

INT 21H

La mayoría de servicios ó funciones del sistema operativo MS-DOS se obtienen a través de la interrupción software 21H. Es por esto que se le denomina DOS-API: DOS-APPLICATION-PROGRAM-INTERFACE La INT 21H está compuesta por un grupo de funciones. Cuando se accede a la INT 21H, hay que indicar el número de función que queremos ejecutar. La llamada a la INT 21H se realizará como sigue:

- Introducimos en (AH) el número de función a la que deseamos acceder.
- En caso de que deseemos acceder a una sub-función dentro de una función, debemos indicarlo introduciendo en (AL) el número de esa sub-función.
- Llamar a la INT 21H.

INT 21H Función 01H**Entrada de Carácter con Eco (ó salida)**

LLAMADA:

AH = 01H

DEVUELVE:

AL = Código ASCII del Carácter leído y Echo a pantalla
(si AH =08H) es servicio sin Echo)

EFECTO: Se lee un carácter del dispositivo de entrada estándar, y se envía al dispositivo estándar de salida. Si al llamar a la función no había ningún carácter disponible, se esperará a que lo haya. Normalmente el dispositivo estándar de entrada es el teclado, y el dispositivo estándar de salida es la pantalla.

INT 21H Función 02H**Salida de Carácter**

LLAMADA:

AH = 02H

DL = Código ASCII a enviar al dispositivo de salida.

DEVUELVE:

NADA.

EFECTO: Se envía el carácter depositado en el registro DL al dispositivo estándar de salida. La salida puede ser redireccionada hacia un fichero, impresora, etc

INT 21H Función 05H**Envío de Carácter a la Impresora**

LLAMADA:

AH = 05H

DL = Código ASCII a enviar al dispositivo de salida.

DEVUELVE: NADA

EFECTO: Se envía el carácter depositado en DL al puerto paralelo. Si no se ha redireccionado la salida, el dispositivo por defecto en el puerto paralelo de salida (LPT1 ó PRN) es la impresora.

INT 21H Función 06H**Envío de Carácter a la pantalla como símbolo**

LLAMADA:

AH = 06H

DL = símbolo entre "cuotation marks" 'A' ó "#".

DEVUELVE: NADA

INT 21H Función 09H**Visualización de una cadena de caracteres**

LLAMADA:

AH = 09H

DS:DX = Segmento: Desplazamiento de la cadena a visualizar (in DX - offset address of string in data segment)

DS debe apuntar al segmento donde se encuentra la cadena.

DX debe contener el desplazamiento de la cadena dentro de ese segmento.

Ejemplo: Definition of the message in Data segment

MSG DB "This is a message\$" or "This...", "\$"

Program in Code segment

Mov AH, 09H

Mov DX, Offset MSG

Int 21 H

DEVUELVE: NADA.

EFECTO: Se envía una cadena de caracteres al dispositivo estándar de salida. Si no se ha redireccionado la salida, la cadena se enviará a la pantalla.

INT 21H Función 0AH**Leer cadena de símbolos desde teclado y su código se almacena al buffer**

LLAMADA:

AH = 0AH

DX =dirección de buffer en segmento de datos

DEVUELVE:

En dirección DS:DX la cadena de ASCII códigos de símbolos entrantes

Ejemplo. Define Buffer address in data segment

BUF DB 6,0,0,0,0,0 (6-simbols, 5 places for input simbols by zero, and last zero is for RETURN). More than 6 - beep and ignore

In code segment the program

Mov AH; 0Ah

Mov DX, Offset BUF

INT 21H ; input 5 symbols to buffer with "return" code and output all of them to screen

INT 21H Función 0BH**Comprobación del estado de la entrada**

LLAMADA:

AH = 0BH

DEVUELVE:

AL = 00H si no hay carácter disponible.

AL = FFH si hay algún carácter disponible.

EFECTO: Se comprueba si hay algún carácter procedente del dispositivo estándar de entrada. La entrada puede ser redireccionada. Si no hay tal redirección, se comprueba el buffer de teclado

INT 21H Función 0EH**Selección de disco**

LLAMADA:

AH = 0EH

DL = Código de la unidad (0=A , 1=B , etc...)

DEVUELVE: AL = Número de unidades lógicas del sistema

EFECTO: Selecciona una unidad de disco para que se convierta en la unidad por defecto o actual. Ofrece además información acerca del número total de unidades lógicas del sistema. Una unidad física como un disco duro puede estar particionada en varias unidades lógicas designadas por C , D, E ,etc

INT 21H Función 19H

Obtener disco actual

LLAMADA:

AH = 19H

DEVUELVE:

AL = Código de la unidad actual (0=A , 1=B , etc...)

EFECTO: Devuelve el código de la unidad de disco activa o por defecto.

INT 21H Función 39H

Crear directorio

LLAMADA:

AH = 39H

DS:DX = Segmento: Desplazamiento de una cadena ASCII con el nombre del directorio.

DEVUELVE: Si se ejecutó correctamente: Flag de acarreo (Cf) = 0

Si NO se ejecutó correctamente: Flag de acarreo (Cf) = 1

AX = Código de error.

EFECTO: Se crea un nuevo directorio usando la unidad de disco y la vía de acceso especificada en la cadena ASCII.

INT 21H Función 3AH

Borrar directorio

LLAMADA:

AH = 3AH

DS:DX = Segmento: Desplazamiento de una cadena ASCII con el nombre del directorio a borrar.

DEVUELVE:

Si se ejecutó correctamente: Flag de acarreo (Cf) = 0

Si NO se ejecutó correctamente: Flag de acarreo (Cf) = 1

AX = Código de error.

EFECTO: Se elimina el directorio indicado de la unidad de disco y la vía de acceso especificada en la cadena ASCII.

INT 21H Función 3BH

Establecer directorio actual

LLAMADA:

AH = 3BH

DS:DX = Segmento: Desplazamiento de una cadena ASCII con el nombre del directorio.

DEVUELVE:

Si se ejecutó correctamente: Flag de acarreo (Cf) = 0
 Si NO se ejecutó correctamente: Flag de acarreo (Cf) = 1
 AX = Código de error.

EFECTO: Establece como directorio actual el indicado mediante la cadena ASCII.

INT 21H Función 47H **Obtener directorio actual**

LLAMADA:

AH = 47H

DL = Código de la unidad (0 = unidad por defecto, actual; 1 = A ; 2 = B ; etc...)

DS:SI = Segmento: Desplazamiento de un buffer de 64 bytes. Este buffer contendrá el nombre del directorio, con toda la vía de acceso al mismo, en forma de cadena ASCII.

DEVUELVE:

Si se ejecutó correctamente: Flag de acarreo (Cf) = 0

Si NO se ejecutó correctamente: Flag de acarreo (Cf) = 1

AX = Código de error.

EFECTO: Establece como directorio actual el indicado mediante la cadena ASCII. Esta función se abortará (indicándolo mediante Cf y AX) si el código de unidad no es válido.

Funciones de manejo de Ficheros

INT 21H Función 3CH

Crear Fichero

LLAMADA:

AH = 3CH

CX = Atributos del fichero: 00H Fichero Normal. 01H Fichero de Sólo Lectura. 02H Fichero Oculto. 03H Fichero de Sistema.

DS:DX = Segmento: Desplazamiento de una cadena ASCII con el nombre de fichero.

DEVUELVE:

Si se ejecutó correctamente: Flag de acarreo (Cf) = 0

AX = Handle o manejador de fichero.

Si NO se ejecutó correctamente: Flag de acarreo (Cf) = 1

AX = Código de error.

EFECTO: Si el fichero indicado mediante la cadena ASCII ya existía, entonces se vacía su contenido, quedando con longitud 0. Si el fichero no existía, entonces se crea. En ambos casos, el fichero se abre, y se devuelve un Handle ó manejador de fichero para los restantes accesos a dicho fichero. Esta función se abortará (indicándolo mediante Cf y AX) si:

-La vía de acceso indicada en la cadena ASCII es errónea.

-Se va a crear el fichero en el directorio raíz y éste ya está lleno

El Handle es un número mediante el cual se referencia a un fichero dado. De esta forma es más cómodo trabajar con ficheros. Para cada uno de los ficheros con que trabajemos, tendremos un número (Handle) que lo representará.

INT 21H Función 3DH

Abrir Fichero

LLAMADA:

AH = 3DH

AL = Modo de acceso. Bits 0-2: Permiso de lectura/escritura.

000b Acceso de sólo lectura.

001b Acceso de sólo escritura.

010b Acceso de lectura/escritura.

Bit 3: 0b (Reservado).

Bits 4-6: Modo de compartición de Fichero

000b Sólo el programa actual puede acceder al archivo

001b Sólo el programa actual puede acceder al archivo.

010b Otro programa puede leer el archivo, pero no escribir en él.

011b Otro programa puede escribir en el archivo, pero no leerlo.

100b Otro programa puede leer y escribir en el archivo.

Bit 7: Bit de 'herencia' ó Handle-flag.

0b Si el handle se hereda por un programa hijo.

1b Si el handle no se hereda.

DS:DX = Segmento: Desplazamiento de una cadena ASCII con el nombre de fichero.

DEVUELVE:

Si se ejecutó correctamente Flag de acarreo (Cf) = 0

AX = Handle o manejador de fichero

Si NO se ejecutó correctamente: Flag de acarreo (Cf) = 1

AX = Código de error.

EFECTO: Mediante esta función se abre un fichero ya existente, y se devuelve un Handle para acceder al fichero en lo sucesivo.

INT 21H Función 3EH

Cerrar Fichero

LLAMADA:

AH = 3EH

BX = Handle.

DEVUELVE:

Si se ejecutó correctamente:

Flag de acarreo (Cf) = 0

Si NO se ejecutó correctamente: Flag de acarreo (Cf) = 1

AX = Código de error.

EFECTO: Mediante esta función se cierra un fichero que estuviera abierto. Se utiliza el Handle para indicar el fichero a cerrar. Tras cerrar el fichero, dicho Handle se libera para nuevos ficheros. Se vuelvan al disco todos los buffers internos asociados al fichero.

INT 21H Función 3FH

Lectura de Fichero o dispositivo

LLAMADA:

AH = 3FH

BX = Handle.

CX = Número de bytes a leer.

DS:DX = Segmento: Desplazamiento del buffer donde se depositarán los caracteres leídos

DEVUELVE:

Si se ejecutó correctamente: Flag de acarreo (Cf) = 0

AX = Bytes transferidos

Si NO se ejecutó correctamente: Flag de acarreo (Cf) = 1

AX = Código de error.

EFECTO: Dado un handle válido, se realiza una transferencia desde el fichero referenciado por ese handle hacia el buffer de memoria especificado mediante DS:DX. Se transferirán tantos caracteres como se especifique en CX. Acto seguido, se actualiza el puntero de fichero hasta el carácter que sigue al bloque leído. Mediante esta función es posible leer caracteres del teclado, usando el handle 0.

INT 21H Función 40H

Escritura en Fichero o dispositivo.

LLAMADA:

AH = 40H

BX = Handle.

CX = Número de bytes a escribir.

DS:DX = Segmento: Desplazamiento del buffer desde donde se van a tomar los caracteres a escribir.

DEVUELVE:

Si se ejecutó correctamente: Flag de acarreo (Cf) = 0

AX = Bytes transferidos.

Si NO se ejecutó correctamente: Flag de acarreo (Cf) = 1

AX = Código de error

EFECTO Dado un handle válido, se realiza una transferencia desde el buffer de memoria indicado mediante DS:DX hacia el fichero o dispositivo referenciado por el Handle. Se transferirán tantos caracteres como se especifique en CX. Acto seguido, se actualiza el puntero de fichero una posición por delante del bloque escrito, para que futuras escrituras no 'machaquen' los datos que ya hubiera. Mediante esta función es posible escribir caracteres en la pantalla, usando el handle 1.

INT 21H Función 41H

Borrar Fichero.

LLAMADA:

AH = 41H

DS:DX = Segmento: Desplazamiento de la cadena ASCII con el nombre del fichero a borrar.

DEVUELVE:

Si se ejecutó correctamente: Flag de acarreo (Cf) = 0

Si NO se ejecutó correctamente: Flag de acarreo (Cf) = 1

AX = Código de error.

EFECTO: Se borra el fichero indicado mediante la cadena ASCII

INT 21H Función 42H

Establecer puntero de fichero.

LLAMADA:

AH = 42H

AL = Código de desplazamiento:

00h Desplazamiento desde el inicio del fichero.

01h Desplazamiento desde la posición actual del puntero.

02h Desplazamiento desde el final del fichero.

BX = Handle del fichero.

CX = Mitad más significativa del desplazamiento.

DX = Mitad menos significativa del desplazamiento.

DEVUELVE:

Si se ejecutó correctamente
Flag de acarreo (Cf) = 0
DX = Mitad más significativa del puntero actualizado.
AX = Mitad menos significativa del puntero actualizado.
Si NO se ejecutó correctamente
Flag de acarreo (Cf) = 1
AX = Código de error.

EFECTO:

Cambia el valor del puntero de fichero, permitiendo así un acceso aleatorio al fichero. Podremos, mediante esta función, escribir y leer caracteres en cualquier posición del fichero, sin pasar por las anteriores.

**INT 21H Función 43H Subfunción 00h
Obtener atributos de fichero****LLAMADA:**

AH = 43H

AL = 00h

DS:DX = Segmento: Desplazamiento de la cadena ASCIIZ con el nombre del fichero.

DEVUELVE

Si se ejecutó correctamente:

Flag de acarreo (Cf) = 0

CX = Atributos del fichero

Si Bit 0 = 1, Fichero de sólo lectura.

Si Bit 1 = 1, Fichero oculto.

Si Bit 2 = 1, Fichero de sistema.

Si Bit 5 = 1, El archivo ha sido modificado desde el último backup.

Si NO se ejecutó correctamente:

Flag de acarreo (Cf) = 1

AX = Código de error.

EFECTO: Mediante esta función se obtiene información acerca de los atributos de un determinado fichero, indicado mediante la cadena ASCII.

**INT 21H Función 43H Subfunción 01h á
Establecer atributos de fichero****LLAMADA:**

AH = 43H

AL = 01h

CX = Nuevos atributos para fichero.

Si ponemos Bit 0 = 1, atributo de sólo lectura.

Si ponemos Bit 1 = 1, atributo de oculto.

Si ponemos Bit 2 = 1, atributo de sistema.

Debemos establecer Bit 3 = 0

Debemos establecer Bit 4 = 0.

Si ponemos Bit 5 = 1, indicará que el fichero ha sido modificado desde el último backup.

DS:DX = Segmento: Desplazamiento de la cadena ASCIIZ con el nombre del fichero.

DEVUELVE:

Si se ejecutó correctamente

Flag de acarreo (Cf) = 0

Si NO se ejecutó correctamente:

Flag de acarreo (Cf) = 1
 AX = Código de error.

EFECTO:

Mediante esta función se establecen nuevos atributos a un fichero dado. Este fichero se indica mediante la cadena ASCII.

INT 21H Función 56H**Renombrar Fichero ó Mover Fichero**

LLAMADA:

AH = 56H

DS:DX = Segmento: Desplazamiento de la cadena ASCII con el nombre actual del fichero.

ES:DI = Segmento:Desplazamiento de la cadena ASCII con el nuevo nombre para el fichero.

DEVUELVE: Si se ejecutó correctamente:

Flag de acarreo (Cf) = 0

Si NO se ejecutó correctamente:

Flag de acarreo (Cf) = 1

AX = Código de error.

EFECTO:

Esta función permite cambiar el nombre a un fichero dado. También permite mover el fichero a otro directorio dentro del mismo dispositivo. En el segundo caso, lo que debemos hacer es cambiar el camino que nos lleva al fichero, en vez del nombre de fichero en sí.

VERSION DE MS-DOS: 2.0 ó superior.

NOTAS: Se producirá un error si:

- la cadena ASCII indicada es errónea.
- ya existe un fichero con el mismo nombre que el indicado.
- Se intenta mover el fichero a otro dispositivo o unidad.
- El fichero se mueve al directorio raíz, y éste está lleno.

Funciones de terminación de procesos**INT 21H Función 00H****Terminar ejecución del Programa**

LLAMADA:

AH = 00H

CS = Dirección del segmento del PSP.

DEVUELVE:

NADA

EFECTO: Se finaliza la ejecución del programa en curso. Esto conlleva:

- Liberación de toda la memoria asignada al programa.
- Todos los buffers de fichero son vaciados.
- Se cierra cualquier fichero abierto por el programa.
- Se restauran los tres vectores de interrupción (INT 22H, INT 23H, INT 24H) cuyo contenido original fue almacenado en la pila. Esta función está pensada para programas COM, en los cuales, el PSP está en el mismo segmento que el código del programa. En los programas EXE esto no sucede así, estando el PSP en diferente segmento al código del programa, por tanto, no se debe llamar a esta función desde un programa .EXE. En ambos casos (COM y EXE), es preferible utilizar la función 4CH, ya que devuelve un código de retorno la programa padre.

INT 21H Función 4CH**Terminación de Programa con Código de Retorno**

LLAMADA:

AH = 4CH

AL = Código de retorno para el programa padre

DEVUELVE:

NADA.

EFECTO: Se finaliza la ejecución del programa en curso, y se devuelve un código de retorno al programa padre. Mediante este código de retorno, se puede ofrecer información al programa padre acerca de la ejecución del programa (si se ha producido error, etc...) La terminación del programa conlleva:

- Liberación de toda la memoria asignada al programa.
- Todos los buffers de fichero son vaciados.
- Se cierra cualquier fichero abierto por el programa.
- Se restauran los tres vectores de interrupción (INT 22H, INT 23H, INT 24H) cuyo contenido original fue almacenado en la pila.

Este es el método idóneo de terminación de programas, ya que no necesita que el registro CS tenga ningún contenido especial. Y aparte, devuelve información al programa padre.

INT 21H Función 31H**Finalizar el Programa y Dejar Residente**

LLAMADA:

AH = 31H

AL = Código de retorno para el programa padre.

DX = Cantidad de memoria a dejar residente (en párrafos).

DEVUELVE:

NADA

EFECTO: Se finaliza la ejecución del programa en curso, y se devuelve un código de retorno al programa padre. Mediante este código de retorno, se puede ofrecer información al programa padre acerca de la ejecución del programa (si se ha producido error, etc...) Además de esto, y lo más importante: Se deja residente el programa o parte del mismo, de forma que las siguientes ejecuciones de programas no lo 'machaquen'. La terminación del programa conlleva:

- Liberación de toda la memoria asignada al programa.
- Todos los buffers de fichero son vaciados.
- Se cierra cualquier fichero abierto por el programa.
- Se restauran los tres vectores de interrupción (INT 22H, INT 23H, INT 24H) cuyo contenido original fue almacenado en la pila.

Un párrafo equivale a 16 bytes. Por tanto, en DX habrá que indicar el número total de bytes a dejar residentes dividido por 16. Esta función se utiliza en programación de utilidades residentes (como SideKick o SanBit). Una vez que el programa queda residente en memoria, la activación del mismo se realiza de acuerdo a algún criterio del propio programa (como la pulsación de una combinación de teclas). En este caso, el programa residente toma el control y al acabar su tarea le devuelve el control al programa que se estuviera ejecutando antes de la activación del residente.