

SISTEMAS BASADOS EN MICROPROCESADORES (SBM)

Grado de Ingeniería Informática
Doble Grado Informática-Matemáticas
Escuela Politécnica Superior – UAM

PARCIAL 3/FINAL MAYO - Curso 20-21

La siguiente aplicación está formada por el código de una **aplicación (.EXE) formada por un programa principal escrito en lenguaje C que llama a subrutinas o funciones escritas en ensamblador del 8086 (ASM) y de un programa residente escritos en lenguaje ensamblador del 8086 (ASM)**. El programa residente se utiliza para imprimir un carácter en una impresora conectada al puerto LPT1 utilizando el protocolo BUSY.

Programa Principal

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

extern int far DetectarDriver ();
extern void far DesinstalarDriver ();
extern int far Imprimir (char c);

void main(void) {
    if (DetectarDriver()==1){
        printf("Driver no instalado.\n");
        exit(0);
    }
    else {
        Imprimir('B');
        DesinstalarDriver();
    }
}
```

Rutinas en Ensamblador

```
_codigo_rutinas segment byte public
    assume cs:_codigo_rutinas

_DetectarDriver proc far
    mov ah,0
    int 64h
    cmp ax,ABCDh
    jne _detectar_nodriver
    xor ax,ax
    jmp _detectar_fin
_detectar_nodriver:
    mov ax,1
_detectar_fin:
    ret
_DetectarDriver endp

_DesinstalarDriver proc far
    mov ah,01h
    int 64h
```

```

        ret
_DesinstalarDriver endp

_Imprimir proc far
    push bp
    mov bp,sp
    mov ah,02h
    mov al,..... (1)
    int 64h
    pop bp
    ret
_Imprimir endp

public _DetectarDriver
public _DesinstalarDriver
public _Imprimir

_codigo_rutinas ends
end

```

Programa residente

```

code segment
    assume cs:code
    org 100h
driver_start:
    jmp instalar

;Interfaz con el programa residente
interfaz proc
func0:
    cmp ah,00h
    jne func1
    call detectar
    jmp fin
func1:
    cmp ah,01h
    jne func2
    call desinstalar
    jmp fin
func2:
    cmp ah,02h
    jne fin
    call imprimir
fin:
    iret
interfaz endp

;Rutinas del programa residente

;Funcion que desinstala el programa residente
desinstalar proc
    push ax
    push es
    xor ax,ax

```

```

        mov es,ax
        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
        ..... (2)
        .....
detectar endp

imprimir proc
        ..... (3)
        .....
imprimir endp

;Procedimiento de instalación del programa residente
instalar proc
        xor ax,ax
        mov es,ax

        cli
        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
Doble Grado Informática-Matemáticas
Escuela Politécnica Superior – UAM

PARCIAL 3 - Curso 20-21

PREGUNTAS

NOMBRE: _____ SOLUCION _____ DNI : _____
APELLIDOS: _____

P1. A la vista del código fuente y del enunciado del problema, escriba el código que falta (1) en la rutina `_Imprimir` del programa principal, en lenguaje ensamblador del 8086, para conseguir que el driver imprima el carácter pasado como parámetro en dicha función. Dibuje la situación de la pila (datos en la misma) en el momento en que la CPU va a ejecutar el código de la línea (1). Justifique su respuesta. (2 p.)

BP = SP →			
[BP+2] →	BP		
[BP+4] →	Dir. Retorno (offset)	← [BP+3]	
[BP+6] →	Dir. Retorno (segmento)	← [BP+5]	
SP ini →	'B'	← [BP+7]	

(*) **SP ini** es la posición del puntero de pila (SP) antes de que se guarde en la pila el parámetro de la rutina `_Imprimir` del programa principal. Ese parámetro es el código ASCII del carácter a imprimir.

`mov al, [bp+6] (1)`

P2. De acuerdo con el código fuente del programa principal y del programa residente, escriba el código de la rutina *imprimir* del programa residente (3). Tenga en cuenta que el código ASCII del carácter a imprimir se pasa a través del registro AL y que se utiliza el protocolo BUSY para gestionar la impresión con la impresora conectada al puerto LPT1. Justifique su respuesta. (5 p)

```

imprimir proc
    push bx dx es
    mov bx, 0
    mov es, bx
    mov dx, es:[0408h]                ;obtenemos dirección base del LPT1

espera_libre:
    add dx, 1                        ;apuntamos al reg. estado de LPT1
ocupada:
    in bl, dx                        ;leemos reg. Estado
    test bl, 10000000b              ;comprueba bit #BUSY
    jz ocupada                       ;impresora ocupada si #BUSY=0
                                     ;impresora ya libre si salimos del bucle
    sub dx, 1                        ;apuntamos al reg. Datos
    out dx, al                       ;escribimos el código ASCII carácter a
                                     ;imprimir en reg, datos, pasado en AL
    add dx, 2                        ;apuntamos al reg, control
    in bl, dx                        ;leemos el reg. Control
    or bl, 00000001b                ;ponemos bit STROBE=1
    out dx, bl

espera_ocupada:
    sub dx, 1                        ;apuntamos al reg, estado
libre:
    in bl, dx                        ;leemos reg. Estado
    test bl, 10000000b              ;comprobamos bit #BUSY
    jnz libre                        ;impresora libre cuando #BUSY=1
    add dx, 1                        ;apuntamos al reg. Control
    and bl, 11111110b
    out dx, bl                       ;ponemos STROBE=0

    pop es dx bx
    ret
imprimir 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 (2). Justifique su respuesta. (2 p.)

```
detectar poc
    mov ax, ABCDh
    ret
detectar endp
```

P4. Analizando el código del enunciado (1 p.):

P4.1 Explique cómo funciona la rutina *desinstalar* del programa residente (0.4 p)

Utilizando un servicio vinculado a la interrupción 21h asociado a la función 49h de su rutina de servicio. Este servicio debe ser llamado 2 veces, una con la dirección (segmento) donde se encuentra el driver y otra con la dirección (segmento) donde se encuentran las variables de entorno y que se obtiene del PSP.

P4.2 Indique la información que dicha rutina necesita que le sea pasada desde el programa principal (.exe) para efectuar la desinstalación (0.3 p)

Ninguna información, sólo llamar a la función de desinstalación (ah = 01h) del driver haciendo uso del mecanismo de la interrupción software con el vector 64h (int 64h).

P4.3 ¿Podría ser una función del programa principal (.exe) sin necesidad de formar parte del programa residente? (0.3 p)

En principio no, la información (direcciones de los segmentos de variables del entorno y del propio driver) se conocen dentro del driver.