

LABORATORIO 3c

Para el presente laboratorio se requiere Contar con Turbo Assembler, la utilidad de DOS debug y una guía de interrupciones .

El presente laboratorio es una compilación de los ejemplos de rutinas básicas que se presentan en parte teórica capítulo dedicado a Turbo Assembler:

Parte I:

1. Digitar el siguiente programa y compilarlo, luego haciendo tracing con turbo debugger asegúrate de entender lo que hace:

```
%TITLE "Demo: leer un caracter de teclado con interrupción DOS"

IDEAL
MODEL small
STACK 256

DATASEG
    excode      DB    0
    test1       DW    ?

CODESEG
inicio:
    mov ax, @data
    mov ds,ax

    mov ah,01h
    int 21h
    cmp al, 'Y'
    jne aqui
    mov bx, 9999h
    jmp endo

aqui:   cmp al, 'N'
        jne endo
        mov bx, 5555h

endo:   mov test1, bx

salir:
    mov ah, 04Ch
    mov al, [excode]
    int 21h

    END inicio
```

Parte II:

1. Digitar el siguiente programa y compilarlo, luego haciendo tracing con turbo debugger asegúrate de entender lo que hace:

```
%TITLE "Demo: leer una cadena de teclado y mostrarla con interrupción DOS"

IDEAL
MODEL small
STACK 256
```

```
DATASEG
    excode      DB      0
    cadena      DB      80 DUP(' '), '$'
    linea       DB      0Ah, 0Dh, '$'

CODESEG
inicio:
    mov ax, @data
    mov ds,ax

    lea bx, cadena
    mov ah, 01h
    mov cx, 00h
aqui:  int 21h
        cmp al, 0Dh      ; es salto de linea
        je nomas
        mov [cadena+bx], al
        cmp cx, 79
        je nomas
        inc cx
        inc bx
        jmp aqui

nomas: lea dx, linea
        mov ah, 09
        int 21h
        lea dx, cadena
        mov ah, 09
        int 21h

salir:
    mov ah, 04Ch
    mov al, [excode]
    int 21h

    END inicio
```

Parte III:

1. Digitar el siguiente programa y compilarlo, luego haciendo tracing con turbo debugger asegúrate de entender lo que hace:

```
%TITLE "Demo: Dibujar puntos en pantalla BIOS"

IDEAL
MODEL small
STACK 256

DATASEG
    excode      DB      0

CODESEG
inicio:
    mov ax, @data
    mov ds,ax

    mov ah, 00      ;modo 230*200
```

```
mov al, 04
int 10h
mov ah, 11      ;paleta de colores
mov bh, 00
mov bl, 01
int 10h
mov ah, 11
mov bh, 01
mov bl, 00
int 10h
mov al, 02
mov ah, 12
mov dx, 64h
mov cx, 9Eh
int 10h
mov ah,12
mov cx, 0A0h
int 10h
mov ah,12
mov cx, 0A2h
int 10h

salir:
mov ah, 04Ch
mov al, [excode]
int 21h

END inicio
```

Parte IV:

1. Digitar el siguiente programa y compilarlo, luego haciendo tracing con turbo debugger asegúrate de entender lo que hace:

```
%TITLE "Demo: Dibujar linea en pantalla BIOS"

IDEAL
MODEL small
STACK 256

DATASEG
    excode      DB    0

CODESEG
inicio:
    mov ax, @data
    mov ds,ax

    mov ah, 00      ;modo 640*200
    mov al, 06
    int 10h

    mov al, 01
    mov dx, 00
    mov cx, 00
repite:  mov ah, 12
    int 10h
    inc dx
```

```
inc cx
inc cx
inc cx
cmp dx, 0C8h
je salir
jmp repite

salir:
mov ah, 04Ch
mov al, [excode]
int 21h

END inicio
```

NOTA: Los programas presentados, hasta no estar seguros de su total funcionalidad, es decir, hayas corregido los errores de digitación es mejor probarlos con TURBO DEBUGGER.