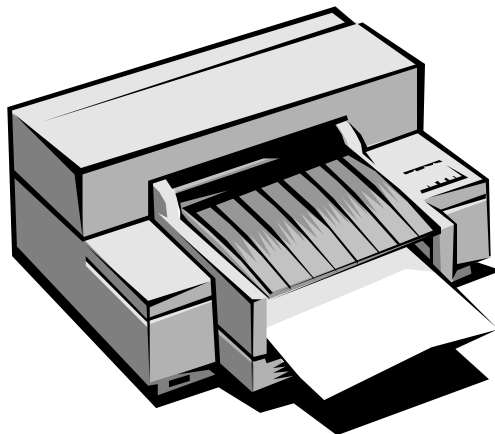


# AMPLIACIÓN DE SISTEMAS OPERATIVOS

*Controlador de la impresora del Minix 2.0*



printer.c

# EL MANEJADOR DE IMPRESORA. PRINTER.C

## 1. INTRODUCCIÓN.

Entre las diferentes interrupciones de las que dispone la BIOS para la llamada de sus servicios, la interrupción 17h está exclusivamente reservada para la comunicación con el puerto paralelo. A pesar de que a este puerto se pueden conectar teóricamente otros periféricos, no sólo la impresora, se habla frecuentemente de la interrupción BIOS de la impresora, ya que en la práctica no se pueden encontrar otros periféricos en este puerto. Los conceptos “impresora” y “puerto paralelo” en este sentido son intercambiables.

En el PC se pueden conectar un máximo de tres puertos paralelos diferentes. Se pueden utilizar mediante tres funciones distintas, que la BIOS pone a nuestra disposición a través de la interrupción 17h. Las tres funciones se ocupan de tareas muy distintas y variadas en el marco de las comunidades con un puerto paralelo, y pueden utilizarse, según se desee, en cada uno de los tres puertos paralelos.

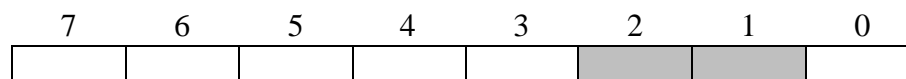
Tareas de las tres funciones de la interrupción de la impresora de la BIOS

Función	Tarea
00h	Enviar carácter
01h	Inicializar impresora
02h	Preguntar estado impresora

### *El estado de la impresora*

Aparte de la transferencia del número de puerto de comunicaciones existe una cosa en común en las tres funciones, ya que cada una de ellas devuelve el estado actual de la impresora en el registro AH, después de su llamada. Los diferentes bits de este byte de estado nos informan sobre si la impresora está ocupada en este momento; si aún tiene papel, o si ha ocurrido un error durante la transmisión de un carácter. Para la comunicación con el puerto paralelo, este estado tiene una gran importancia.

Estructura del bit de estado de la impresora



Significado de los bits:

Bit 0:Error de time-out, lógica positiva

Bit 1:No usado

Bit 2:No usado

Bit 3:Error en la transmisión, lógica negativa

Bit 4:Impresora ON-LINE, lógica positiva.

Bit 5:Impresora sin papel, lógica positiva

Bit 6:Confirmación de recepción, lógica positiva

Bit 7:Impresora ocupada, lógica negativa

**Los Ports I/O del puerto**

Se pueden conectar hasta tres puertos paralelos al PC sin ningún problema, ya que hay reservadas tres zonas en el espacio de direcciones de I/O del PC para puertos paralelos:

Port I/O de los puertos paralelos

Port	Puerto
3BCh-3BFh	Puerto paralelo en la tarjeta MDA
378h-37Fh	Primer puerto paralelo
278h-27Fh	Segundo puerto paralelo

La BIOS analiza durante su inicialización los Ports enumerados, para descubrir todos los puertos paralelos instalados.

**Selección de LPT1 hasta LPT3**

La selección de los diferentes puertos paralelos como LPT1, LPT2 y LPT3 se realiza colocando sus direcciones de base en las variables del segmento de variables del BIOS. Allí se encuentra, a partir de la dirección de Offset 0008h un array de cuatro palabras, en las que se guardan las direcciones de Port de los puertos paralelos.

Variables BIOS con las direcciones de Port de los puertos paralelos

Puerto	Significado
0040:0008h	Dirección base LPT1
0040:000Ah	Dirección base LPT2
0040:000Ch	Dirección base LPT3
0040:000Eh	Dirección base LPT4

A pesar de que el BIOS al arrancar sólo busca tres puertos paralelos, en la zona de variables aparentemente queda espacio para cuatro puertos. Y realmente, las funciones de la BIOS también se pueden utilizar en relación a un cuarto puerto paralelo, poniendo simplemente su dirección de base(a mano) en el segmento de variables de la BIOS, y al llamar la función de la BIOS se especifica como número de puerto un 3.

**Los registros del puerto**

Independientemente de su posición en la zona de direcciones, todos los puertos paralelos muestran un Register-Interface unificado, que se compone de tres Ports. Ocupan las tres primeras direcciones Port de la tarjeta.

Los bits de los registros de puerto están en conexión directa con las líneas de conexión de un cable Centronics. Si se escribe el valor 1 en uno de estos Bits de registro, inmediatamente se pone bajo tensión la línea correspondiente. Y al revés, la línea pasa a Low, cuando el bit que la controla se pone a cero. La norma básica es que la línea mantiene su estado hasta que el bit correspondiente se modifica por software.

**Líneas de datos**

Los ocho bits del primer registro de un puerto paralelo acoge ocho bits de datos, que se corresponden a las líneas de datos D0 a D7 y que por ello han de ser transferidos al contrario. Este registro del puerto Centronics está concebido como registro puro de salidas, ya que no se planeó ninguna recepción de datos.

**Estado de impresora**

El estado actual de la impresora se obtiene del segundo registro, que sólo puede ser leído, ya que en él se reflejan las diferentes líneas de estado de la impresora.

**Control de la impresora**

El tercer registro sirve de control de la impresora y el hardware. Además juega un papel importante durante la transmisión de caracteres. Excepto el bit 4, todos los demás bits están unidos a las diferentes líneas del puerto Centronics.

Además este registro contiene un bit, con cuya ayuda se puede provocar una interrupción de hardware, en cuanto la señal ACK pasa a estado Low y con ello la impresora confirma la recepción del último carácter. De qué interrupción se trata se puede determinar habitualmente mediante los interruptores DIP que hay en el puerto de comunicaciones. Se pueden elegir entre IRQ7 y IRQ5, que están unidas con las interrupciones 0Fh y 0Dh, sin embargo, y contrariamente a los puertos serie, esta posibilidad se utiliza muy raramente, ya que los puertos paralelos se hacen funcionar con el método Polling y no el de interrupciones. Esto también es válido para la BIOS, que deja en paz estos vectores de interrupción.

**Los cables**

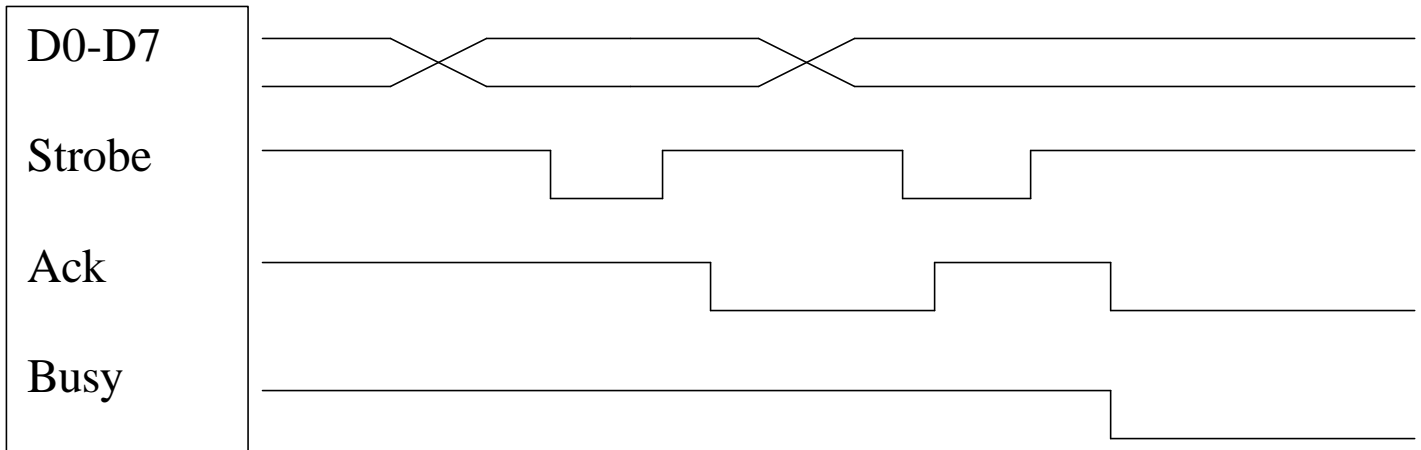
Todo el ir y venir entre ordenador e impresora naturalmente sólo funciona cuando los pins correctos de los dos puertos están conectados entre sí de forma adecuada.

Conexión entre un puerto paralelo y una impresora mediante un cable Centronics.

Pin del ordenador	Pin de la Impresora	Nombre de la línea	Significado
1	1	-STROBE	Indica transmisión
2	2	DO	Línea de datos Bit 0
3	2	D1	Línea de datos Bit 1
4	2	D2	Línea de datos Bit 2
5	2	D3	Línea de datos Bit 3
6	2	D4	Línea de datos Bit 4
7	2	D5	Línea de datos Bit 5
8	2	D6	Línea de datos Bit 6
9	2	D7	Línea de datos Bit 7
10	10	-ACK	Ultimo carácter aceptado
11	11	-BUSY	Impresora ocupada
12	12	PE	Impresora sin papel
13	13	SLCT	Impresora en ON LINE
14	14	-AUTO FEED	CR Automático después de LF
15	32	-ERROR	Error en la transmisión de datos
16	31	-INCT	Realizar reset de la impresora
17	36	SLCT IN	Activar ON LINE en la impresora
18-25	19-30	GND	Masa

### El protocolo de comunicación

Este protocolo es bastante sencillo como se podrá observar en la figura y su funcionamiento se detalla a continuación, teniendo en cuenta que las líneas BUSY, STROBE y ACK, son activas mediante lógica negativa.



Tenemos que inicialmente ponemos los datos en la línea de datos, luego tenemos que se activa la señal de STROBE que indica que se encuentra disponible el dato. A continuación se reconoce el dato y se vuelve a enviar otro dato y se activa la señal de STROBE para indicar que este se encuentra disponible. Posteriormente se activa la señal BUSY debido a que el buffer está lleno.

## 2.- EL PRINTER.C

El *printer.c* contiene el manejador de la impresora y posee las siguientes características:

- Sólo permite el uso de una impresora.
- No efectúa ninguna operación sobre los caracteres que le llegan (es el propio comando **lpr** el que realiza la expansión de los tabuladores y la inserción de los retornos de carro).
- Sólo es capaz de tratar simultáneamente una tarea de impresión (análogamente a lo que ocurre en MSDOS).

El mensaje a imprimir (contenido en el buffer de usuario) se va fragmentando en unidades de 128 bytes que se almacenan en un buffer intermedio, para luego ser pasadas al puerto base de la impresora hasta completar el mensaje inicial.

El mecanismo de impresión es controlado mediante interrupciones, cada una de las cuales provoca la salida de tantos caracteres como la impresora sea capaz de admitir en su buffer (si existe) o de tantos como haya hasta el final del buffer intermedio.

### 3.- CARACTERÍSTICAS DEL MANEJADOR

#### a) Acceso al controlador.

El manejador de impresora tiene tres puntos de entrada declarados como PUBLIC, que son las rutinas *printer\_task*, *pr\_handler* y *pr\_restart*.

La rutina *printer\_task* es llamada desde el programa **main.c**, cuando se realiza la inicialización de procesos de MINIX. En este proceso se crea su entrada en la tabla de procesos y se deja la tarea lista para su ejecución. La primera vez que esta tarea consiga el acceso a la CPU inicializará la impresora y se quedará a la espera de un mensaje de tipo HARD\_INT que le pueda llegar desde el procedimiento **interrupt**, que reactivará la tarea. Este tipo de mensaje provoca la ejecución del procedimiento *do\_done* que en esta primera llamada no haría nada. Posteriores llamadas al mismo procedimiento debidas a recepciones de este tipo de mensajes, reanudarán el proceso de impresión en el caso de que éste no haya finalizado todavía (que quede algo en el buffer de usuario).

El FS también puede enviar otros dos tipos de mensajes a la rutina: DEV\_WRITE, CANCEL. El primero indica que se comience una tarea de impresión y el segundo indica que se cancele.

Existen dos últimos tipos de mensajes que puede enviar el FS: DEV\_OPEN y DEV\_CLOSE. Sirven para preguntar si la impresora está abierta o cerrada respectivamente.

El procedimiento *pr\_handler* es un manejador de interrupciones llamado desde el **mpx88.x**, cuando llega una interrupción desde la impresora al 8259.

El acceso a *pr\_restart* se hace desde el **clock.c**, cuando se considera que puede existir algún problema con la impresora, como por ejemplo que esté colgada. En este caso, esta rutina se encargaría de reinicializar la impresora.

### 4.- PROCEDIMIENTOS

<b>printer_task</b> ( )	Procedimiento principal del manejador.
<b>print_init</b> ( )	Inicializa la impresora.
<b>do_write</b> ( )	Comprueba las condiciones de impresión.
<b>do_done</b> ( )	Se ejecuta al finalizar la impresión del contenido del buffer intermedio o cuando se ha producido algún error durante la impresión.
<b>do_cancel</b> ( )	Cancela la tarea de impresión.
<b>reply</b> ( )	Retorna un mensaje al procedimiento que llamó al manejador de la impresora.
<b>pr_start</b> ( )	Transfiere los datos a imprimir del buffer de usuario al buffer intermedio.
<b>pr_handler</b> ( )	Manejador de interrupciones. Envía caracteres a la impresora.

**pr\_restart ( )**      Chequea si la impresora está colgada en cuyo caso la reanuda.

## 5. COMENTARIO DEL LISTADO DEL PROGRAMA

### a) *printer\_task*

Procedimiento principal del manejador. Forma parte de las tareas básicas del sistema que son inicializadas al arrancar MINIX. Es uno de los procesos cuyo estado se muestra al pulsar la tecla F1.

Lo primero que hace es llamar al *print\_init* y luego permanece en un bucle a la espera de la llegada de algún mensaje. Si el mensaje es de tipo DEV\_OPEN o DEV\_CLOSE responde con un mensaje de OK. Si el mensaje es de tipo DEV\_WRITE, llama al procedimiento *do\_write*; si es de tipo CANCEL llama a *do\_cancel* y si es de tipo HARD\_INT llama a *do\_done*. En cualquier otro caso, responde con un mensaje de error.

### b) *print\_init*

Se encarga de inicializar la impresora. Para ello en primer lugar obtiene la dirección del puerto base de la primera impresora del sistema, luego lanza por este puerto el bit de inicialización de impresora, espera un pequeño intervalo de tiempo (por exigencias de la interface Centronics), y siguiendo el mismo procedimiento antes descrito, selecciona la impresora y habilita las interrupciones de impresión.

### c) *do\_write*

Se encarga de comprobar si se verifican las condiciones necesarias para llevar a cabo la impresión. En primer lugar testea que haya finalizado la última escritura y que la cuenta de bytes a imprimir sea positiva.

Si no hay errores, se salva la información relativa al estado de la tarea, el número del proceso del usuario y el número del proceso llamador (FS). A continuación comprueba que el usuario no rebase los límites de su área de trabajo y, si no es así, llama al procedimiento *pr\_start* e indica que la función de escritura está activada.

Finalmente llama a *reply* para contestar al FS. Toma como parámetro de entrada la estructura mensaje.

### d) *do\_done*

Se accede a este procedimiento cuando se finaliza de imprimir el contenido del buffer intermedio o cuando se ha producido algún error durante esta impresión. Si se trata del primer caso y aún queda algo más por imprimir llama al procedimiento *pr\_start*. Si no es así, puede ser debido a alguna condición de error o porque se ha acabado de imprimir todo el buffer de usuario. Ambas situaciones provocan el envío de un mensaje al proceso llamador.

Otro posible caso que se comprueba es que la interrupción (HARD\_INT) se haya producido mientras se estaba en un proceso de cancelación de una tarea; si es así, simplemente se efectúa un retorno.

### **e) do\_cancel**

La razón habitual por la cual se accede a este procedimiento es que un proceso de usuario que estaba realizando una impresión haya sido matado por una señal, con lo que es necesario cancelar la tarea de impresión.

Sólo permite que una tarea la cancele el mismo proceso que la generó. Recibe como parámetro un puntero al mensaje recién llegado al manejador.

### **f) reply**

Este procedimiento se ocupa de retornar al proceso que llamó (normalmente el FS) al manejador de impresora un mensaje indicando que ha comenzado la impresión o bien que ésta ha finalizado.

Los parámetros de entrada son el tipo de mensaje a devolver, el destinatario (normalmente el FS), el identificador de proceso al cual pertenece el mensaje y el estado (conteniendo el número de caracteres impresos si no se ha producido error o el código de error en otro caso).

### **g) pr\_start**

Procedimiento encargado de copiar un bloque del buffer de usuario al buffer intermedio. Testea la longitud del buffer de usuario que resta por imprimir y en base a esto traslada un trozo del mismo (128 bytes) o bien el buffer completo. También se ocupa de poner los valores adecuados a los flags necesarios para indicar que hay algo listo para imprimir.

### **h) pr\_handler**

Este procedimiento es el manejador de interrupciones que se generan a través del 8259A. Es llamado por una de las rutinas del MPX88.x. Tal como se explica en el listado del programa, hay un posible error que se puede producir por la generación de interrupciones espúreas por parte del 8259A, ya que estas se generan justamente por la línea correspondiente a las interrupciones de impresora. Esta situación no causa ningún problema al manejador (la trata como un efecto lateral, y se considera que corresponde a un estado de busy en la impresora), pero si podría causárselos al proceso que debería recibir la interrupción y no la recibe.

El procedimiento *pr\_handler* es el que se encarga realmente de enviar caracteres a la impresora a través de su puerto base. En primer lugar comprueba que quede algo por imprimir en el buffer intermedio. Si no es así, deselecciona la impresora y retorna.

Si queda algo entra en un bucle en el cual intentará enviar a la impresora consecutivamente tantos caracteres como sea posible, en función de la rapidez de la impresora y de que disponga de buffer o no. Después de sacar cada carácter, lee la palabra de estado y en caso de que muestre un estado ocupado significa que la impresora ya no es capaz de recibir más datos.



El procedimiento coloca un carácter en el puerto, y antes de activar el STROBE para que la impresora lea aquel, deshabilita las interrupciones para asegurarse de que el tiempo que el STROBE permanece activo no es excesivo (podría ocurrir que la impresora leyera dos veces el mismo carácter); cuando ya ha negado el STROBE, habilita de nuevo las interrupciones.

Si durante la permanencia en el bucle de enviar caracteres finaliza el contenido del buffer intermedio, llama a interrupt con un estado OK.

### ***i) pr\_restart***

Este procedimiento constituye el tercer punto de acceso público al manejador. Si quedan bytes en el buffer de salida, llama a *pr\_handler* y habilita las interrupciones.

## **6.- CUESTIONES**

### • ***¿Cómo se realiza la comunicación con la impresora a través de su puerto base?***

La comunicación se realiza utilizando alguno de los tres registros según el propósito:

- Primer registro: Es el registro de datos.
- Segundo registro: Es el registro de estado de la impresora.
- Tercer registro: Es el registro de control de la impresora.

### • ***¿Quién puede llamar al printer.c?***

El printer.c puede ser llamado por:

- Main.c, cuando realizamos la inicialización de Minix.
- File System, cuando es llamado a petición de un usuario.
- Mpx386.c, cuando el puerto de impresora genera una interrupción.
- Clock.c, que la llama cada click de reloj.

### • ***Realizar un seguimiento de una tarea de impresión.***

1. El FS envía un mensaje DEV\_WRITE al proceso en ejecución *PRINTER\_TASK*.
2. *PRINTER\_TASK* atiende el mensaje y ejecuta el procedimiento *do\_write*, que inicializa las variables para indicar que hay una impresión en curso:

- |  |  |
|--|--|
| - user_left = COUNT.   | Número total de bytes que quedan por imprimir. |
| - Orig_count = COUNT.  | Número total de bytes.                         |
| - User_vir = ADDRESS.  | Dirección del buffer del usuario.              |
| - writing = TRUE.  | Hay un proceso de impresión en marcha.         |
| - Reply(..., SUSPEND).   | Suspende al proceso que realizó la petición.   |
| - Llamada a <i>pr_start</i> para inicializar el buffer intermedio (obuf). Copia 128 bytes (o menos si se van a imprimir menos de 128 bytes) desde el buffer del usuario, inicializa las variables que conciernen al buffer intermedio (optr, opending, oleft). |  |

En este punto se termina la ejecución de *do\_write*, y como resultado se tienen datos en el buffer intermedio y el proceso que hizo la petición está suspendido.

3. Se ejecuta *pr\_restart* llamado desde *clock.c* en cada tick de reloj. Este procedimiento hace una llamada a *pr\_handler* que se encarga de escribir los caracteres en el puerto de la impresora. Cuando la impresora termina de escribir los caracteres envía una interrupción, que es atendida por *pr\_handler*, por lo que así se realiza el bucle de impresión. Cuando *pr\_handler* ha vaciado

el contenido del buffer *obuf*, pone la variable *done\_status* a OK y envía un mensaje HARD\_INT a *PRINTER\_TASK*.

4. *PRINTER\_TASK* recibe el mensaje HARD\_INT y ejecuta *do\_done*, que comprueba el valor de *done\_status*. En caso de ser OK y quedar bytes por imprimir (*user\_left* <> 0), se llama a *pr\_start* para llenar de nuevo el buffer intermedio y continuar la impresión.
5. Cuando se ha impreso todo, *done\_status* = OK y *user\_left* = 0, por lo que *do\_done* enviará un mensaje REVIVE al FS, con el número de bytes que se han impreso, y se habrá completado la petición de impresión.