Tema 5. Gestión de entrada/salida

Competencias:

- ✓ Justificar los problemas que presenta la diversidad de dispositivos de E/S, así como conocer los objetivos que el software de E/S intenta conseguir para resolver dichos problemas.
- ✓ Comprender la relación y comunicación entre cada uno de los niveles que componen el software de E/S, así como las estructuras de datos mantenidas por cada uno de dichos niveles.
- ✓ Conseguir capacidad de abstracción.
- ✓ Ser capaz de enfrentarse a problemas nuevos recurriendo conscientemente a estrategias que han sido útiles en problemas resueltos anteriormente.

Tema 5. Gestión de entrada/salida

1. Principios de la gestión de E/S.
   1. Problemática de los dispositivos de E/S.
   2. Objetivos generales del software de E/S.
   3. Principios hardware de E/S.
      1. E/S controlada por programa.
      2. E/S controlada por interrupciones.
2. Estructura del software de E/S.
   1. Niveles del software de E/S.
   2. Ejemplo del funcionamiento de los distintos niveles.
   3. La técnica del “buffering”.
   4. La técnica del “spooling”.

1. Principios de la gestión de E/S (I)

1.1. Problemática de los dispositivos de E/S.

Aspectos en los que pueden diferir los dispositivos de E/S:

1. *Diferencia de velocidad:* de los periféricos frente a la C.P.U. y entre los propios dispositivos de E/S.
2. *Unidad de transferencia:* caracteres, palabras, bytes, bloques o registros.
3. *Representación de los datos:* usar distintos códigos para un mismo elemento de información en diferentes periféricos.
4. *Operaciones permitidas:* por ejemplo, existen periféricos que solo realizan entradas y otros que solo realizan salidas, etc.
5. *Condiciones de error:* las causas por las que no se puede completar con éxito una transferencia de datos depende del periférico que se utilice.

1. Principios de la gestión de E/S (II)

1.2. Objetivos generales del software de E/S. (I)

1. **Independencia del periférico.** Aspectos a considerar:
   a) Un programa debe de ser independiente del modelo del periférico de un tipo determinado de periférico que le sea asignado.
   b) Un programa sea independiente lo más posible, del tipo de periférico empleado.
2. **Eficiencia.**
3. **Tratamiento uniforme de los periféricos.** Consecuencias:
   a) *Independencia del código de los caracteres:* el sistema de E/S es el responsable de reconocer los distintos códigos que usan los periféricos y traducirlos a una representación interna uniforme, denominada *código interno de los caracteres,* para los programas. Esta conversión se realiza después de la entrada y antes de la salida.

1. Principios de la gestión de E/S (III)

1.2. Objetivos generales del software de E/S. (II)

b) *Los programas deberán trabajar sobre periféricos virtuales (streams o ficheros) y no sobre físicos:* el S. O. es el que asocia los streams con los periféricos reales, guardando en una lista de descriptores de streams dicha correspondencia.

1. Principios de la gestión de E/S (III)

1.2. Objetivos generales del software de E/S. (III)

c) *El sistema de E/S se debe construir de forma que las características de los periféricos estén ligadas a ellos, en vez de a las rutinas que los gestiona.* Esto se obtiene codificando dichas características en una *tabla de descriptores de periférico,* de forma que los programas de gestión de los periféricos obtienen de ellas la información necesaria. Entre otra:

- ✓ Identificación del periférico,
- ✓ Instrucciones con las que actúa,
- ✓ Punteros a las tablas para la traducción de los caracteres,
- ✓ Estado actual (ocupado, libre o estropeado),
- ✓ Proceso de usuario en curso.

1. Principios de la gestión de E/S (IV)

1.3. Principios hardware de E/S. (I)

A. Tipos de dispositivos de E/S.

1. *dispositivos de bloques:* almacenan la información en bloques de tamaño fijo.
2. *dispositivos de caracteres:* producen o aceptan flujos de caracteres sin ninguna estructura de bloques.

B. Controladores de dispositivos: parte que controla al dispositivo.

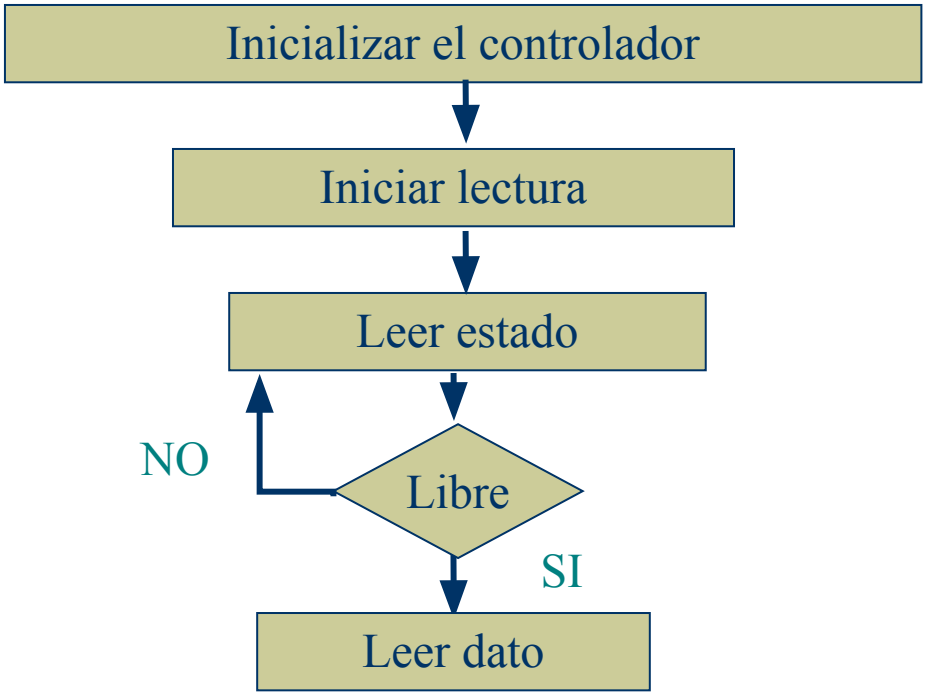
Los controladores usan unos cuantos registros denominados *puertos* para comunicarse con el procesador. A través de ellos se transfieren: *comandos* u *órdenes*, que hacen que el controlador inicie determinadas operaciones sobre el dispositivo; *parámetros* que indican cómo debe funcionar el dispositivo; y *datos*, ya sean de escritura o de lectura según el sentido de la operación.

1. Principios de la gestión de E/S (V)

1.3. Principios hardware de E/S. (II)

1.3.1. E/S controlada por programa.

La C. P.U. está ocupada mientras que se realiza la operación de E/S, pues debe de comprobar cuando termina dicha operación. Pasos:



1. Principios de la gestión de E/S (VI)

1.3. Principios hardware de E/S. (III)

1.3.2. E/S controlada por interrupciones.

Permite que la C. P.U. esté ocupada en alguna otra actividad mientras que se realiza la operación de E/S, pues se enterara de que dicha operación se ha completado cuando se produzca una interrupción.

Las interrupciones son un mecanismo que permite sincronizar la C.P.U. con los sucesos externos, y por lo tanto solapar una multitud de operaciones de E/S.

2. Estructura del software de E/S (I)

El software de E/S se organiza en niveles de forma que los niveles inferiores se encargan de ocultar las características del hardware a los niveles superiores, que, a su vez, se ocupan de presentar una interfaz simple y uniforme a los usuarios.

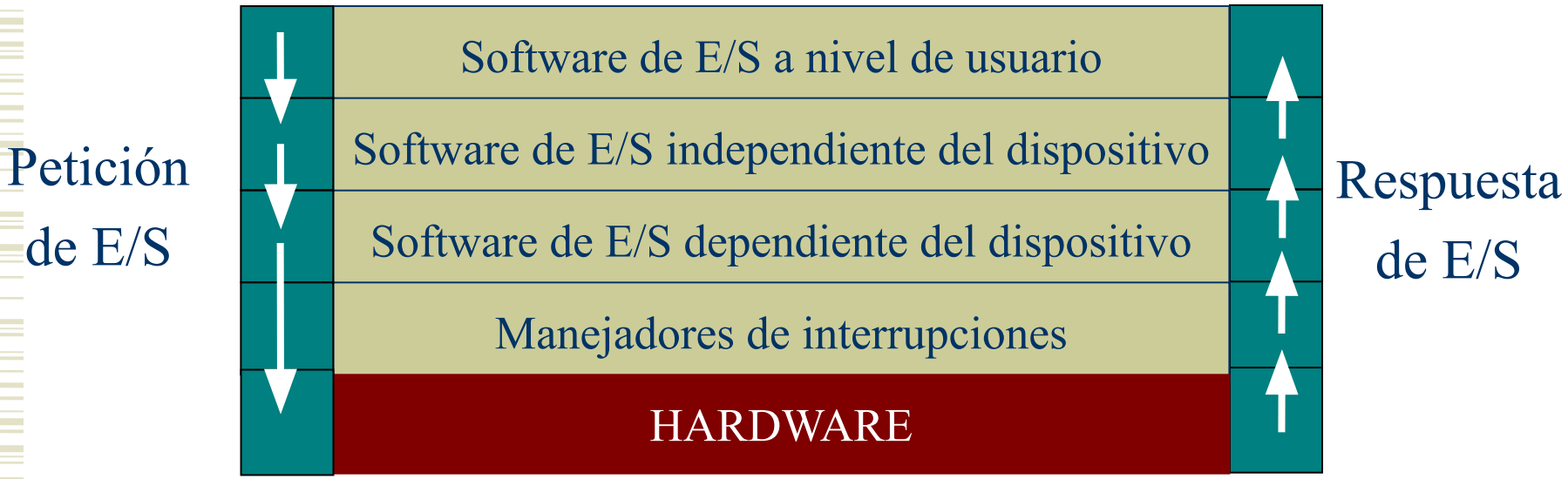
Objetivos:

1. *Independencia del dispositivo*: posibilidad de no tener que modificar los programas para cada tipo de dispositivo.
2. *Manejo de errores*: es mejor tratar los errores cuanto más cerca del hardware.
3. *Tipo de transferencias*: hacer que las operaciones que son controladas por interrupción parezcan al programa controladas por él.
4. *Tipos de dispositivos*: gestionar los dispositivos no compartibles.

2. Estructura del software de E/S (II)

2.1. Niveles del software de E/S. (I)

Niveles (Tanenbaum)



2. Estructura del software de E/S (III)

2.1. Niveles del software de E/S. (II)

Software de E/S a nivel de usuario.

No forma parte del S.O.. Consiste en bibliotecas que se enlazan con los programas de usuario y a través de las cuales se realizan las llamadas al S.O..

Funciones:

1. colocar los parámetros en el lugar adecuado para realizar la petición a niveles inferiores.
2. informar del error en caso de que se haya producido al realizar la operación de E/S.
3. opcionalmente, interpretar la información que se recibe o se manda (*formatear*).

2. Estructura del software de E/S (IV)

2.1. Niveles del software de E/S. (III)

Software de E/S independiente del dispositivo.

Pertenece al S.O.. Cuando el nivel anterior realiza una petición a este se origina la ejecución de una rutina de este software de E/S.

Funciones:

1. implementar las operaciones de E/S que son comunes a todos los dispositivos y presentar una interfaz uniforme a los programas de usuario;
2. asignar nombres simbólicos a los dispositivos de E/S y establecer la correspondencia entre el nombre simbólico del dispositivo y el manejador correspondiente;
3. proteger los dispositivos por parte de los usuarios que no tienen permiso;

2. Estructura del software de E/S (V)

2.1. Niveles del software de E/S. (IV)

Software de E/S independiente del dispositivo.

Funciones:

4. ocultar el hecho de que los diferentes tipos de disco tengan distintos tamaños de sector, proporcionando un tamaño de bloque uniforme a los niveles superiores;
5. gestionar el almacenamiento temporal de los datos en las operaciones de E/S;
6. gestionar la asignación de espacio en dispositivos de bloques;
7. gestionar los dispositivos no compartibles, atendiendo las peticiones de utilización de dichos dispositivos y aceptándolas o rechazándolas dependiendo de la disponibilidad del dispositivo;
8. realizar un tratamiento del error independiente del dispositivo.

2. Estructura del software de E/S (VI)

2.1. Niveles del software de E/S. (V)

Software de E/S dependiente del dispositivo (manejador de dispositivo, gestor de periférico, Device Driver).

Está íntimamente relacionado con la estructura del dispositivo.

Su función es recibir peticiones abstractas de las rutinas independientes de los dispositivos y comprobar que dichas peticiones se realizan.

Pasos:

1. traducir la petición de términos abstractos a otros más concretos;
2. escribir en los puertos del controlador los comandos determinados;
3. detectar cuando la operación ha finalizado: si es controlada por interrupción, el manejador se bloquea hasta que la reciba, y si es controlada por programa el manejador no se bloquea;
4. comprobar si ha habido errores, devolviendo al nivel superior información del estado y del posible error.

2. Estructura del software de E/S (VII)

2.1. Niveles del software de E/S. (VI)

Software de E/S dependiente del dispositivo (manejador de dispositivo, gestor de periférico, Device Driver).

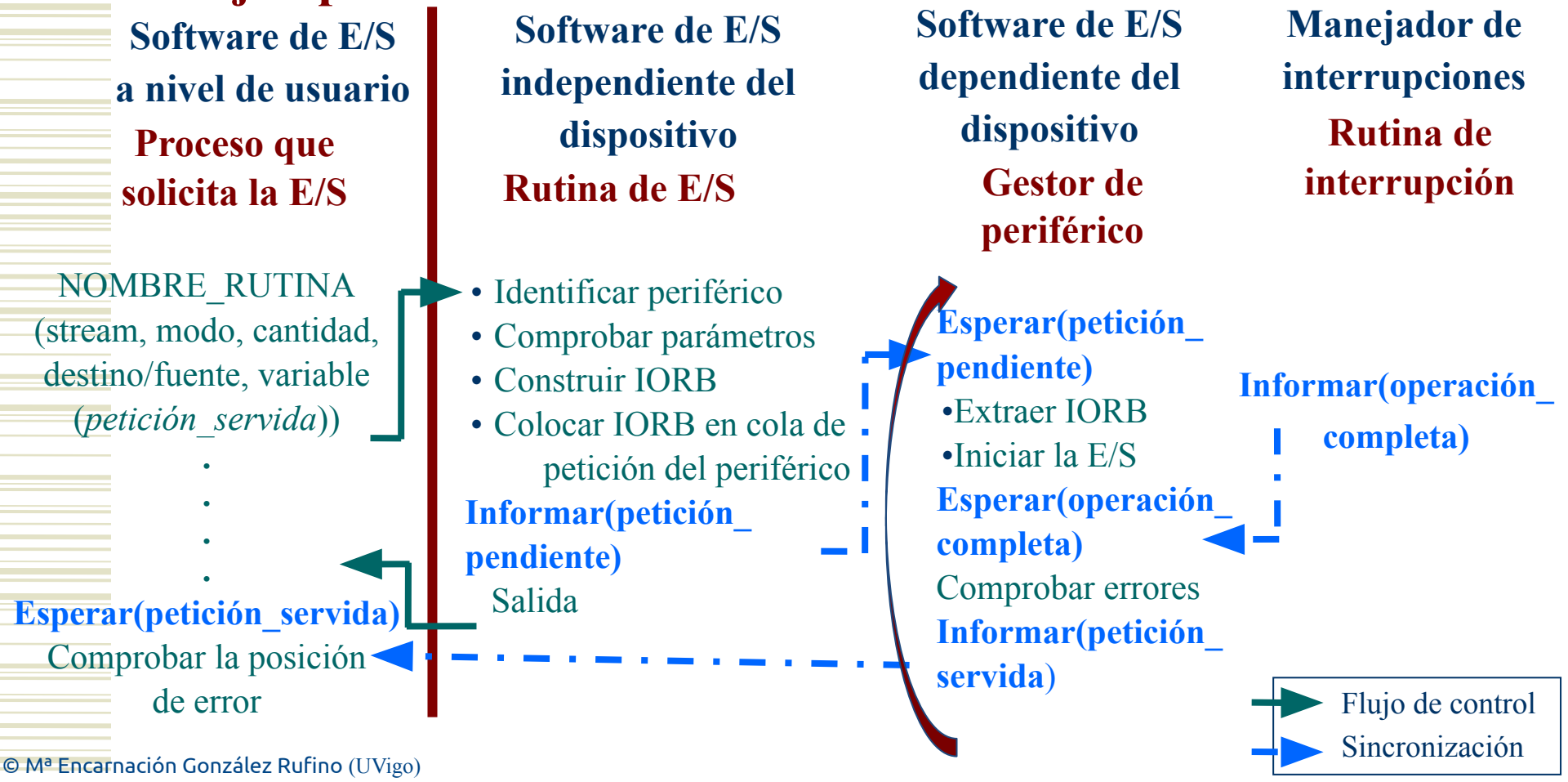
Si existe alguna petición pendiente, entonces se selecciona una y se sirve; si no hay, entonces el manejador queda esperando que le llegue la siguiente.

Manejador de interrupciones (Rutinas de tratamiento de interrupciones).

Procedimientos encargados de esperar la llegada de una interrupción y realizar las operaciones necesarias para desbloquear al manejador.

2. Estructura del software de E/S (VIII)

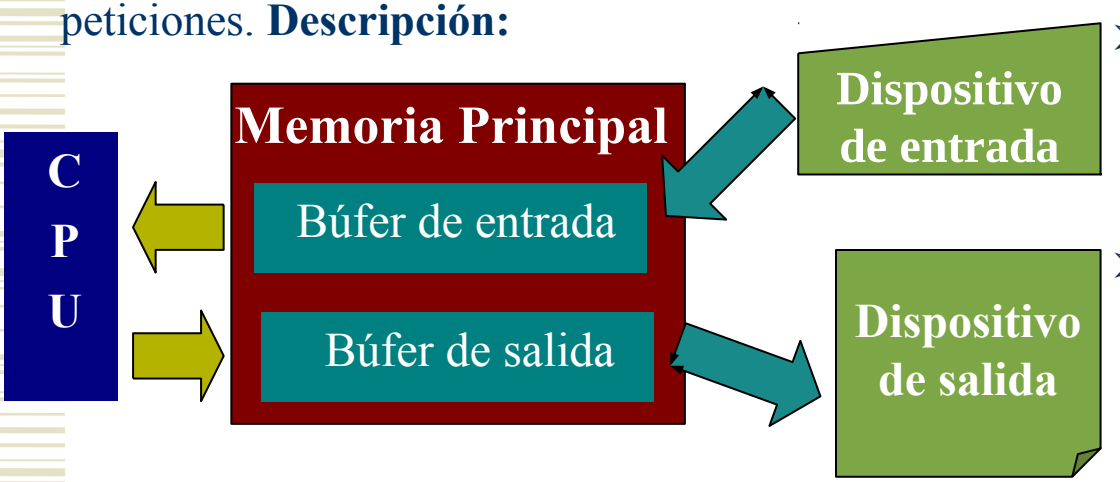
2.2. Ejemplo del funcionamiento de los distintos niveles.



2. Estructura del software de E/S (IX)

2.3. La técnica del "buffering".

Si la transferencia es directa, por cada petición de E/S el proceso se bloquea mientras que se realiza la operación sobre el periférico. La técnica del *buffering* pretende evitar estas pérdidas de tiempo llevando a cabo las transferencias de E/S antes de que el proceso realice las peticiones. **Descripción:**



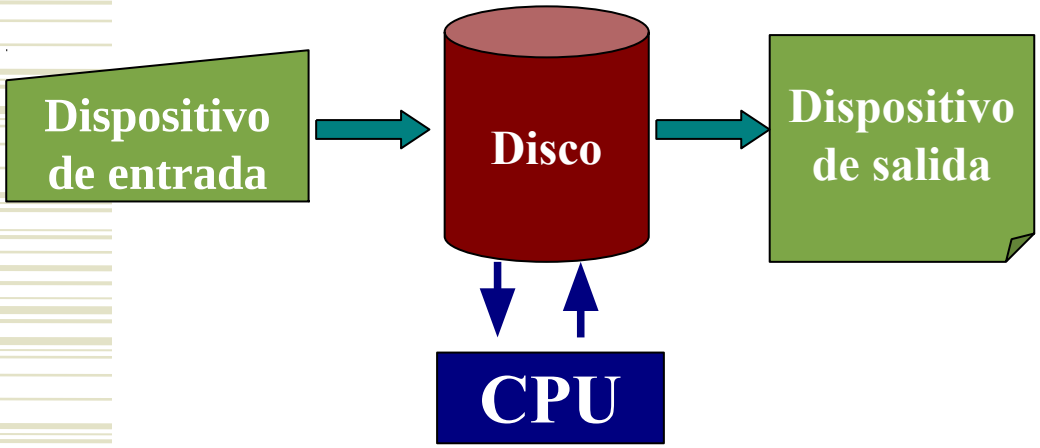
- el S. O. realiza las transferencias de entrada al búfer de entrada y el proceso de usuario toma los datos de dicho búfer;
- las transferencias de salida de un proceso de usuario se dirigen a un búfer de salida, siendo el S. O. el encargado de vaciarlo sobre el periférico.

Esta técnica no es útil si el proceso realiza sus E/S a una velocidad superior a la que pueden trabajar los periféricos de E/S. Es decir, se usará cuando el promedio de demandas de E/S de un proceso no es mayor que la que pueden atender los periféricos de E/S.

2. Estructura del software de E/S (X)

2.4. La técnica del “spooling”.

Cuando se usan dispositivos no compartibles puede ocurrir que durante periodos de mucha demanda varios procesos queden bloqueados esperando por el uso de estos periféricos. La técnica del *spooling* pretende evitar estas pérdidas de tiempo haciendo que la transferencia se efectúe sobre un soporte intermedio y no directamente sobre el periférico. **Descripción**:



- cuando un proceso abre un stream asociado a un dispositivo no compartible, la rutina de E/S le asigna un fichero anónimo en un soporte intermedio, de forma que se dirige a él todas las salidas del stream;
- cuando se cierra el stream, el fichero se añade a una cola que contiene ficheros similares;
- el *spooler* (proceso independiente asociado al dispositivo no compartible) se encarga de transferir la información de los ficheros almacenados en esa cola sobre el dispositivo físico.