

Tema: Gestión de Entrada / Salida

Objetivo:

- Analizar el funcionamiento y manejo de los dispositivos de entrada/salida en un Sistema Operativo.

Introducción

Evolución de las funciones de la E/S

1. Inicialmente el procesador controlaba directamente los dispositivos periféricos
2. Posteriormente, cuando un controlador o módulo de E/S se añadía a un sistema
 - El procesador utiliza E/S programada sin interrupciones
 - El procesador comenzó a aislarse de las particularidades de los dispositivos (drivers o controladores)
3. Comienza el uso de interrupciones
 - El procesador no gasta tiempo en esperar la finalización de las operaciones
4. Posteriormente los dispositivos comenzaron a controlar el acceso a la memoria con la aparición de la DMA
 - Ahora podían mover un bloque de datos a la memoria o desde la misma sin que intervenga el procesador
5. El módulo de E/S evoluciona a ser un procesador separado con su propio juego de Instrucciones específicas para E/S
 - Esto permite al procesador determinar un conjunto o secuencia de instrucciones (operaciones a realizar) y será interrumpido solamente cuando haya terminado la secuencia entera
6. El módulo de E/S posteriormente evoluciona a ser un computador independiente con su propia memoria local

Caracterización de los dispositivos de E/S

Todo los dispositivos instalados en una máquina pueden ser accedidos a través de un interfaz de ficheros bajo el directorio `/dev/` . Allí podemos encontrar los dispositivos conectados a las controladoras IDE (hda, hdb, hdc y hdd), el dsp de la tarjeta de sonido, etc.

Los ficheros asociados a un dispositivo pueden ser de dos tipos, de caracter y de bloque . Si ejecutamos un `ls -l /dev`, la primera columna nos indica el tipo de dispositivo, ```c"` para dispositivos de caracter y ```b"` para dispositivos de bloque.

Los dispositivos de caracter son accedidos secuencialmente, un caracter cada vez. Algunos ejemplos de dispositivos de caracter son el ratón, el teclado, un terminal de texto, una cinta magnética, null, etc.

Los dispositivos de bloque se caracterizan por ser de acceso aleatorio, la unidad mínima de lectura-escritura no es un caracter, sino un bloque (1KB). Algunos ejemplos de dispositivos de caracter son los discos duros, los disquetes, los CDROMS, etc.

Técnicas de E/S

- **E/S programada (polling o encuesta)**
 - El procesador está ocupado esperando a que la operación finalice
 - Espera activa
- **E/S dirigida por interrupciones**
 - Se emite una orden de E/S
 - Continúan la ejecución de instrucciones siguientes
 - El módulo de E/S envía una interrupción cuando finaliza la operación
- **Acceso directo a memoria (DMA)**
 - El módulo DMA controla el intercambio de datos entre la memoria principal y el dispositivo de E/S
 - El procesador envía una petición de transferencia al módulo DMA y es interrumpido cuando se ha transferido el bloque entero

Hardware de E/S

Elementos hardware del sistema de E/S y arquitectura:

- ❖ Bus
- ❖ Puerto
- ❖ Controlador (driver)
- ❖ Dispositivo

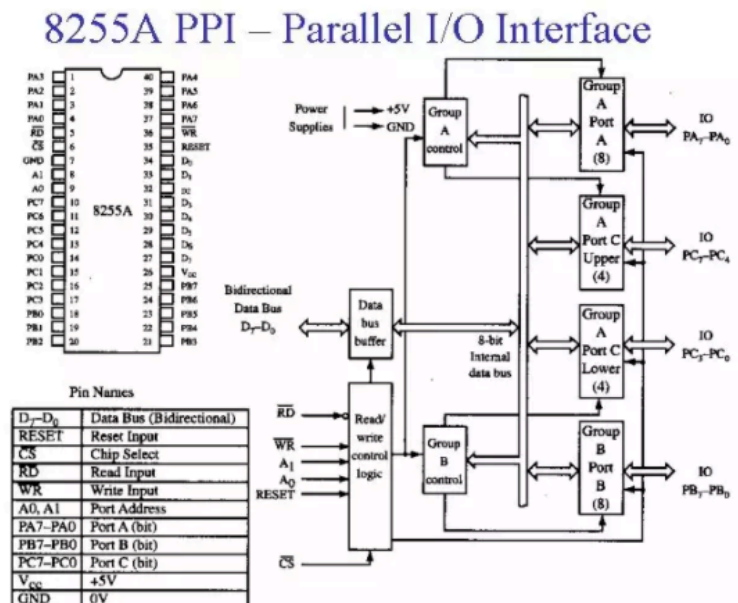


Figura 1. Puertos en un sistema computacional

Bus es una vía eléctrica sobre la cual viajan señales eléctricas, de datos y otras. Usted puede pensar de ellos como conexiones eléctricas que permiten a la CPU recibir señales y enviar una respuesta con base al software almacenado en memoria. En los computadores personales, el término bus se refiere a las vías de acceso entre los componentes de un computador. Existen dos tipos de buses en una computadora; el bus de datos y el bus de direcciones.

Puertos. Se encargan de transmitir una serie de datos entre dos extremos, todos ellos tienen un reloj y pines.

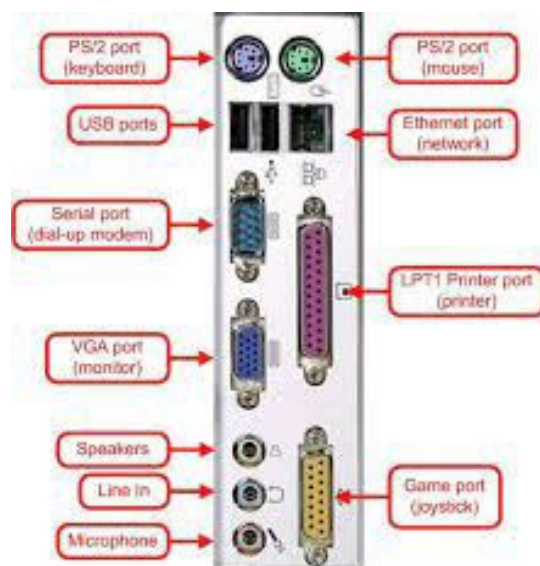


Figura 2. Puertos E/S

Un reloj: Esta señal marca cada cuanto se realiza una transferencia de datos.

Pines de datos: Los pines que transmiten la información de un lado a otro, hay pines de recepción, pines de envío y Full Duplex que permiten que los datos vayan en un sentido u otro. Si hay varios pines de datos en la interfaz del puerto diremos que es un puerto paralelo, si hay pocos pines entonces diremos que es un puerto de serie.



Figura 2. Puertos USB

Controlador o driver. Es más que un programa que le dice al kernel como se usa un determinado dispositivo de entrada / salida. El driver informa al kernel de los registros que configuran al dispositivo, como se direccionan, en que formato espera recibir los datos a imprimir, etc.

Comunicación con el hardware de E/S:

- Sondeo
- Interrupciones
- DMA

Arquitectura hardware del sistema de E/S

El hardware asociado con un dispositivo de E/S consta de cuatro elementos básicos:
ver figura 1

1. Un *bus* para comunicarse con la CPU y es compartido entre varios dispositivos.
2. Un *puerto* que consta de varios registros:
 - ❖ **Estado**-indica si esta ocupado, los datos están listos, o ha ocurrido un error.
 - ❖ **Control**-operación que ha de realizar.
 - ❖ **Datos_entrada** - datos a enviar a CPU.
 - ❖ **Datos_salida** - datos recibidos de la CPU.
3. Un *controlador* que recibe ordenes del bus del sistema, traduce ordenes en acciones del dispositivo, y lee/escribe datos desde/en el bus del sistema.
4. El propio *dispositivo*.
 - ❖ Existen una gran variedad de dispositivos:
 - ⚙ Tradicionales: discos, impresoras, teclado, modem, ratón, pantalla.
 - ⚙ No tradicionales: joystick, actuador de robot, superficie de vuelo de un avión, sistema de inyección de un coche, etc.

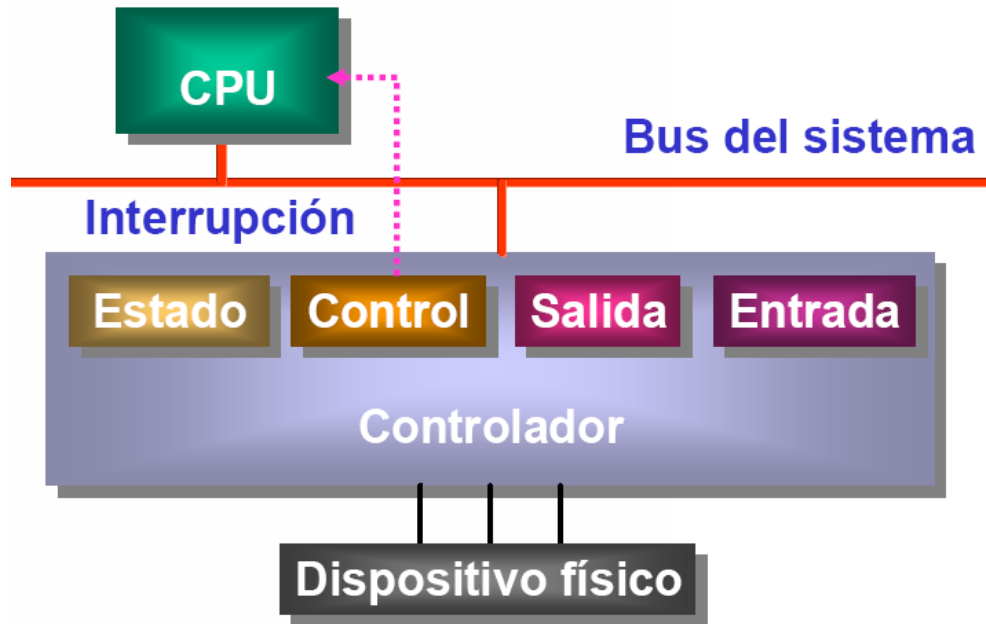


Figura 3. Esquema de Arquitectura General

Comunicación entre SO y dispositivo de E/S: Sondeo

Los pasos a seguir al usar sondeo son:

1. CPU espera hasta que el estado sea libre.
2. CPU ajusta el registro de ordenes y datos-salida, si la operación es de salida.
3. CPU ajusta el estado a orden-preparada.
4. El controlador reacciona a orden preparada y pone estado a ocupado. Lee registro de ordenes y ejecuta orden, pone un valor en datos-salida, si es una orden de salida.
5. Suponiendo que la orden tiene éxito, el controlador cambia el estado a ocioso.
6. La CPU observa el cambio a ocioso y lee los datos si es una operación de salida.

Es buena elección si los datos van a ser manejados al instante (un modem o teclado), ya que los datos se perderían si no se retiran del dispositivo lo suficientemente rápido; pero ¿y si el dispositivo es lento comparado con la CPU?

Comunicación entre SO y dispositivo: interrupción

En lugar de tener la CPU ocupada esperando la disponibilidad del dispositivo, el dispositivo interrumpe a la CPU cuando ha terminado una operación de E/S.

Cuando se produce la interrupción de E/S:

- ⚙ Determinar que dispositivo la provocó.
- ⚙ Si la última orden fue una operación de entrada, recupera los datos del registro del dispositivo.
- ⚙ Inicia la siguiente operación para el dispositivo.

Acceso directo a memoria (DMA)

La CPU recupera la información byte a byte, no adecuado para grandes volúmenes.

DMA (Direct Memory Access) – Controlador de dispositivo que puede escribir directamente en memoria. En lugar de registros de e/s, tiene un registro de dirección.

- ✿ La CPU indica al DMA la ubicación de la fuente/destino de la transferencia.
- ✿ DMA opera el bus e interrumpe a la CPU cuando se completa la transferencia.

“DMA y CPU compiten por el bus de memoria.”

Servicios de E/S

1. Denominación de archivos y dispositivos.
2. Control de acceso.
3. Operaciones adecuadas para archivos y dispositivos.
4. Asignación de dispositivos.
5. Búfering, caché, y spooling, para suministrar una comunicación eficiente con el dispositivo.
6. Planificación de E/S.
7. Gestión de errores y recuperación de fallos asociados con el dispositivo.
8. Aislar en un módulo las características y conducta específica del dispositivo.

Mecanismos y funciones de los manejadores de dispositivos (drivers)

Podemos estructurar el software de E/S en capas:

1. Manejadores de dispositivos
2. Software de E/S independiente del dispositivo
3. Software a nivel de usuario.

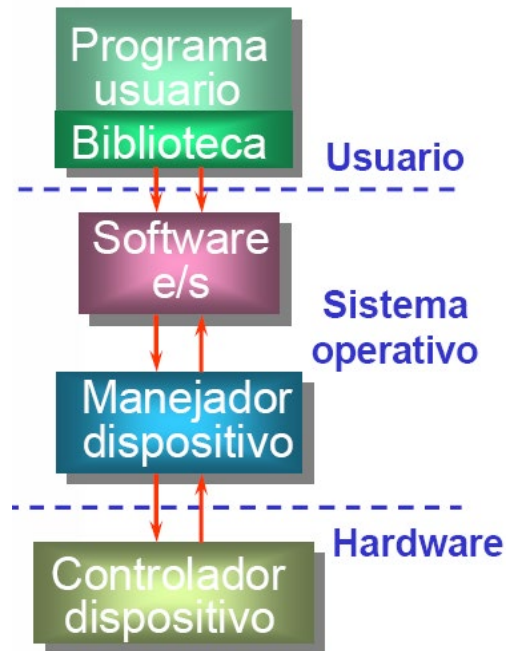


Figura 4. Arquitectura software del sistema de E/S

1. Manejador de dispositivos

Contienen todo el código dependiente del dispositivo. Cada manejador gestiona un tipo o clase de dispositivo. Acepta peticiones “abstractas” de la capa de software independiente del dispositivo, y controla que la petición se realiza:

1. Traduce petición abstracta en órdenes para el controlador del dispositivo.
2. Se bloquea o no, según tipo de operación
3. Si no hay errores, da respuesta si es necesario, y retorna al llamador.

2. Software de E/S independiente del dispositivo




Realiza las tareas comunes a todos los dispositivos y suministra una interfaz común al usuario. Las principales funciones son:

- ✎ Suministra interfaz uniforme a los manejadores
- ✎ Realiza la designación de dispositivos
- ✎ Implementa la protección de dispositivos
- ✎ Establece el tamaño de bloque independiente del dispositivo (para dispositivos de bloques)
- ✎ Implementa el “buffering”
- ✎ Realiza la asignación de almacenamiento en dispositivos de bloques (básicamente discos)
- ✎ Asignar y liberar dispositivos dedicados
- ✎ Informar de errores producidos en la realización de la operación.

3. Designación de dispositivos

Espacio de nombre de dispositivos –define cómo identificar y nombrar los dispositivos

Existen diferentes espacios de nombre:

1. Espacio de nombres hardware –especifica el dispositivo por el controlador al que esta ligado y el número de dispositivo lógico dentro del controlador.
2. Espacio de nombres kernel -utilizado por el núcleo, suele basarse en el anterior.
3. Espacio de nombres de usuario -debe ser un esquema sencillo y familiar.
4. El sistema de e/s independiente del dispositivo define las semánticas de los espacios de nombres kernel y usuario, y establece las correspondencias entre ellas.
5. En Unix, el espacio de nombres kernel identifica un dispositivo por:
 -  **número principal** - identifica el controlador
 -  **número secundario** - instancia del dispositivo.
 -  **tipo de dispositivo** - carácter o bloque.

- c. Las operaciones de E/S se suministran por medio de llamadas al sistema y gestión de interrupciones, que son lentas.
- d. Reducir el número de veces que los datos son copiados manteniéndolos en caché.
- e. Reducir la frecuencia de interrupciones utilizando, si es posible, grandes transferencias de datos.
- f. Descargar computación de la CPU principal utilizando controladores DMA.
- g. Aumentar el número de dispositivos para reducir la contención de uno único, y así, mejorar el uso de CPU.
- h. Incrementar memoria física para reducir la cantidad de tiempo en paginación y por ello mejorar el uso de CPU.

Referencias:

- Stallings, W. **Sistemas Operativos**. 5a. Edición. Editorial Prentice-Hall. (TEXTO). Marzo 2014.

Infografía:

1. Apuntes de Laboratorio de Arquitectura de Ordenadores. Disponible en: <http://www.it.uc3m.es/amarin/curs/lao/lao.html>
2. Puertos de entrada – salida: <https://hardzone.es/tutoriales/componentes/puertos-entrada-salida/>