SISTEMAS BASADOS EN MICROPROCESADORES (SBM)

Grado de Ingeniería Informática – Escuela Politécnica Superior – UAM

Parcial 2 - Curso 15-16

A continuación se incluye el código de una aplicación formada por un programa principal escrito en lenguaje C que llama a subrutinas o funciones escritas en ensamblador y de un programa residente escritos en lenguaje ensamblador del 8086. El programa residente implementa funciones de cálculo matemático que pueden ser llamadas desde cualquier aplicación utilizando una interrupción software e intercambiando información a través de registros de la CPU, tanto de entrada como de salida. En nuestro caso, llamaremos a alguna de esas funciones desde el programa principal en C utilizando funciones escritas en ensamblador que harán de interface con el programa residente.

Programa Principal

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>

extern int far  DetectarDriver ();
extern void far DesinstalarDriver ();
extern int far  Promediar (int* buffer, int numval);

void main(void) {
  int n = 7;
  int valores() = {5, 7, 15, 20, 25, 2, 4};
  long promedio;

  if (DetectarDriver()==1){
    printf("Driver no instalado.\n");
    exit(0);
  }
  else {
      promedio = Promediar (valores, n);
      Printf("El valor promedio es: %d\n", promedio);
      DesinstalarDriver();
  }
}
```

Rutinas en Ensamblador

```
_codigo_rutinas segment byte public
      assume cs:_codigo_rutinas

_DetectarDriver proc far
      push es
      xor ax,ax
      mov es,ax
      cmp word ptr es:[65h*4],0h
      jne _detectar_int
```

```
      cmp word ptr es:[65h*4+2],0h
      je _detectar_nodriver
_detectar_int:
      mov ah,00h
      int 65h
      cmp ax,0F0F0h
      jne _detectar_nodriver
      xor ax,ax
      jmp _detectar_fin
_detectar_nodriver:
      mov ax,1
_detectar_fin:
      pop es
      ret
_DetectarDriver endp

_DesinstalarDriver proc far
      mov ah,01h
      int 65h
      ret
_DesinstalarDriver endp

_Promediar proc far
      ..............
      ..............
      ..............
_Promediar endp


public _DetectarDriver
public _DesinstalarDriver
public _Promediar

_codigo_rutinas ends
end
```

Programa residente

```
code segment
      assume cs:code

      org 100h

driver_start:
      jmp instalar

;Variables del driver
      old_65h       dw 0,0

;Interfaz con el prog. residente
interfaz proc
      cmp ah,00h
      jne desinst
      call detectar
      jmp fin
desinst:
```

```
      cmp ah,01h
      jne promedio
      call desinstalar
      jmp fin
promedio:
      cmp ah,02h
      jne fin
      call promediar
fin:
      iret
interfaz endp


;Rutinas del programa residente

;Funcion que desinstala el driver
desinstalar proc
      push ax
      push es
      xor ax,ax
      mov es,ax

      cli

      ;Vector 65h
      mov ax,old_65h
      mov es:[65h*4],ax
      mov ax,old_65h+2
      mov es:[65h*4+2],ax

      sti

      mov es,cs:[2ch]
      mov ah,49h
      int 21h

      mov ax,cs
      mov es,ax
      mov ah,49h
      int 21h

      pop es
      pop ax
      ret
desinstalar endp

detectar proc
      ............
      ............
detectar endp

promediar proc
      ............
      ............
promediar endp
```

```
;Funcion que instala el driver
instalar proc
      xor ax,ax
      mov es,ax

      cli

      mov ax,es:[65h*4]
      mov old_65h,ax
      mov ax,es:[65h*4+2]
      mov old_65h+2,ax

      mov es:[65h*4],offset interfaz
      mov es:[65h*4+2],cs

      sti

      mov dx,offset instalar
      int 27h
instalar endp

code ends
end driver_start
```

SISTEMAS BASADOS EN MICROPROCESADORES (SBM)

Grado en Ingeniería Informática – Escuela Politécnica Superior – UAM

Parcial 2 - Curso 15-16

PREGUNTAS

NOMBRE: _____ **DNI :** _____
APELLIDOS: _____

P1. A la vista del código fuente y del enunciado del problema, escriba el código de la rutina _Promediar del programa principal, en lenguaje ensamblador del 8086, teniendo en cuenta que es una rutina interface con el programa residente y que la dirección del array de enteros será pasada al programa residente en BX (offset dirección) y CX (segmento dirección) junto con el número de valores en DX. El resultado del promedio será devuelto en AX desde el programa residente y tendrá que ser devuelto al programa principal en C. Justifique su respuesta. (3 p.)

```
_Promediar  proc far
            push   bp
            mov    bp, sp
            push   bx cx dx
            mov    bx, [bp+6]
            mov    cx, [bp+8]
            mov    dx, [bp+10]
            mov    ah, 02h
            int    65h
            pop    dx cx bx bp
            ret
_Promediar  endp
```

P2. De acuerdo con el código fuente del programa principal y del programa residente, escriba el código de la rutina *promediar* del programa residente. Tenga en cuenta que la dirección del array de datos enteros a promediar la recibe en BX (offset de la dirección) y CX (segmento de la dirección), y que él número de valores a promediar lo recibe en DX. El promedio de los valores, como suma de los mismos dividida por el número de valores, será devuelto en AX. Justifique su repuesta. (3 p)

```
promediar   proc
            push si es cx
            mov   es, cx          ;segmento de dirección del buffer
            mov   cx, dx          ;número de valores
            mov   ax, 0h
            mov   si, 0h
suma
            add   ax, es:[bx][si];ax = suma de valores
            add   si, 2h
            loop  suma            ;loop utiliza cx como contador
            div   dx              ; ax/dx=ax    ;promedio de valores
            pop   cx es si
            ret
promediar   endp
```

P3. Teniendo en cuenta el código fuente del enunciado, tanto el programa principal (código C y rutinas en ensamblador) así como el del programa residente, escribe el código en ensamblador de la rutina *detectar* del programa residente. Justifique su repuesta. (2 p.)

```
detectar    proc
            mov ax, 0F0Fh
            ret
detectar    endp
```

P4. En el programa residente, ¿cómo funciona la rutina *desinstalar (P4.1)*? ¿Qué información necesitamos pasarle desde el programa principal (*.exe) para que el programa residente sea eliminado de la memoria *(P4.2)*? ¿Porqué debe formar parte del programa residente y no del principal *(P4.3)*? (2 p.)

P4.1 Hace dos llamadas a un servicio de desinstalación (vector 21h y función 49h) pasándole información de dónde se encuentra el código residente a borrar (CS) y del segmento de variables de entorno (CS:[2ch]).

P4.2 Ninguna. Sólo es necesario llamar a la función del driver que realiza la desinstalación. Toda la información que necesita está en el código residente.

P4.3 La información que necesita el servicio de desinstalación vinculado a la interrupción 21h y función 49h sólo puede obtenerse desde el código del programa residente en tiempo de ejecución (dirección del segmento de código CS y dirección del segmento de las variables de entorno CS:[2Ch]).