

Entrada / Salida Repaso

Sistemas Operativos

May 5, 2023

Categorías de dispositivos de E/S

- Dispositivos por bloque (*block devices*).
- Dispositivos por caracter (*character devices*).

Drivers Módulos de software que pueden ser añadidos al SO para manejar los dispositivos de E/S.

Controllers Componente mecánico y/o electrónico que trabaja como una interfaz entre un dispositivo y el driver.

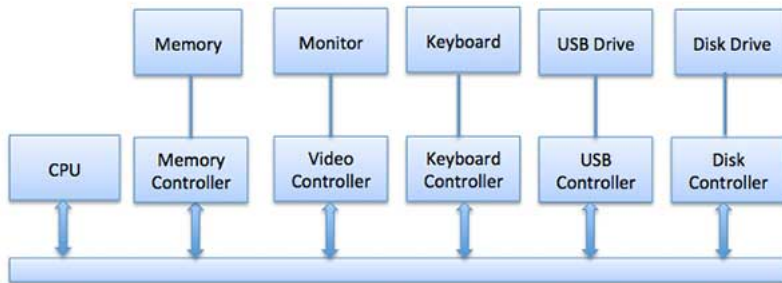


Figura 1: Controladores de hardware (*Controllers*)

E/S Asíncrona vs síncrona

Síncrona La ejecución de la CPU que solicita la E/S, espera por su culminación.

Asíncrona Cada E/S procede concurrentemente con la ejecución del CPU que la solicita.

Comunicación a los dispositivos de E/S

- E/S con instrucciones especiales.
- E/S mapeada en memoria.
- Acceso directo a memoria (DMA).

Comunicación: Instrucciones Especiales



Figura 2: Comunicación por instrucciones especiales

Comunicación: E/S Mapeada en memoria

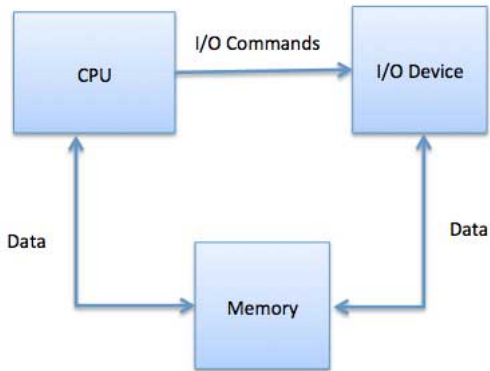


Figura 3:E/S Mapeada en memoria

Comunicación: E/S con DMA

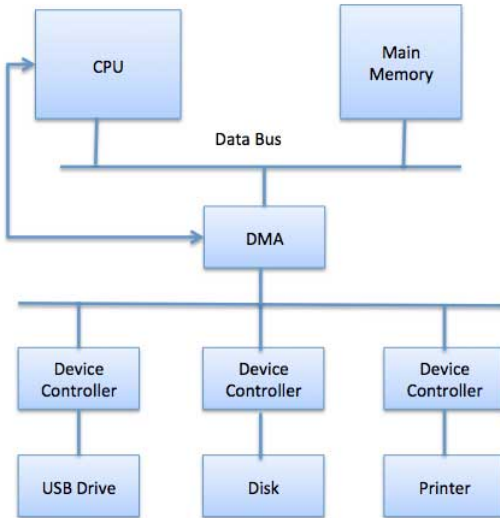


Figura 4:E/S con DMA

Polling vs Interrupciones

- Vías para detectar la llegada de cualquier tipo de entrada.
- Las dos técnicas le permiten a la CPU atender los eventos que suceden en cualquier momento, y que no están relacionados a los procesos en ejecución.
- Polling:
 - Más simple.
 - Revisión periódica del estatus del dispositivo.
 - Utiliza un registro de estatus.
- Interrupciones:
 - El *controller* coloca una señal en el bus, cuando necesita atención de la CPU.
 - Utiliza los *handlers* o rutinas de software para manejar cada tipo de interrupción.

- **Bibliotecas a nivel de usuario:** Ejemplo, **stdio** del lenguaje C y C++.
- **Módulos de nivel del Kernel:** Son los drivers.
- **Hardware:** Por ejemplo, los *firmware*.

Software para E/S

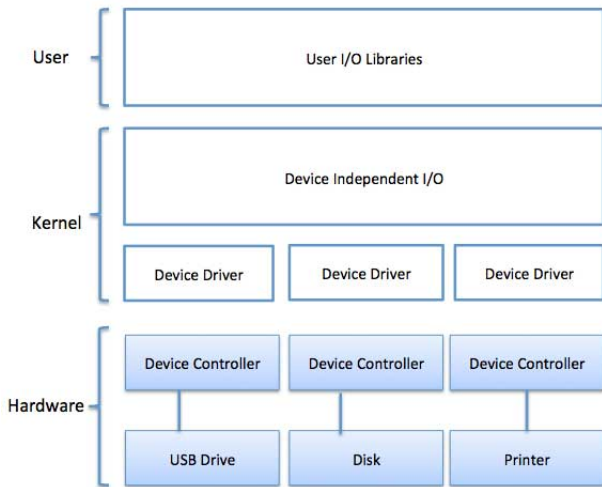


Figura 5:Software para E/S

- Acepta solicitudes del software independiente del dispositivo, que a su vez, las recibe del nivel de usuario.
- Interactúa con el *controller* para recibir o enviar una E/S.
- Maneja los errores respectivos.
- Se asegura que la solicitud de ejecute exitosamente.

Handlers de interrupciones

- Conocidas como rutinas de servicios de interrupciones o **ISR** (por su siglas en inglés).
- Específicamente se les denomina *callback functions*.
- Se alojan en el *driver*.
- Se ordenan numéricamente, a lo que se le denomina direcciones.

Software de E/S independiente del dispositivo

- Hace que la interacción con los distintos drivers sea uniforme.
- Utiliza un nombramiento de dispositivos: nombres mnemotécnicos mapeados números de dispositivos denominados Major y Minor.
- Protección de dispositivos.
- Permite configurar el tamaño de bloque.
- Servicio de *buffering*.
- Reserva el almacenamiento en lo *block devices*.
- Reserva y libera dispositivos dedicados.
- Reporte de errores.

Software a nivel de usuario

- Interfaz simplificada.
- Consiste en procedimientos/funciones alojados en bibliotecas.
- Están en el espacio de usuario (**stdio**).

- Scheduling.
- Buffering.
- Caching.
- Spooling.
 - Un *spool* es un buffer que mantiene los datos enviados a un dispositivo.
 - No acepta envíos intercalados, solo solicitudes completas.
 - Típicamente usado para el servicio de impresión.
 - Mantiene una cola de archivos enviados. Se atienden uno a la vez.
- Manejo de errores.