Escuela Politécnica Superior

UAM
Universidad Autónoma de Madrid

Laboratorio de Sistemas Basados en Microprocesadores

Práctica 4: Diseño de programas residentes.

Diseñar un programa residente que se ejecute mediante la INT 60h y que proporcione un primer servicio (con AH=12h) de conversión de un entero representado como cadena de caracteres de dígitos decimales (ej.: "65535") al entero correspondiente representado como cadena de dígitos hexadecimales (ej.: "FFFF"), y un segundo servicio (con AH=13h) que haga la conversión inversa: de hexadecimal a decimal.

Las direcciones de las cadenas se pasarán a la RSI en DS:BX y volverán convertidas en DS:CX. Ambas cadenas terminarán en '$'. Se supone que los números hexadecimales podrán tener 4 dígitos como máximo (ej.: "FFFF") y que los enteros son **SIN** signo.

Escuela Politécnica Superior

UAM
Universidad Autónoma de Madrid

Interrupciones

INT: ejecuta la rutina de servicio a la interrupción indicada por el número. INT número

IRET: retorno de la rutina de servicio.

Las interrupciones son llamadas a rutinas del sistema (normalmente servicios del BIOS o del SO).

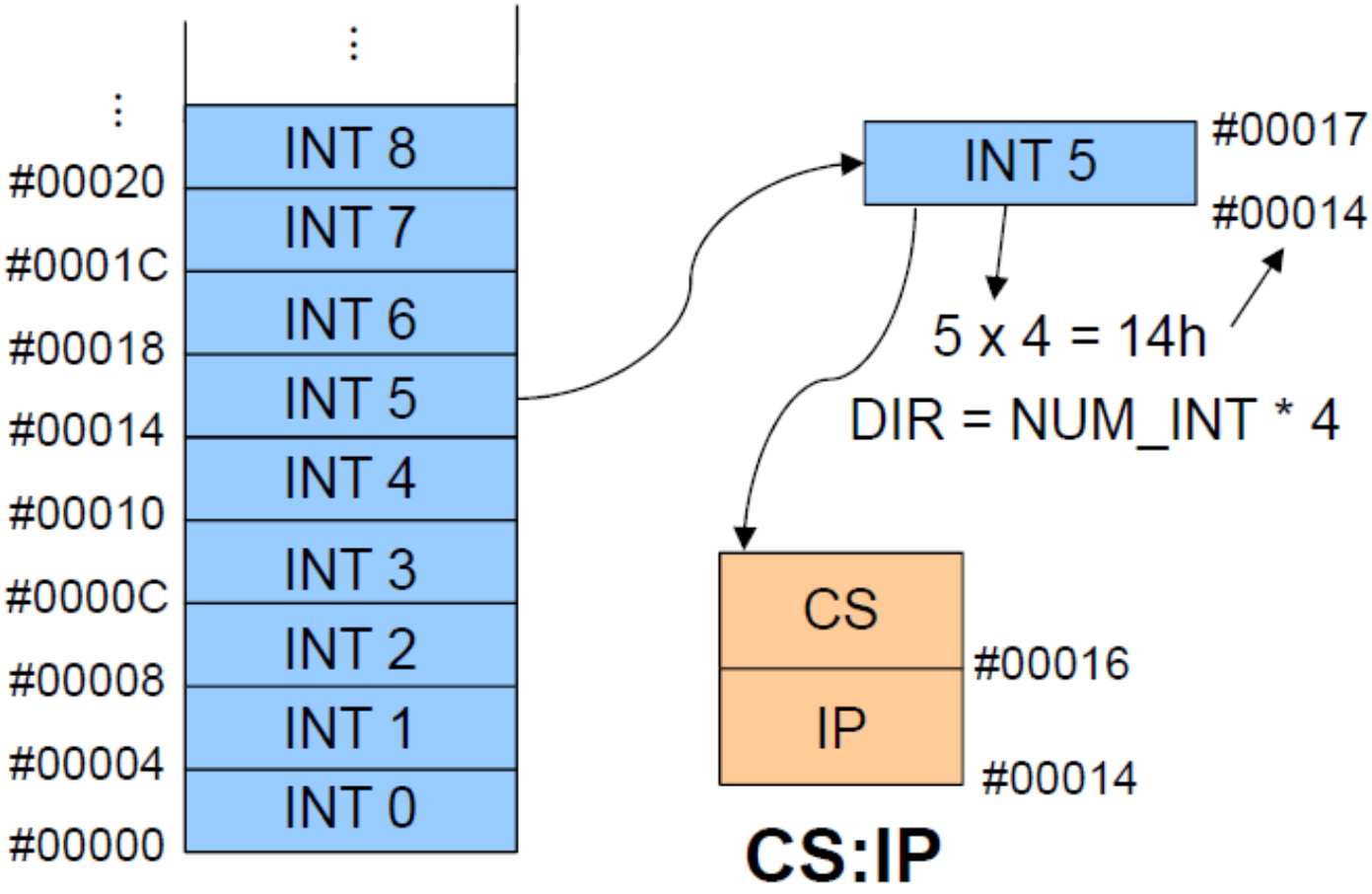
Estas rutinas están “residentes” en memoria.

Las posiciones de memoria donde empiezan las rutinas se guardan en una tabla en memoria. Esta tabla se encuentra al principio de la memoria en DOS: desde la dirección 0 a la 3FFh.

Escuela Politécnica Superior

UAM
Universidad Autónoma de Madrid

Cada 4 bytes de esta tabla constituyen un vector de interrupción (offset y segmento donde comienza la rutina de servicio a esa interrupción).



Escuela Politécnica Superior

Instalación de una rutina de servicio a interrupción:

UAM Universidad Autónoma de Madrid

```
DIR equ 4 * NUM_INT
mov ax, 0
mov es, ax
cli
mov es:[ DIR ], OFFSET rutina_servicio
mov es:[ DIR + 2 ], SEG rutina_servicio
sti
```

Fases de ejecución de una interrupción por la CPU:

1. Se apilan banderas y dirección de retorno.
2. Se ponen a 0 bit de interrupción **IF** y de traza **TF** (enmascarando interrupciones hardware y desactivando ejecución paso a paso).
3. Se lee vector de interrupción (**CS**:**IP**) con dirección de primera instrucción de la rutina de servicio.
4. Se ejecuta la rutina de servicio.
5. La rutina de servicio acaba con instrucción **IRET**.
6. Se desapilan dirección de retorno y estado.

Escuela Politécnica Superior

UAM
Universidad Autónoma de Madrid

Tres tipos de ficheros ejecutables en DOS:
.BAT comandos del DOS (no código máquina)

.EXE
Son programas en código máquina.
Generados por un montador (*linker*) a partir de uno o varios ficheros de código objeto generados por un compilador o ensamblador.

.COM
Son programas en código máquina.
El programa ocupa un único segmento físico de 64 KB con código, datos y pila.

La primera instrucción ejecutable está en la dirección 256 (100h) respecto al origen del segmento. Se debe usar la directiva **ORG** 256 antes de la primera instrucción de ensamblador.
Se crean a partir de un .EXE con el comando **EXE2BIN** o directamente con la opción **/t** del montador (TLINK).

Escuela Politécnica Superior

UAM
Universidad Autónoma de Madrid

Ejecución de programas **.EXE**:

CS y **SS** inicializados por el DOS. **DS** y **ES** apuntan al PSP.
IP inicializado con dirección indicada en directiva **END**.
SP inicializado con valor más alto del segmento de pila.
Al acabar el programa se devuelve el control al sistema operativo (intérprete de comandos) y se libera la zona de memoria donde se cargó el programa.

Ejecución de programas **.COM**:

CS, **DS**, **ES** y **SS** apuntan al PSP.
IP se inicializa a 256 (posición siguiente al PSP).
SP se inicializa con 0FFFEh.
Al acabar el programa se devuelve el control al sistema operativo (intérprete de comandos) y se libera la zona de memoria donde se cargó el programa.

Escuela Politécnica Superior

UAM
Universidad Autónoma de Madrid

PSP (Prefijo de Segmento de Programa)

Zona de datos de 256 bytes que encabeza los programas .EXE o .COM una vez están cargados en memoria RAM para su ejecución.

Generada por el DOS mediante el intérprete de comandos (COMMAND.COM).

Campos más destacados del PSP

Offsets 2Ch y 2Dh (2 bytes)

Número de segmento físico que contiene una copìa de las variables de entorno del DOS. Permite al programa acceder a esas variables.

Offset 80h (1 byte)

Tamaño en bytes de los parámetros del programa en línea de comandos.

Offsets 81h a FFh (127 bytes)

Códigos ASCII de los parámetros del programa en línea de comandos. Acaba con código 13 (retorno de carro). Permite al programa acceder a los parámetros indicados por línea de comandos.

Escuela Politécnica Superior

Ejemplo

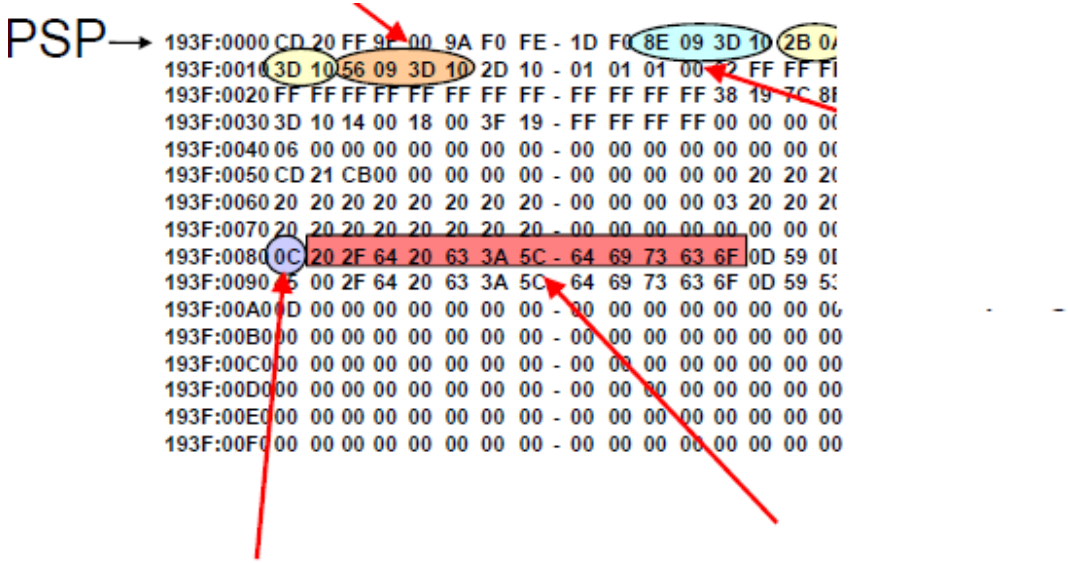
- Dadas las siguientes variables de entorno (comando **SET** de DOS):

```
COMSPEC=C:\DOS60\COMMAND.COM
PROMPT=$P$G
TEMP=C:\TEMP
PATH=C:\TD;C:\TASM
```

- Si se ejecuta el programa PROGRAMA con los parámetros /D y C:\DISCO:

```
C:\> PROGRAMA /D C:\DISCO
```

El PSP tendría la siguiente forma:



Escuela Politécnica Superior

UAM
Universidad Autónoma de Madrid

PSP →

```
193F:0000 CD 20 FF 9F 00 9A F0 FE - 1D F0 8E 09 3D 10 2B 0A
193F:0010 3D 10 56 09 3D 10 2D 10 - 01 01 01 00 02 FF FF FF
193F:0020 FF FF FF FF FF FF FF FF - FF FF FF FF 38 19 7C 8F
193F:0030 3D 10 14 00 18 00 3F 19 - FF FF FF FF 00 00 08 00
193F:0040 06 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00
193F:0050 CD 21 CB00 00 00 00 00 - 00 00 00 00 00 20 20 20
193F:0060 20 20 20 20 20 20 20 20 - 00 00 00 00 03 20 20 20
193F:0070 20 20 20 20 20 20 20 20 - 00 00 00 00 00 00 00 00
193F:0080 0C 20 2F 64 20 63 3A 5C - 64 69 73 63 6F 0D 59 0D
193F:0090 45 00 2F 64 20 63 3A 5C - 64 69 73 63 6F 0D 59 53
193F:00A0 0D 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00
193F:00B0 00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00
193F:00C0 00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00
193F:00D0 00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00
193F:00E0 00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00
193F:00F0 00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00
```

Número de segmento con copia de variables de entorno del DOS: 1938h

```
1938:0000 43 4F 4D 53 50 45 43 3D - 43 3A 5C 44 4F 53 36 30
1938:0010 5C 43 4F 4D 4D 41 4E 44 - 2E 43 4F 4D 00 50 52 4F
1938:0020 4D 50 54 3D 24 70 24 67 - 00 54 45 4D 50 3D 43 3A
1938:0030 5C 54 45 4D 50 00 50 41 - 54 48 3D 43 3A 5C 54 44
1938:0040 3B 43 3A 5C 54 41 53 4D - 00 00 01 00 43 3A 5C 41
```

```
COMSPEC=C:\DOS60
\COMMAND.COM.PRO
MPT=$P$G.TEMP=C:
\TEMP.PATH=C:\TD
;C:\TASM....C:\A
```

Escuela Politécnica Superior

UAM
Universidad Autónoma de Madrid

Programas residentes (*Terminate & Stay Resident, TSR)*

Programas .COM o .EXE que terminan su ejecución dejando sin liberar parte de la memoria que ocupan.
Su posición en memoria suele almacenarse en forma de vector de interrupción.
Pueden ser llamados desde otros programas en ejecución o desde rutinas de servicio de interrupción.

Programas residentes .COM (*instalación*)

Finalizan con **INT** 27h.
DX debe contener el *offset* de la posición siguiente a la última que se quiere dejar residente.

Constan de dos partes:

La información (código, variables, …) que queda residente.
El código que instala la información.

Escuela Politécnica Superior

Universidad Autónoma de Madrid

Programas residentes .COM (*desinstalación*)

Ha de ejecutarse un programa o rutina (desinstalador) que libere la memoria que se dejó residente.
Se libera un segmento físico de memoria mediante **INT** 21h con **AH**=49h y **ES**=número de segmento.

Se deben liberar dos segmentos físicos:

Segmento de código del programa residente (suele guardarse en algún vector de interrupción).
Segmento de variables de entorno (offset 2Ch del PSP).

Antes de liberar un programa es conveniente comprobar que está realmente instalado:

Vector de interrupción distinto de cero
Primeros bytes de la rutina de servicio son los del programa que se desea desinstalar (firma digital del programa).

```
codigo SEGMENT
        ASSUME cs : codigo

        ORG 256

inicio: jmp instalador

; Variables globales
tabla   DB  "abcdf "
flag    DW 0

; Rutina de servicio a la interrupción
rsi PROC FAR
        ; Salva registros modificados
        push ...
        ; Instrucciones de la rutina
        ...

        ; Recupera registros modificados
        pop ...
        iret
rsi ENDP
...
```

```
...
instalador PROC
        mov ax, 0
        mov es, ax
        mov ax, OFFSET rsi
        mov bx, cs
        cli
        mov es:[ 40h*4 ], ax
        mov es:[ 40h*4+2 ], bx
        sti
        mov dx, OFFSET instalador
        int 27h ;  Acaba y deja residente
                ;  PSP, variables y rutina rsi.
instalador ENDP

codigo ENDS
END inicio
```

```
desinstalar_40h PROC        ; Desinstala RSI de INT 40h
    push ax bx cx ds es

    mov cx, 0
    mov ds, cx                  ;  Segmento de vectores interrupción
    mov es, ds:[ 40h*4+2 ]      ;  Lee segmento de RSI
    mov bx, es:[ 2Ch ]          ;  Lee segmento de entorno del PSP de RSI

    mov ah, 49h
    int 21h         ;  Libera segmento de RSI (es)
    mov es, bx
    int 21h         ;  Libera segmento de variables de entorno de RSI

    ;  Pone a cero vector de interrupción 40h
    cli
    mov ds:[ 40h*4 ], cx        ;  cx = 0
    mov ds:[ 40h*4+2 ], cx
    sti

    pop es ds cx bx ax
    ret
desinstalar_40h ENDP
```

Escuela Politécnica Superior

Detalles de la P4

UAM
Universidad Autónoma de Madrid

- **Programa 1: p4a.asm (3 puntos)**

- Programa verifique estado y haga instalador / desinstalador .
- Además debe contener la RSI a instalar en la int 60h

- Verificar el estado:
  - Extraer del PSP la informacion. (/I o /D)
  - En funcion de la información, instalar / desinstalar o dar estado.

- Escribir en ensamblador un programa **(.COM).**

Escuela Politécnica Superior

Detalles de la P4

UAM
Universidad Autónoma de Madrid

- **Programa 2: p4b.asm (4 puntos)**

- Desarrollar un programa para probar la rutina int 60h.
- Programa .EXE
- Datos a convertir pueden estar en memoria o introducirse desde el teclado.

- Importante:

  La RSI no se puede depurar en el TD. El bit de traza esta inhabiiltado.

  Para depurar la RSI, se aconseja desarrollar la rutina como una función y hacer la llamada con call.

Escuela Politécnica Superior

Detalles de la P4

UAM
Universidad Autónoma de Madrid

- **Programa 3: p4c.asm (3 puntos)**

Realizar un programa que, dado un entero introducido desde el teclado, escriba en pantalla cada uno de sus dígitos convertidos a decimal o hexadecimal utilizando la interrupción 60h definida en el apartado anterior, a un ritmo de un carácter por segundo aproximadamente, utilizando la interrupción 1Ch para la temporización.

- La interrupción periódica (1Ch)

El PC realiza una petición de interrupción 18,2 veces por segundo. (Int hardware 08)

La rutina de atención, además de ciertas tareas de control y mantenimiento, realiza una llamada a la interrupción software 1Ch. El único contenido RSI es IRET.

Reinstalando los vectores de esta interrupción el programador puede realizar tareas periódicas sin más modificaciones.

Se recomienda **actualizar el programa p4a.asm** incluyendo la RSI de la INT 1Ch para ser utilizado por el programa p4c.asm

- Puertos E/S

Evitar interrupciones durante la ejecución de la rutina: inhibir las interrupciones. NO usar CLI / STI.

Aplicar una inhibición/ desinhibición selectiva del TIMER modificando el bit 0 del registro de máscara (IMR) del controlador. Mediante IN / OUT

Escuela Politécnica Superior

Puertos E/S

UAM
Universidad Autónoma de Madrid

Lectura de puertos de E/S:

- IN: realiza una lectura del puerto .
  - IN AL, 21h ; vuelca en AL el contenido del puerto 21h.

Escritura de puertos E/S:

- OUT: Escribe en el puerto Xh el valor que se desea escribir.
  - OUT 21h, AL ; vuelca el contenido de AL en el puerto 21h.