU DB ' MENU DE LA CALCULADORA '

DB CR

DB LF

DB LF

DB ' 1) OPERACION'

DB CR

DB LF

DB LF

DB ' 2) SALIR ',0AH,0DH,'\$'

DB CR

DB LF

DB LF

LONG DB 6

DB ?

CHAR DB 6 DUP (' ')

OP1 DB 0

OP2 DB 0

DATA ENDS

CODE SEGMENT 'CODE'

ASSUME CS:CODE ,DS:DATA ,SS:PILA

MAIN PROC FAR

MOV SP,OFFSET PILA5

MOV AX,DATA

MOV DS,AX

MOV AX,PILA

MOV SS,AX

MOV AH, 09 ;ESCRIBE MENSAJE

MOV DX, OFFSET MENU

INT 21H

LET1: MOV AH,1 ;INPUT A CHARACTER

INT 21H ;AL = CHARACTER

CMP AL,'1'

JE OPERANDO

CMP AL,'2'

JNE LET2

CMP AL,'2'

JE EXIT

OPERANDO: CALL OPER

SUM: CALL SUMA

RESTI: CALL RESTA

MULTI: CALL MULT

DIVID: CALL DIVIDE

LET2:

MOV AH, 09

MOV DX, OFFSET MENU

INT 21H

JMP LET1

OPER PROC NEAR

MOV AH, 09

MOV DX, OFFSET LETRERO

INT 21H

MOV AH,0AH ; SERVICIO PARA ENTRADA DE TECLADO

MOV DX,OFFSET LONG ;SE ESCRIBE LA CADENA DE CARACTERES.

INT 21H

MOV AH,02

MOV DL,0AH

INT 21H

MOV AH,02

MOV DL,0DH

INT 21H

CALL EMPAQUETA

EMPAQUETA PROC NEAR

MOV AL,0FH

AND CHAR ,AL

AND CHAR+1, AL

AND CHAR+3,AL

AND CHAR+4,AL

MOV AL,CHAR

MOV BL,10

MUL BL

ADD AL, CHAR+1

MOV OP1,AL

MOV AL,CHAR+3

MOV BL,10

MUL BL

ADD AL, CHAR+4

MOV OP2 , AL

EMPAQUETA ENDP

CMP CHAR+2,'+'

JE SUM

CMP CHAR+2,'-'

JE RESTI

CMP CHAR+2,'*'

JE MULTI

CMP CHAR+2,'/'

JE DIVID

OPER ENDP

;------

DESEMPAQUETA PROC NEAR

MOV AH,0

MOV BL,100

DIV BL

ADD AL,30H

PUSH AX

MOV AH,02

MOV DL,AL

INT 21H

POP AX

MOV AL,AH

MOV AH,0

MOV BL ,10

DIV BL

ADD AL,30H

PUSH AX

MOV AH,02

MOV DL,AL

INT 21H

POP AX

ADD AH,30H

MOV AL, AH

MOV AH,02

MOV DL,AL

INT 21H

JMP LET1

DESEMPAQUETA ENDP

SUMA PROC NEAR

MOV AL,[OP1]

ADD AL,[OP2]

CALL DESEMPAQUETA

RET

SUMA ENDP

RESTA PROC NEAR

MOV AL,[OP1]

SUB AL,[OP2]

CALL DESEMPAQUETA

RET

RESTA ENDP

MULT PROC NEAR

```
MOV AH,0H
MOV AL,[OP1]
MOV BL,[OP2]
MUL BX
CALL DESEMPAQUETAR_MUL
RET
```

MULT ENDP

DESEMPAQUETAR_MUL PROC NEAR

```
MOV BX,1000
PUSH AX
DIV BX
MOV CX,AX
MUL BX
MOV BX,AX
POP AX
SUB AX,BX
MOV BX,AX
ADD CX,30H
MOV AH,02
MOV DX,CX
```

INT 21H

MOV AX,BX

MOV BL,100

PUSH AX

DIV BL

MOV AH,0H

MOV CX,AX

MOV BH,0H

MUL BL

MOV BX,AX

POP AX

SUB AX,BX

MOV BX,AX

ADD CX,30H

MOV AH,02

MOV DX,CX

INT 21H

MOV AX,BX

MOV BL,10

PUSH AX

DIV BL

MOV AH,0H

```
MOV CX,AX
MOV BH,0H
MUL BL
MOV BX,AX
POP AX
SUB AX,BX
MOV BX,AX
ADD CX,30H
MOV AH,02
MOV DX,CX
INT 21H
ADD BX,30H
MOV AH,02
MOV DX,BX
INT 21H
JMP LET1
JMP EXIT
DESEMPAQUETAR_MUL ENDP
```

;------

```
DIVIDE PROC NEAR
```

```
MOV AX,0H
MOV AL,[OP1]
DIV [OP2]
```

```
CALL DESEMPAQUETA
RET
DIVIDE ENDP

;-----
EXIT:
    MOV AH,4CH
    INT 21H
MAIN ENDP

CODE ENDS
END MAIN
```

3.- Capturar el siguiente programa para practicar en el laboratorio

Title Programa en lenguaje ensamblador (practica.asm)

; Programa que imprime en pantalla datos obtenidos

; desde el teclado, mediante diversas llamadas a procedimientos

; de la librería.

.model small

.stack 100h

Blanco_azul = 1Fh

SaludoLoc = 0400h

```
.data
saludo db "Programa de practica 1"
db 0dh,0ah,0dh,0ah
db ""Como te llamas? ",0
numSelec db 0dh,0ah
db "Anota un entero de 16 bits con signo:",0
usuario db 50 dup(0)
pressAnyKey db 0dh,0ah,0dh,0ah,"Oprime cualquier tecla...",0

.code
extrn Clrscr:proc, Crlf:proc, Gotoxy:proc, \
Readint:proc, Readstring:proc, Scroll:proc, \
Readkey:proc, Writeint:proc, Writestring:proc
main proc
mov ax,@data
mov ds,ax
; limpia la pantalla, despliega una ventana azul.
call Clrscr
mov cx,0400h ; esquina superior izquierda
mov dx,0F2ch ; esquina inferior derecha
mov bh,Blanco_azul
call Scroll
; muestra el saludo y pregunta
; el nombre del usuario
mov dx,SaludoLoc
```

```
call Gotoxy
mov dx,offset saludo
call Writestring
mov dx,offset usuario
call Readstring
mov cx,5 ;cx es el contador del bucle
repite:
call Crlf
mov dx,offset usuario
call Writestring
loop repite ;salta a repite
; Pide al usuario anotar un numero entero decimal con signo.
; Rescribe el número en hexadecimal y en binario.
mov dx,offset numSelec
call Writestring
call Readint ; acepta un entero
call Crlf
mov bx,16 ; despliega en hexadecimal
call Writeint
call Crlf
mov bx,2 ; despliega en binario
call Writeint
mov dx,offset pressAnyKey
call Writestring
```

Laboratorio de Sistemas Digitales con Microprocesadores

Practica 1: Uso del MASM

M.C. Juan Carlos Olguín Rojas

```
call Readkey
```

```
call Clrscr
```

```
mov ax,4c00h ; fin de programa
```

```
int 21h
```

```
main endp
```

```
end main
```