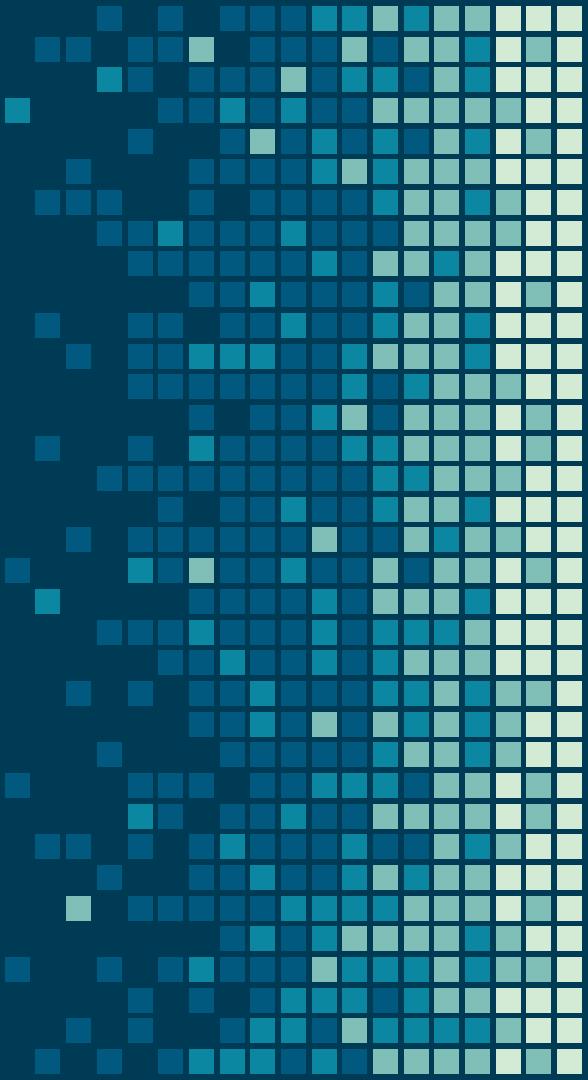
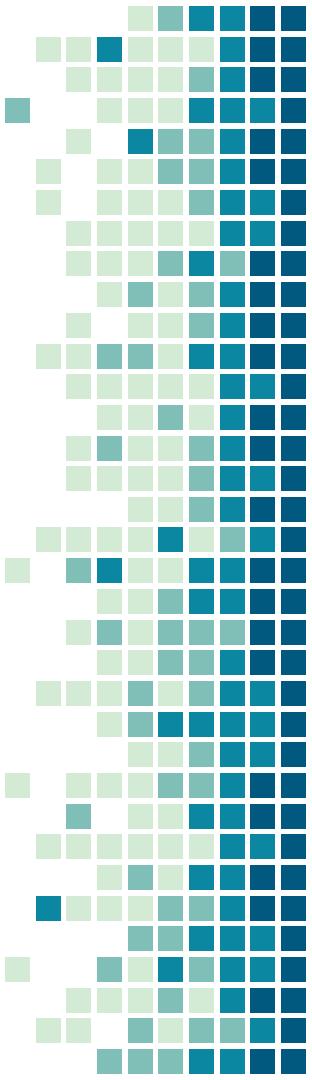


Administración de dispositivos

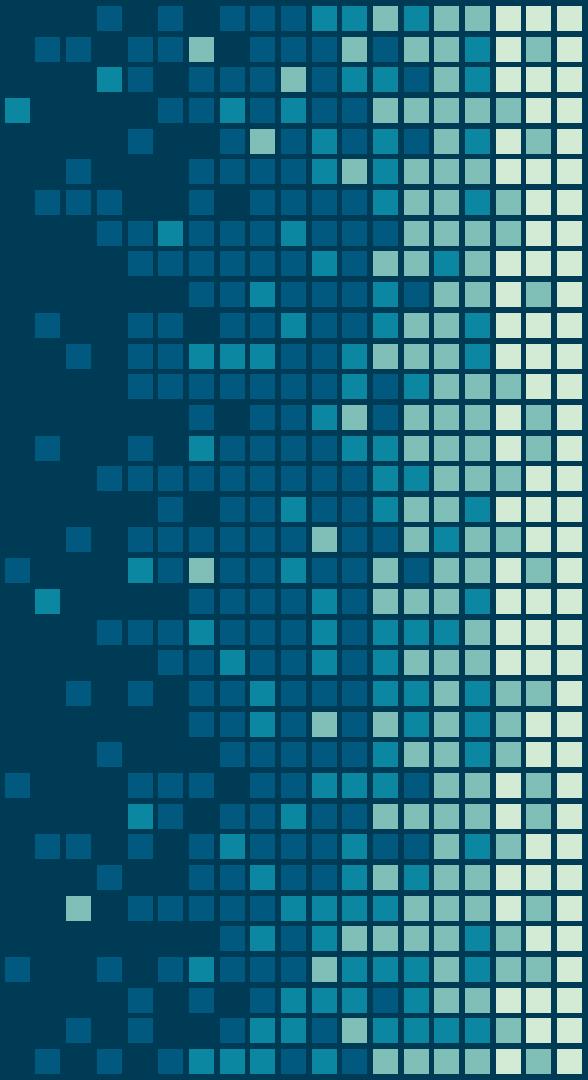


Agenda

- **Introducción**
- **Dispositivos de entrada y salida de datos**
- **Asignación de memoria para I/O**
- **Interrupciones y MDA**
- **Capas de Software**
- **Drivers**



Introducción



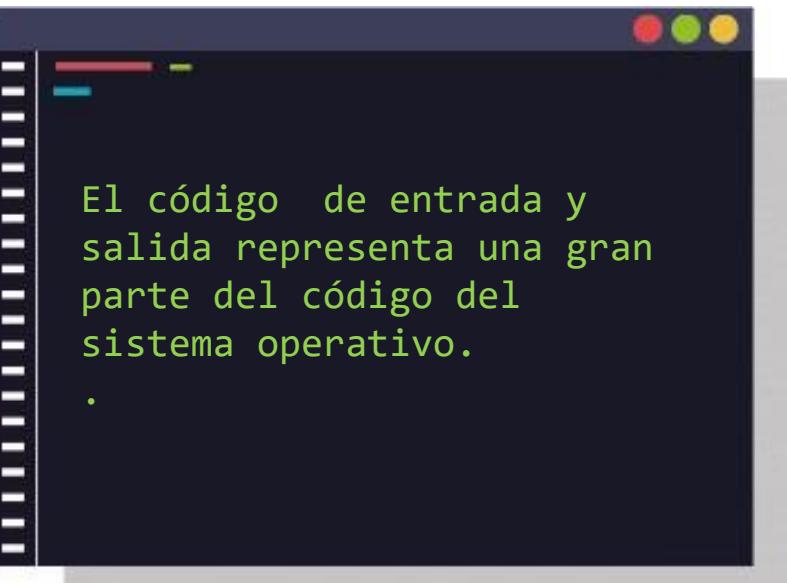
Introducción



- » ¿Cuáles son los dispositivos de un sistema computacional?
- » ¿Quién es el que controla los dispositivos de un computador?
- » El SO debe captar interrupciones y manejar errores.

Introducción

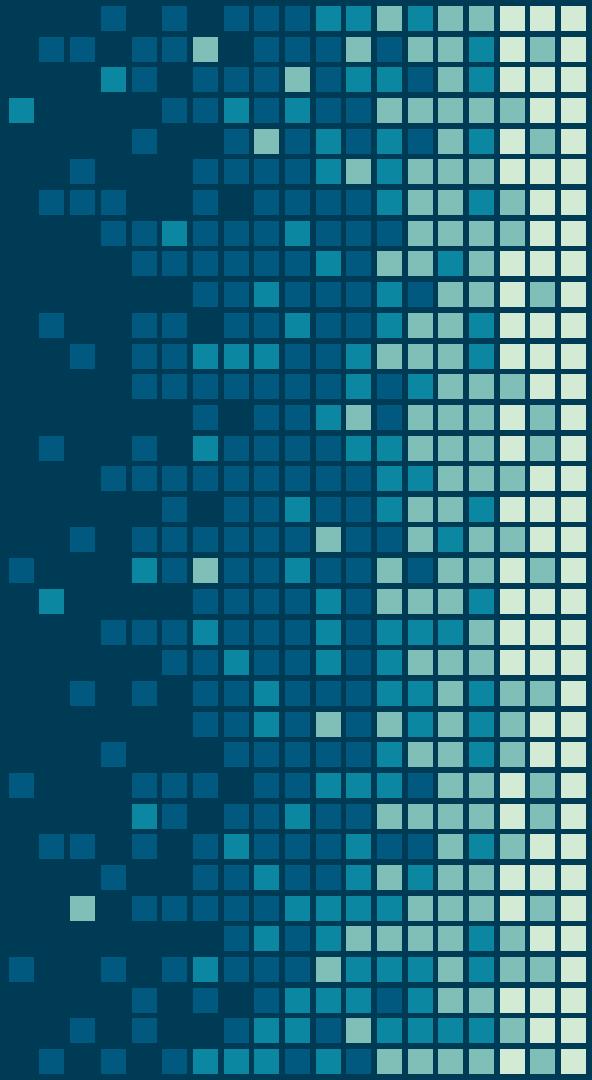
- El sistema operativo debe proporcionar una interfaz simple y fácil de usar entre los dispositivos y el resto del sistema.
- Lo ideal es que la interfaz debe ser igual para todos los dispositivos con el fin de que haya independencia entre ellos.



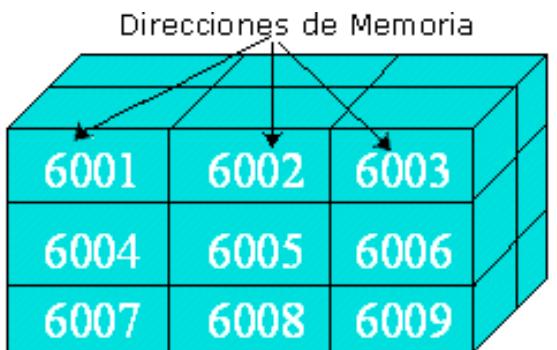
El código de entrada y salida representa una gran parte del código del sistema operativo.

.

Dispositivos de entrada y salida



Tipo 1: Dispositivo de bloque

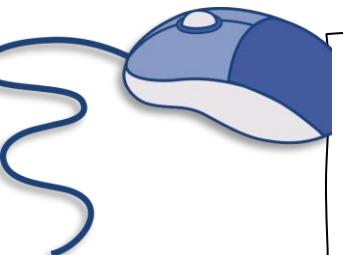


Almacena información en bloques de tamaño fijo.

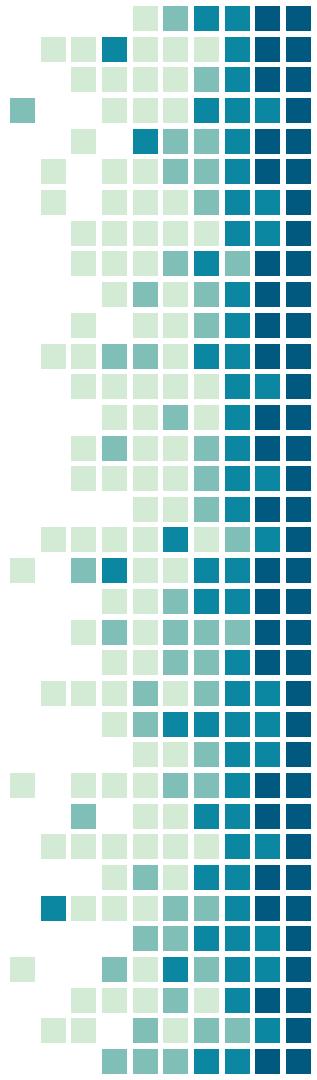
Esto con el fin de poder accederlos de manera independiente.

Las transferencias se pueden realizar entre uno y más bytes.

Tipo 2: Dispositivos de carácter.



- Este tipo de dispositivo envía o acepta un flujo de caracteres.
- No interesa la estructura del bloque ya que no es direccionable.
- Al no ser direccionable no tiene operación de búsqueda.
- La impresora y el mouse son ejemplos de este tipo de dispositivo.



Controlador de dispositivo

- ✓ Los dispositivos cuentan con un componente mecánico y otro electrónico, el componente electrónico es el controlador.
- ✓ Generalmente es un chip o un integrado que se conecta a una ranura.

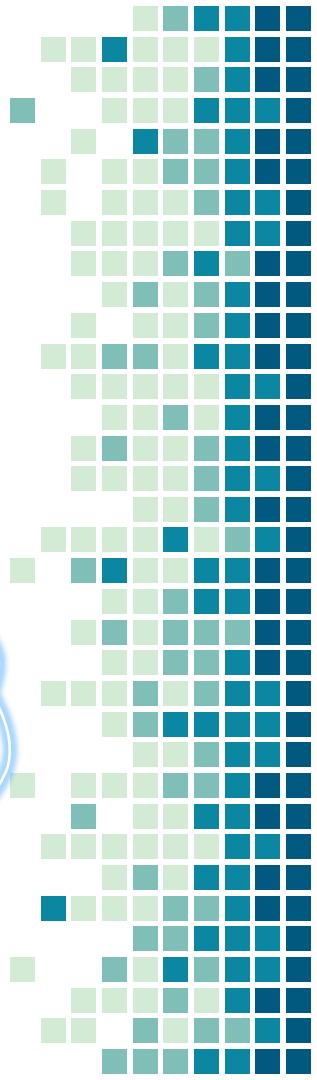


Controlador de dispositivo

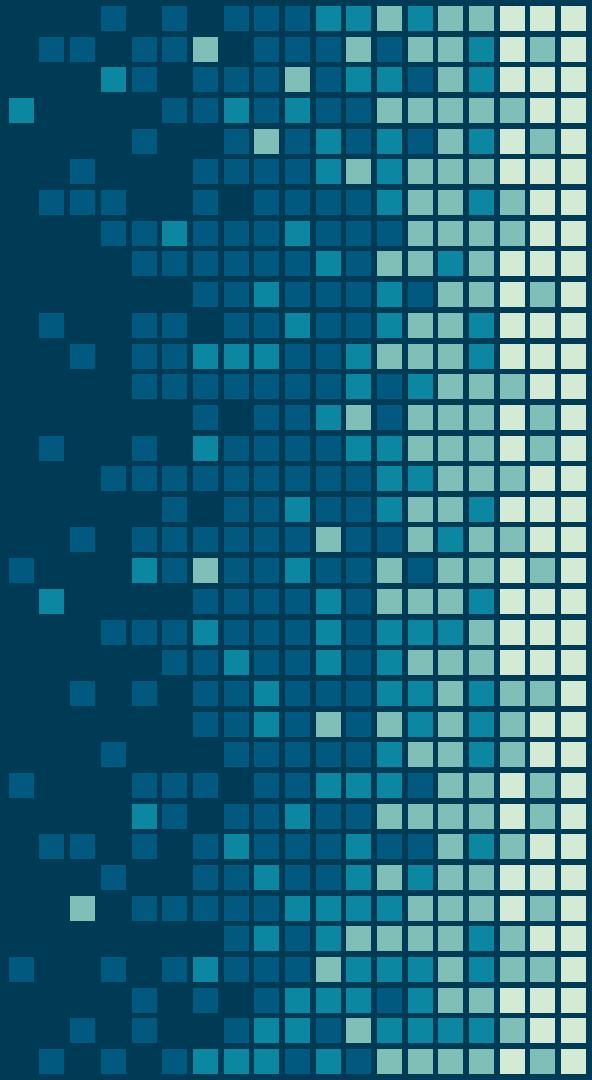
- > La interfaz con el controlador es de bajo nivel por la interacción con el hardware.



Por ejemplo ¿qué ocurre con el controlador de un disco? ¿Cuál es la función?
¿Controlador de un monitor ?

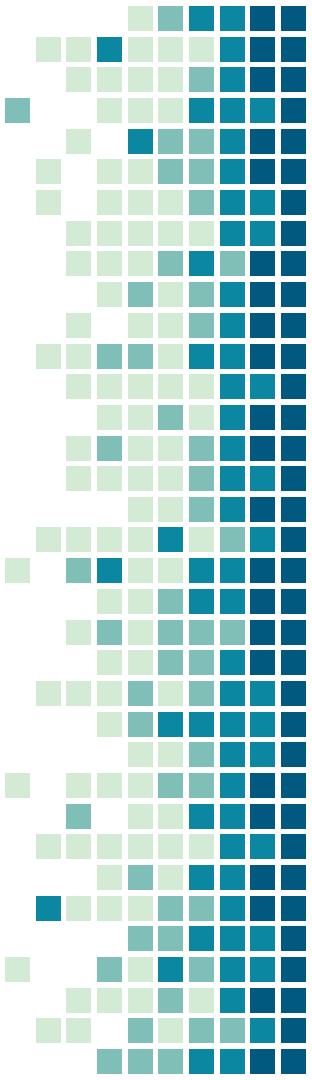


Entrada y salida por asignación de memoria



E/S por asignación de memoria.

Cada dispositivo contiene unos cuantos registros que se utilizan para comunicarse con el CPU. Al escribir en los registros, el SO puede hacer que el dispositivo envíe o acepte datos. El SO puede saber si está listo para alguna acción. Muchos de los dispositivos también contienen un buffer de datos de donde leer y escribir los datos.



E/S por asignación de memoria

¿Cómo se comunica el CPU con los registros de cada dispositivo?



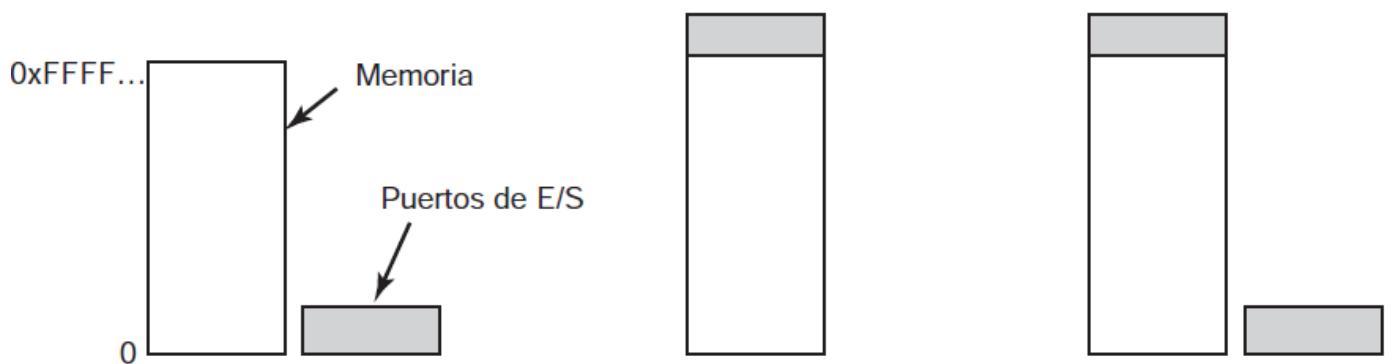
La primera alternativa es por medio de puertos de entrada y salida.



Segunda opción es por asignación de memoria.

E/S por asignación de memoria

- 👉 Para la primera opción, el espacio de direcciones de los puertos es diferente al espacio de memoria.
- 👉 Para la segunda opción, se asigna a cada registro de control un espacio de memoria.



Ventajas

- Para el enfoque de puertos se necesita código ensamblador por lo especial de cada registro, con memoria se podría hacer un controlador en C.
- En asignación de memoria, el mecanismo de seguridad está implícito, ya que es el sistema operativo es quien se encarga de esta función.
- Para la asignación de memoria no se requiere instrucciones especiales ya que todo está en un único espacio de direcciones.



Desventajas



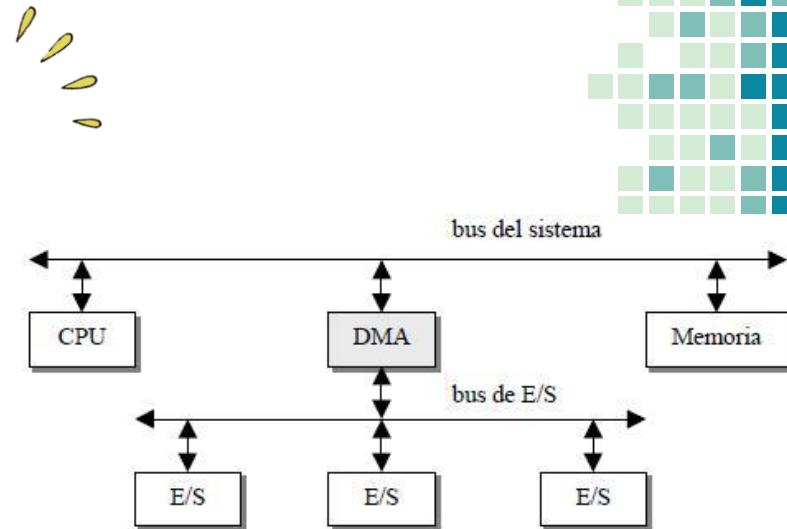
- Asignación de datos en caché, sería desastroso si los registros se colocan en caché, ya que sólo se tomaría la primera referencia (se debe deshabilitar la caché de manera selectiva).
- Si hay un solo espacio, todos los dispositivos deben de examinar todas las referencias a memoria para ver a cual responder.

Acceso directo a memoria (DMA)

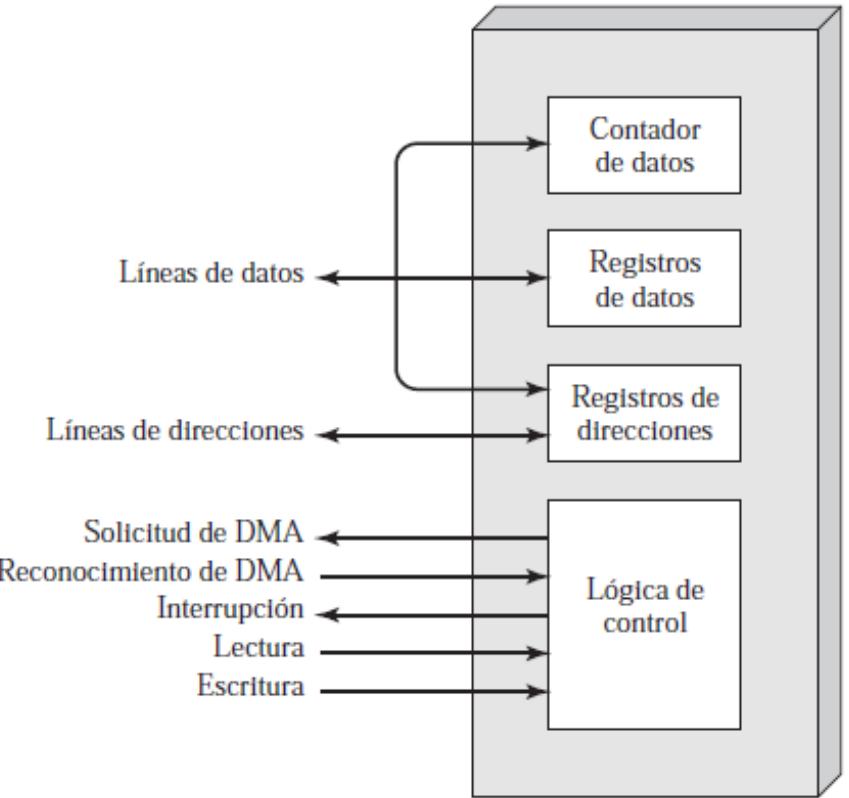


Acceso directo a memoria (DMA)

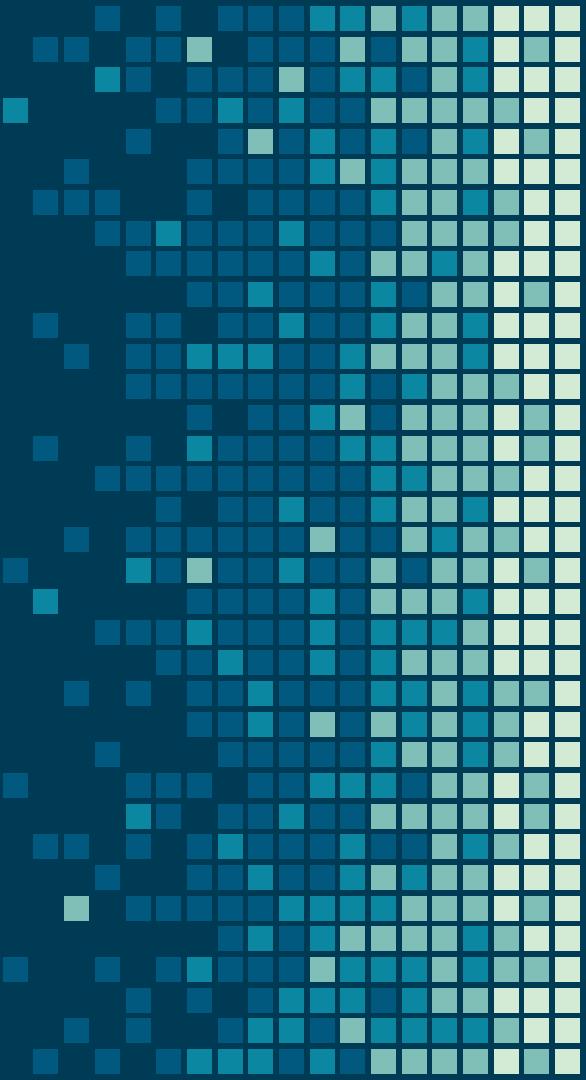
- Definitivamente los dispositivos se deben de comunicar con el CPU.
- La idea fundamental es transferir datos como conjuntos de bits y no de uno en uno.
- El SO puede utilizar DMA si el hardware tiene un controlador que soporte DMA.
- Generalmente se utiliza cuando se requiere transferir alto volumen de datos.



¿Cómo funciona el módulo DMA?



E/S e interrupciones

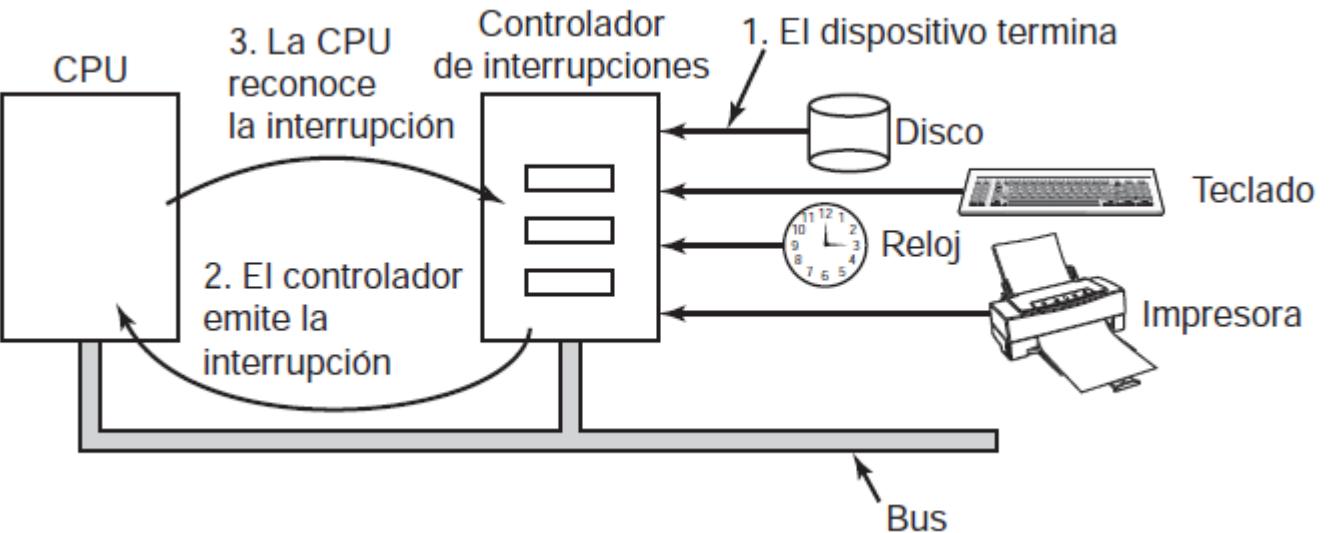


Interrupciones

- Ciertos dispositivos, como el mouse o teclado realizan su trabajo por medio de interrupciones, cuando requieren atención envían una señal en el controlador de interrupciones.
- Si no hay interrupciones pendientes se ejecuta la instrucción en otro caso deberá esperar a ser atendido.
- Entra en juego el vector de interrupciones.



Interrupciones



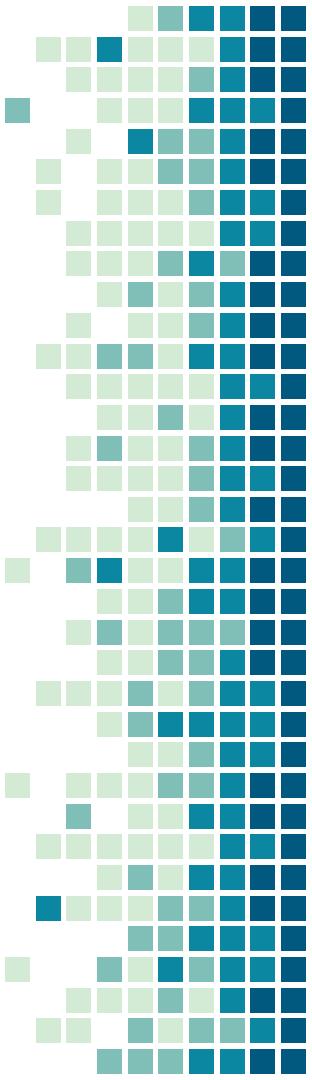
Software de E/S



Objetivos del software de E/S



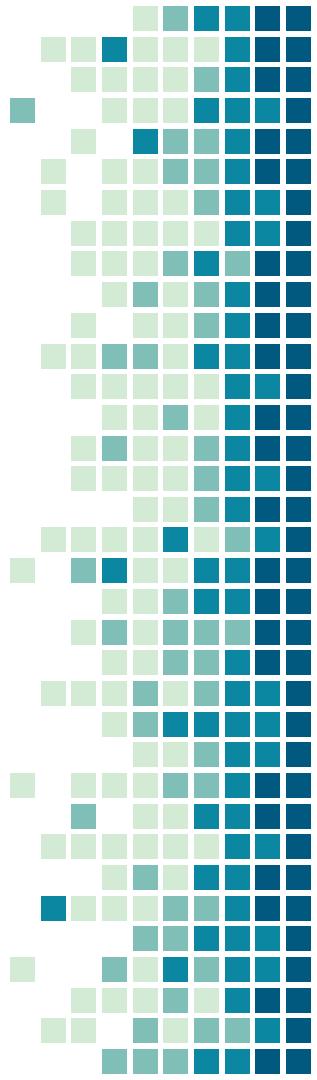
- El objetivo principal de crear este tipo de aplicaciones es obtener independencia de dispositivos.
- Se debe de hacer programas que puedan acceder a cualquier dispositivo de E/S sin tener que especificar el dispositivo por adelantado.
- Denominación uniforme, se refiere a que el nombre de los archivos o dispositivos son independientes del dispositivo en si.



Objetivos del software de E/S



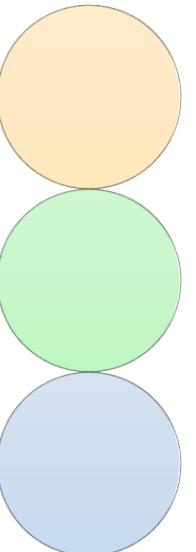
El manejo de errores se debe de llevar a cabo lo más cerca del hardware, ya que se puede corregir por si mismo. En la mayoría de las ocasiones es sólo volver a escribir o leer.



Maneras de llevar a cabo la E/S de datos



Maneras de llevar acabo E/S de datos

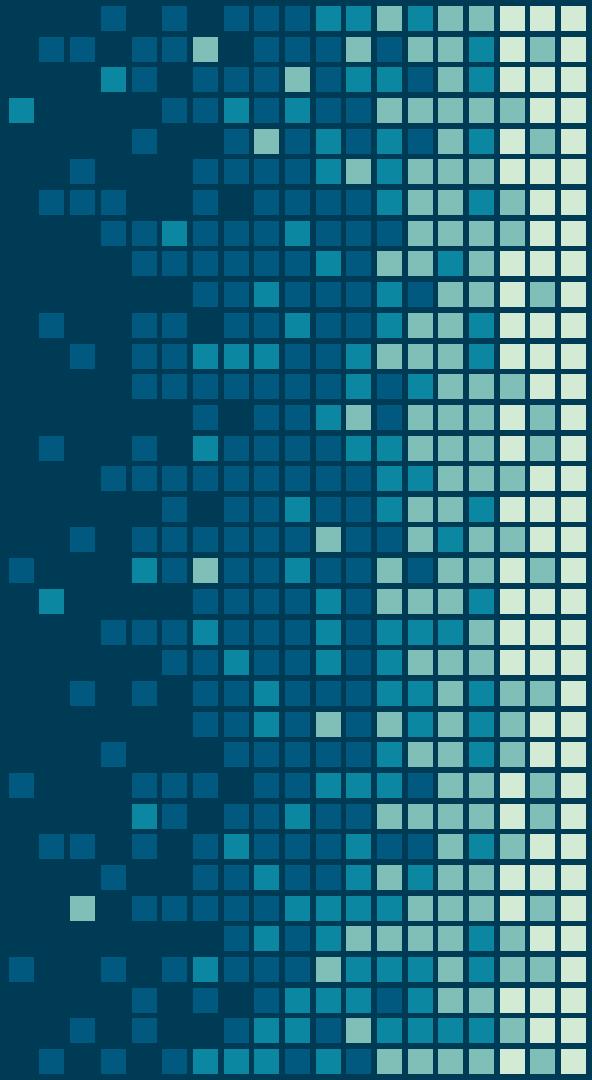


E/S programada.

E/S controlada por interrupciones.

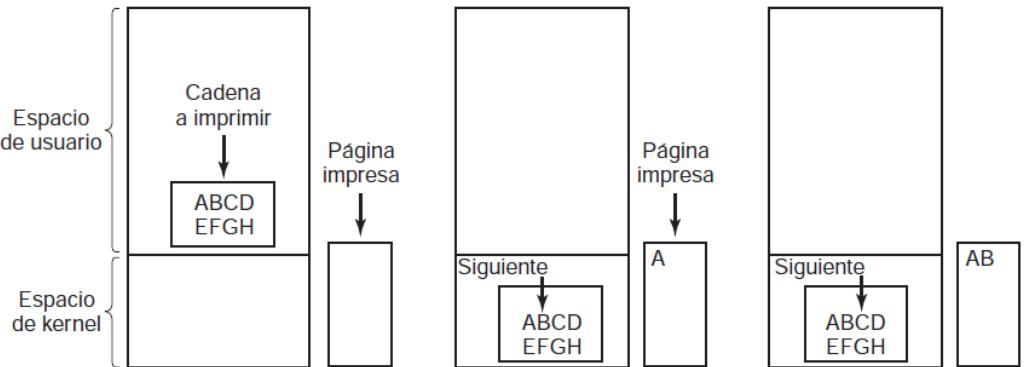
E/S mediante DMA.

E/S programada



E/S programada.

Es el método más sencillo.
Consiste en que el CPU
realice todo el trabajo.
Supongamos que se
quiere imprimir un
documento por este
método.



E/S programada.

La gran ventaja de este método radica en la simplicidad

La mayor desventaja es que ocupa el CPU tiempo completo hasta que se completen todas las operaciones de E/S.



E/S controlada por interrupciones



E/S controlada por interrupciones.



Si una impresora dura 1 segundo en escribir 100 caracteres, significa que un carácter dura 10 ms para imprimirse.



¿Qué pasa con el procesador durante esos 10ms?

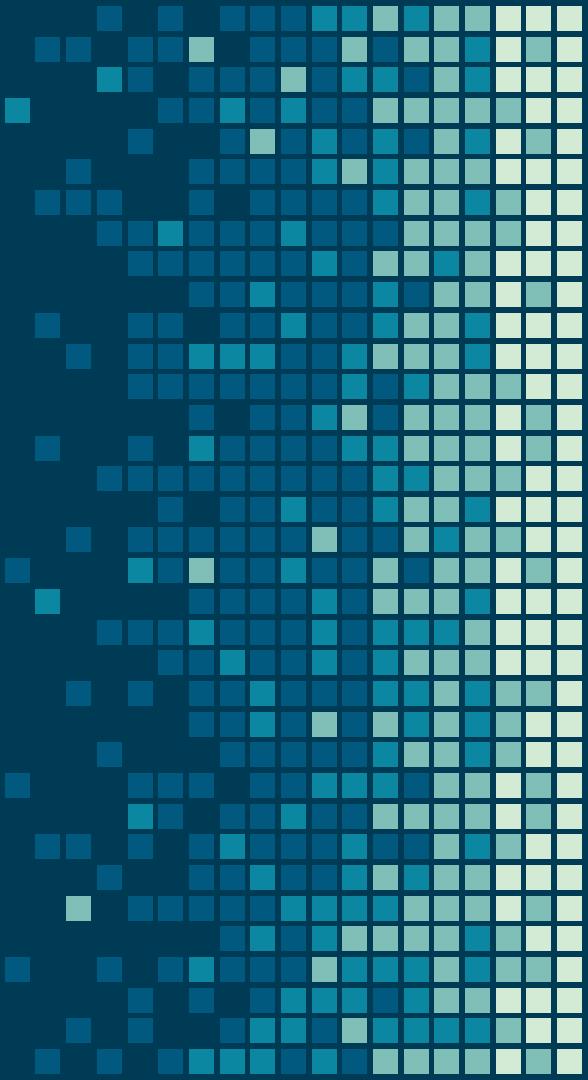
- La idea fundamental de las interrupciones es que el procesador ejecute otros procesos mientras que la E/S se completa.

E/S controlada por interrupciones.



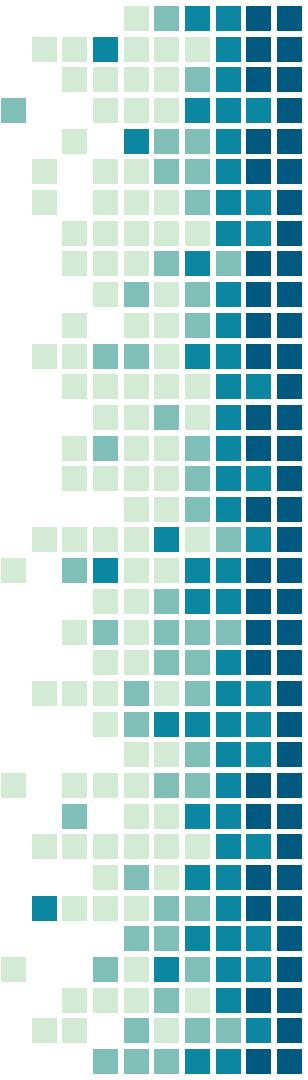
- Una de las desventajas de interrupciones es que si son muy frecuentes se puede traducir a mayor tiempo de ejecución, ya que atender las interrupciones conlleva un tiempo asociado.

E/S mediante el uso de DMA

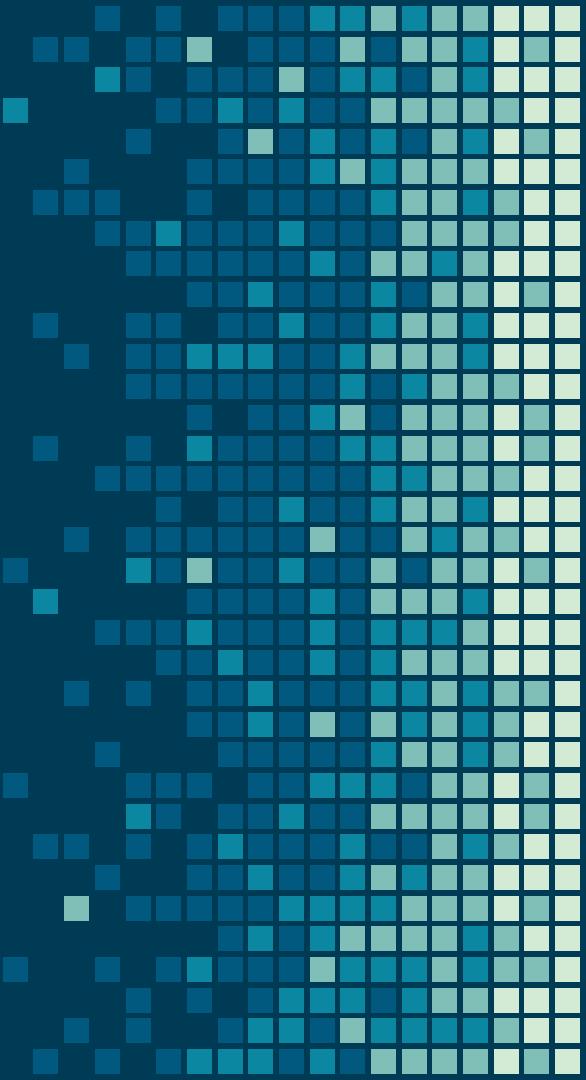


E/S mediante el uso de DMA

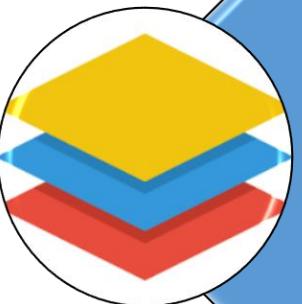
- La idea fundamental del DMA es disminuir la cantidad de interrupciones y con ello el tiempo de respuesta.
- La unidad del DMA realiza el trabajo del CPU, con el fin de que este último se dedique a otros procesos.



Capas del software de E/S



Capas de software de E/S

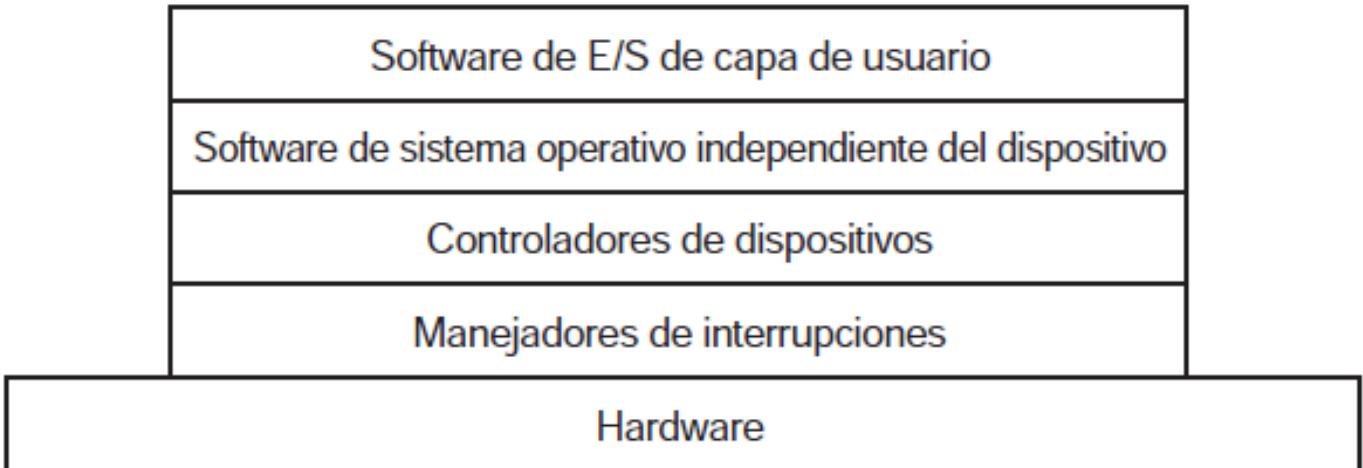


Por lo general, el software de E/S se organiza en cuatro capas.

La funcionalidad y las interfaces entre las capas varían de acuerdo con la versión y tipo de sistema operativo.

Las capas de software que provee el sistema no es específico para un dispositivo

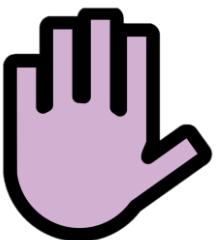
Capas de software de E/S



Manejador de interrupciones



Manejador de interrupciones



 En general las interrupciones son un hecho incómodo para el sistema operativo, ya que alteran el flujo normal de la ejecución de un programa.

 Se debe de ocultar de tal manera que el SO trate lo menos posible con las interrupciones.

 La mejor manera de realizar esto, es bloqueando los dispositivos hasta que se atienda las interrupciones de hardware previas.

Drivers



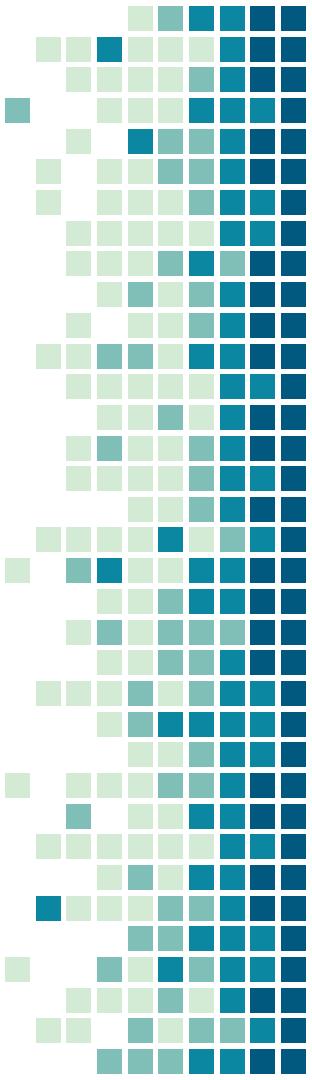
Drivers

Se utilizan para dar comandos a los dispositivos, leer el estado de los registros o ambas, es decir, leer y escribir información al hardware. El driver está ligado al SO y dispositivo por lo que cada uno será diferente. Un driver de teclado no servirá para un mouse.

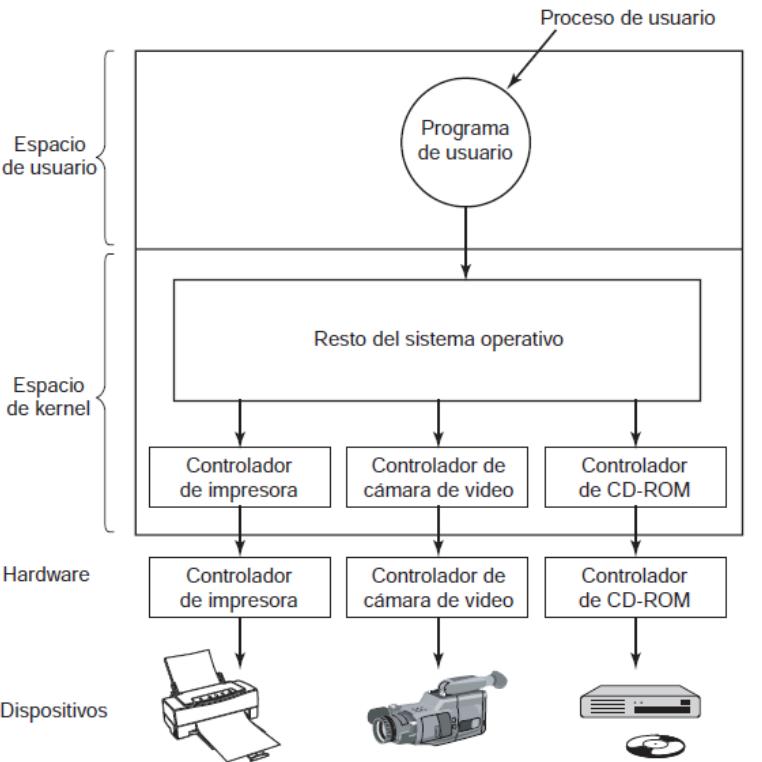


Drivers

- >Formalmente, un driver es el código necesario para controlar un dispositivo.
- >Para utilizar el hardware del dispositivo, el driver debe estar en el kernel del SO, sin embargo, si el sistema provee llamadas para acceder al bus de datos también se podría realizar en espacio de usuario.
- >El SO debe de proporcionar un modelo para instalar drivers.



Drivers



Funciones de los drivers



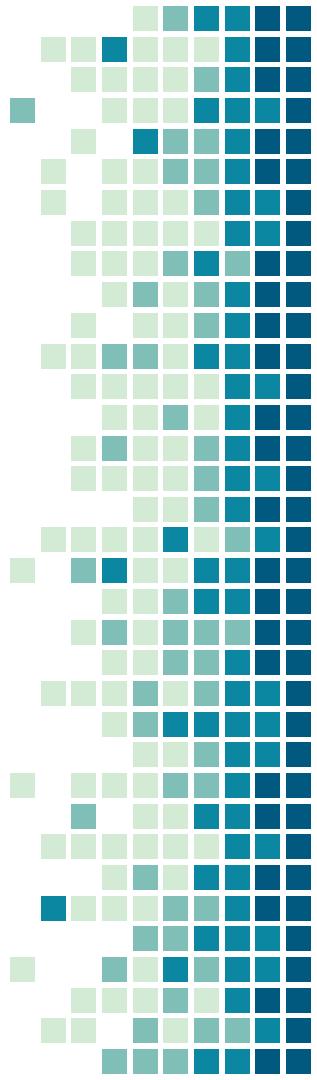
Funciones específicas de los drivers

Aceptar peticiones abstractas de lectura y escritura del software independientemente del dispositivo.

Se encarga de ejecutar las rutinas de inicialización de los dispositivos, en caso de que sea necesario.

Dar formato a los datos, por ejemplo los discos duros.

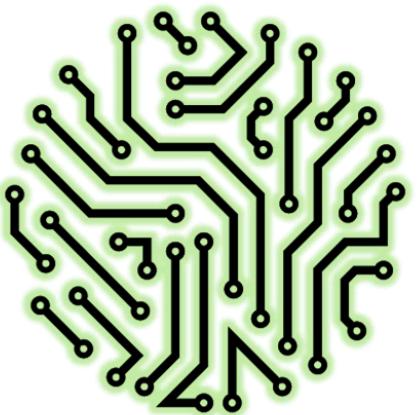
Comprueba si el dispositivo está en uso.



Software E/S independiente del dispositivo

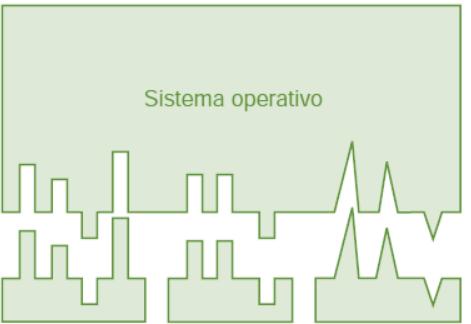


Software de E/S independiente del dispositivo



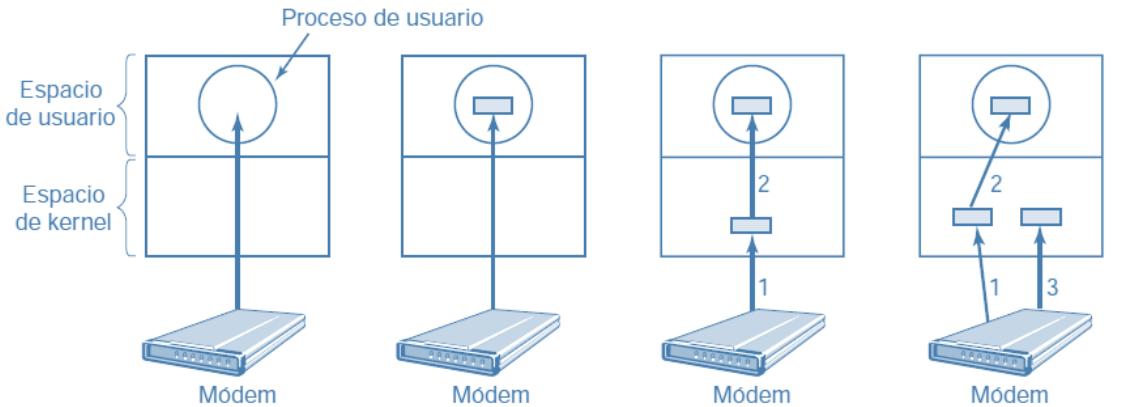
Aunque parte del software de E/S es específico para cada dispositivo, otras funciones se pueden factorizar y hacerlas independientes del hardware.

Interfaz uniforme para los controladores de software de dispositivos



La idea es que todos los controladores se puedan tratar de manera similar.
No es conveniente tener que modificar el sistema operativo para cada nuevo dispositivo.

Uso del buffer



La idea fundamental es llenar el buffer de información y levantar una interrupción cuando ya esté lleno, con el fin de copiarlo al espacio de usuario.



Manejo de errores

Se da cuando un proceso solicita algo imposible ya sea por un error lógico o físico.

En la mayoría de los casos la acción a tomar es sólo reportarlo con un código de error al que hizo la llamada.

Cuando el controlador no sabe que hacer, generalmente trata de volver hacer la tarea y si no entonces pasa el problema a un nivel más alto.



Asignación y liberación de dispositivos dedicados

- Es responsabilidad del sistema operativo decidir quien puede utilizar los dispositivos de acuerdo con la disponibilidad de cada uno de ellos.



Referencias

- Tanenbaum, A. S. (2015).
Sistemas operativos modernos. Pearson Educación
- Stallings, W. (2008).
Sistemas operativos. Martin Iturbide.
- Calvaresi, D., Marinoni, M., Sturm, A., Schumacher, M., & Buttazzo, G. (2017, August).
The challenge of real-time multi-agent systems for enabling IoT and CPS.
In *Proceedings of the international conference on web intelligence* (pp. 356-364). ACM.

¿Preguntas?

Realizado por: Jason Leiton Jiménez.

Tecnológico de Costa Rica
Ingeniería en Computadores
2024

TEC