

(4)



Unidad 4

Recursos de Programación

SISTEMAS BASADOS EN MICROPROCESADORES

**Grado en Ingeniería Informática
EPS - UAM**

(4)

Índice

4. Recursos de programación.

- 4.1. Interrupciones BIOS.
- 4.2. Interrupciones DOS.
- 4.3. Ejecución de programas desde el DOS.
- 4.4. PSP (Prefijo de Segmento de Programa).
- 4.5. Tipos de programas: EXE, COM y residentes (TSR).

(4)

4.1. Interrupciones BIOS (I)

- **BIOS** (*Basic Input/Output System*): Es el *firmware* básico instalado en la placa base.
- Proporciona rutinas básicas de acceso al *hardware*.
- Pueden dividirse en cinco grupos diferenciados:
 - Interrupciones asociadas a la CPU (INT 0 a INT 7)
 - Interrupciones asociadas al controlador de interrupciones 8259 (INT 8 a INT 0Fh)
 - Servicios del BIOS (INT 10h a INT 1Ah e INT 40h)
 - Rutinas de usuario (INT 1Bh e INT 1Ch)
 - Punteros a tablas de datos (INT 1Dh a INT 1Fh e INT 41h)
- Lista de interrupciones de Ralf Brown:
<http://www.ctyme.com/rbrown.htm>

(4)

4.1. Interrupciones BIOS (II)

Asociadas a la CPU

- **INT 0: División por cero**
 - Generada por la CPU cuando el cociente de una división (**DIV** o **IDIV**) es demasiado grande para ser almacenado en **AL** o **AX**.
 - Imprime en la consola "***Divide overflow***" y retorna al DOS.
- **INT 1: Ejecución paso a paso**
 - Se activa cuando la bandera de traza (**TF**) vale **1** y la CPU ha ejecutado cualquier instrucción.
 - El DOS inicializa el vector de interrupción con una dirección que contiene la instrucción **IRET**.
 - Los programas de depuración **DEBUG**, **SYMDEB**, **TD**, ...) cambian el vector a una rutina de servicio que permite la ejecución paso a paso de los programas.

(4)

4.1. Interrupciones BIOS (III)

Asociadas a la CPU

- **INT 2: No Enmascarable**
 - Se activa con flanco ascendente en el pin NMI de la CPU. El pin está conectado al detector de paridad de la RAM.
 - Imprime en la consola "***Parity Check 1***" y detiene la CPU.
- **INT 3: Punto de ruptura (*breakpoint*)**
 - Se activa cuando que se ejecuta una instrucción con el código CCh.
 - Se usa en programas de depuración: permite la ejecución de un programa hasta que se encuentra esa instrucción.
 - El DOS inicializa el vector de interrupción con una dirección que contiene la instrucción **IRET**.

(4)

4.1. Interrupciones BIOS (IV)

Asociadas a la CPU

- **INT 4: Desbordamiento (*overflow*)**
 - Se activa mediante la instrucción **INTO**.
 - Genera una INT 4 sí bandera **O=1**.
 - El DOS inicializa el vector de interrupción con una dirección que contiene la instrucción **IRET**.
- **INT 5: Imprimir pantalla**
 - Esta interrupción imprime el texto que se está mostrando en pantalla.
 - Puede activarse pulsando la tecla **Impr-Pant**.
- **INT 6, INT 7 (No utilizadas)**

(4)

4.1. Interrupciones BIOS (V)

Asociadas al controlador de interrupciones

- Las interrupciones 8 a 15 (0Fh) están asociadas al controlador de interrupciones *hardware* (8259A) y se activan cada vez que se produce un flanco en sus entradas IRQ0 a IRQ7.
- INT 8: Temporizador
 - El temporizador del sistema (8253) activa esta interrupción 18.2 veces por segundo (cada 55 ms).
 - La rutina de servicio incrementa en uno el contador de 32 bits situado en las siguientes direcciones de la BIOS (y lo pone a cero cada 24 horas):
 - 0040h:006Ch (palabra baja)
 - 0040h:006Eh (palabra alta)
 - La rutina de servicio también activa una **INT 1Ch**.

(4)

4.1. Interrupciones BIOS (VI)

Asociadas al controlador de interrupciones

- INT 9: Teclado
 - Se activa cada vez que se pulsa o libera una tecla.
 - La rutina de servicio guarda el código de la tecla en el buffer de teclado.
- INT 0Ah (No utilizada)
- INT 0Bh: Puerto serie 1
- INT 0Ch: Puerto serie 2
- INT 0Dh: Disco duro (XT) o puerto paralelo 2 (AT)
- INT 0Eh: Disquete
- INT 0Fh: Puerto paralelo 1

(4)

4.1. Interrupciones BIOS (VII)

Servicios del BIOS

- **INT 10h: Entrada/Salida de vídeo**
 - Diversas funciones relacionadas con la salida de vídeo según el valor de **AH**.
- **INT 11h: Chequeo del equipo físico**
 - Retorna en **AX** una descripción del hardware instalado (bancos de memoria, número de puertos serie y paralelos, etc.).
- **INT 12h: Tamaño de memoria**
 - Retorna en **AX** el número de bloques de 1 KB de la memoria RAM instalada.
- **INT 13h: Acceso a disco**
 - Diversas funciones relacionadas con acceso a disquete o disco duro a nivel de sector o pista según valor de **AH**.

(4)

4.1. Interrupciones BIOS (VIII)

Servicios del BIOS

- INT 14h: Acceso a puerto serie RS-232
- INT 15h: Acceso a cassette
- INT 16h: Entrada/Salida de teclado
 - Diversas funciones relacionadas con el teclado según el valor de **AH**.
- INT 17h: Entrada/Salida de impresora
- INT 18h: Ejecución del BASIC
- INT 19h: Inicio del sistema
 - Lee el sector 1 de la pista 0 del disco de arranque y ejecuta el programa de arranque del DOS.
- INT 1Ah: Hora del día
 - Acceso al contador de 32 bits del temporizador (**INT 8**).

(4)

4.1. Interrupciones BIOS (IX)

Rutinas de usuario

- **INT 1Bh: Ruptura desde teclado**
 - La activa la rutina de servicio de la **INT 9** (teclado) cuando detecta **Ctrl-C** (Ctrl-Break).
 - El BIOS inicializa el vector de interrupción con una dirección que contiene la instrucción **IRET**.
 - El DOS cambia el vector de interrupción a la rutina de servicio de la **INT 23h** (rutina de servicio de Ctrl-Break).
- **INT 1Ch: Tic del temporizador**
 - La activa la rutina de servicio de la **INT 8** (timer).
 - El BIOS inicializa el vector de interrupción con una dirección que contiene la instrucción **IRET**.

(4)

4.1. Interrupciones BIOS (X)

Punteros a tablas de datos

- Las interrupciones **1Dh** a **1Fh** y **41h** son en realidad direcciones de tablas de parámetros usadas por los servicios de vídeo y disco del BIOS.
- INT 1Dh: Parámetros de vídeo
- INT 1Eh: Parámetros de disquete
- INT 1Fh: Tabla de caracteres gráficos
- INT 41h: Parámetros de disco duro

(4)

4.2. Interrupciones DOS (I)

- **INT 20h: Finaliza programa**
 - Acaba ejecución de programa retornando al intérprete de comandos. Microsoft recomienda usar en su lugar **INT 21h** con **AH=4Ch** (finaliza programa, cerrando ficheros y liberando memoria).
- **INT 21h: *Dispatcher* del DOS**
 - Ejecuta los distintos servicios del DOS según **AH**.
- **INT 22h: Dirección de terminación**
 - Dirección de la rutina que se ejecuta cuando finaliza el programa. No debe llamarse directamente.
- **INT 23h: Rutina de servicio de CTRL-Break**
 - Llamada por DOS cuando detecta CTRL-C (CTRL-Break). No debe llamarse directamente.

(4)

4.2. Interrupciones DOS (II)

- **INT 24h: Manejador de errores críticos**
 - Invocada por DOS cuando se produce un error crítico en acceso a un dispositivo hardware (disco, impresora, ...)
- **INT 27h: Finaliza programa dejando residente**
 - Acaba ejecución de un programa .COM (*driver*) dejándolo residente en memoria.
 - Para dejar residente un programa .EXE su usa en su lugar **INT 21h** con **AH=31h**.

(4)

4.3. Ejecución de programas desde el DOS

- Los programas en código máquina están almacenados en ficheros ejecutables de disco.
- Cuando se ejecuta un programa, el intérprete de comandos carga el contenido de su fichero ejecutable en una zona libre que reserva en memoria RAM.
- Como parte de la carga se añade una zona de 256 bytes que contiene datos relacionados con el programa (Prefijo de Segmento de Programa, PSP)
- Los ficheros ejecutables pueden estar en formato `.EXE` o `.COM`, teniendo su ejecución un comportamiento ligeramente distinto.
- Cuando acaba un programa, se devuelve el control al intérprete de comandos del DOS. La memoria que ocupaba se libera salvo que se deje residente.

(4)

4.4. PSP (Prefijo de Segmento de Programa) (I)

- 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 0 y 1 (2 bytes)
 - Instrucción **INT 20h**.
 - Permite acabar el programa saltando al offset 0 (no recomendado).
 - Offsets 0Ah a 0Dh (4 bytes)
 - Vector original de la rutina de servicio de la **INT 22h** (dirección de terminación de programa)
 - Cuando acaba el programa se copia a la tabla de vectores de interrupción y se salta a esa dirección.

(4)

4.4. PSP (Prefijo de Segmento de Programa) (II)

- Offsets 0Eh a 11h (4 bytes)
 - Vector original de la rutina de servicio de la **INT 23h** (Ctrl-Break)
 - El programa puede cambiar la rutina de esa interrupción para capturar Ctrl-C/Ctrl-Break.
 - Cuando acaba el programa se repone la rutina original copiando su dirección desde este campo a la tabla de vectores de interrupción.
- Offsets 12h a 15h (4 bytes)
 - Vector original de la rutina de servicio de la **INT 24h** (Manejador de errores críticos)
 - El programa puede cambiar la rutina de esa interrupción para capturar errores críticos.
 - Cuando acaba el programa se repone la rutina original copiando su dirección desde este campo a la tabla de vectores de interrupción.

(4)

4.4. PSP (Prefijo de Segmento de Programa) (III)

- Offsets 2Ch y 2Dh (2 bytes)
 - Número de segmento físico que contiene una copia 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.

(4)

4.4. PSP (Prefijo de Segmento de Programa) (IV)

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:

(4)

4.4. PSP (Prefijo de Segmento de Programa) (V)

Ejemplo

Dirección del manejador de error crítico: 103Dh:0956

Dirección del manejador de Ctrl-Break: 103Dh:0A2Bh

```
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 00 00
193F:0040 06 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00
193F:0050 CD 21 CB 00 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 05 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
```

Dirección de la rutina de final de programa: 103Dh:098E

Número de caracteres de los parámetros de entrada (12 bytes)

/D C:\DISCO ↵

(4)

4.4. PSP (Prefijo de Segmento de Programa) (VI)

Ejemplo

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 00 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 20 20 20
193F:0060 20 20 20 20 20 20 20 - 00 00 00 00 03 20 20 20
193F:0070 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
193F:00B0 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
193F:00D0 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
193F:00F0 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
```

(4)

4.5. Tipos de programas: EXE, COM y residentes (I)

- Tres tipos de ficheros ejecutables en DOS:
- **.BAT**
 - Son secuencias de 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).

(4)

4.5. Tipos de programas: EXE, COM y residentes (II)

- **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** indica si es correcta la unidad de disco (C, D, ...) del primer fichero (**AL= 0** es correcta).
 - **AH** indica si es correcta la unidad de disco (C, D, ...) del segundo fichero (**AH= 0** es correcta).
 - 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.

(4)

4.5. Tipos de programas: EXE, COM y residentes (III)

- **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** indica si es correcta la unidad de disco (C, D, ...) del primer fichero (**AL= 0** es correcta).
 - **AH** indica si es correcta la unidad de disco (C, D, ...) del segundo fichero (**AH= 0** es correcta).
 - 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.

(4)

4.5. Tipos de programas: EXE, COM y residentes (IV)

- 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 que se deja residente.
 - Ejemplo de instalación de una rutina de servicio de la interrupción 40h:

(4)

4.5. Tipos de programas: EXE, COM y residentes (V)

codigo **SEGMENT**

ASSUME cs : codigo

ORG 256

inicio: **jmp** instalador

; Variables globales

tabla **DB** "abcdef "

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

(4)

4.5. Tipos de programas: EXE, COM y residentes (VI)

- **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).
 - Ejemplo de desinstalación de rutina de servicio de la interrupción 40h:

(4)

4.5. Tipos de programas: EXE, COM y residentes (VII)

```
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
```