

Administración de procesos de entrada salida mediante interrupciones. Criterios. Prioridades. Estructura de sistemas basados en interrupciones

José Luis CAERO

Introducción

- El sistema de entrada salida es la parte del S.O. encargada de la gestión de los dispositivos de E/S.
- Es el encargado de atender los requisitos propios de los dispositivos de E/S y ocuparse de que funcionen de forma eficiente y confiable, al mismo tiempo de ocultar los detalles específicos de cada uno y presentando una abstracción uniforme de las operaciones genéricas de E/S a las capas más externas del sistema operativo y a los programas del usuario.
- Su planificación deberá ser cuidadosa cuando manipule tareas concurrente de E/S.

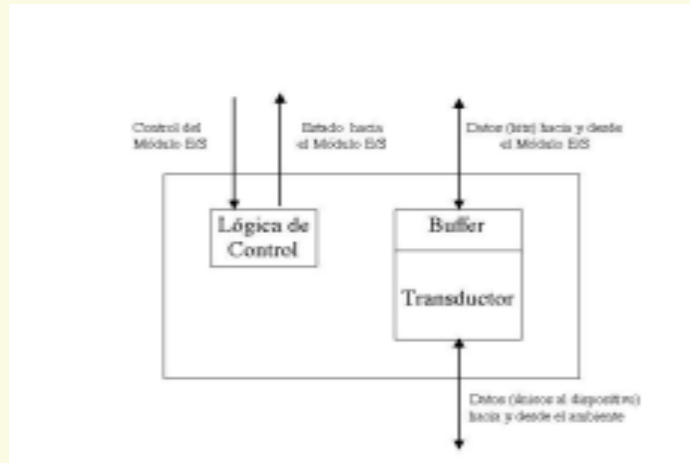
Tipos de dispositivos externos de E/S

- **De interacción con humanos:** comunicación entre el usuario y la computadora.
- **De interacción con maquinas:** posibilitan comunicación con elementos del equipo. (drive de soportes de almacenamiento masivos).
- **De comunicación:** permiten la comunicación con dispositivos remotos.

Características de los dispositivos de E/S

- Existe una gran variedad de dispositivos E/S con gran cantidad de métodos de operación.
- La velocidad de transferencia con estos dispositivos es mucho menos que la que se tiene con la memoria o el procesador.
- Los sistemas expertos típicamente usan formatos de datos y longitudes de palabra diferentes.

Estructura Interna de un Dispositivo Externo

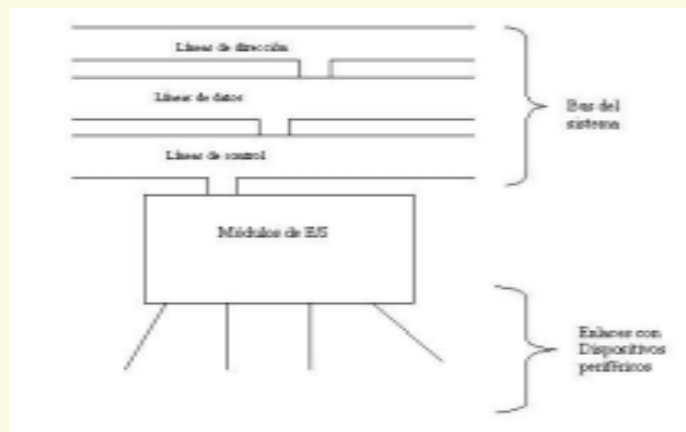


Módulo de E/S

Este módulo es el encargado del control de uno o más dispositivos externos, del intercambio de datos entre estos dispositivos y la memoria principal y/o los registros de la CPU.

Constituyéndose de esta forma en la interfaz entre la CPU y la memoria a través del bus del sistema, y la interfaz entre uno o más dispositivos periféricos mediante enlaces de datos específicos.

Modelo Genérico de un Módulo de E/S



Funciones del módulo de E/S

▪ **Control y temporización:** la CPU puede comunicarse con uno o más dispositivos externos en cualquier orden según las necesidades de E/S del programa.

▪ **Comunicación con el procesador:**

Decodificación de ordenes, el módulo de E/S acepta ordenes de la CPU que generalmente se envían utilizando líneas del bus de control.

Intercambio de datos por medio del bus de datos.

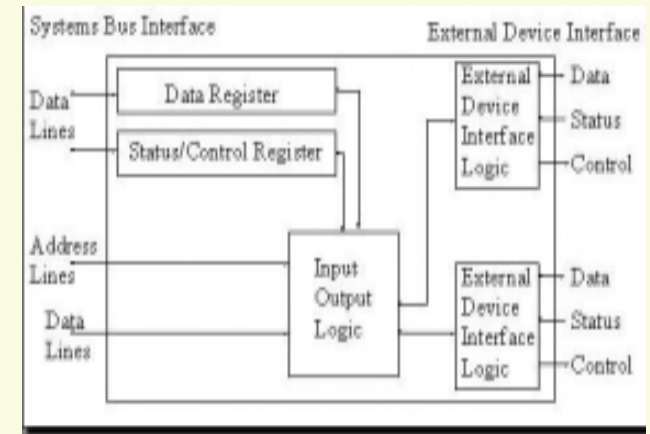
Información de estado del módulo de E/S, necesario por ser los dispositivos lentos respecto de la CPU.

Reconocimiento de dirección, permite que el módulo reconozca la dirección para cada uno de los periféricos.

Funciones del módulo de E/S

- **Comunicación con los dispositivos externos** para intercambiar ordenes, información del estado y datos.
- **Almacenamiento de datos en buffers**, permite adaptar la velocidad de transferencia desde y hacia la memoria principal o la CPU respecto de los dispositivos periféricos.
- **Detección de errores**, es el responsable de detectar e informar los errores a la CPU. (p.e. por efectos mecánicos y eléctricos en el funcionamiento del dispositivo).

Diagrama de Bloques de un Módulo de E/S



Componentes del módulo de E/S

- Registros de datos** (buffer de E/S), que le permite adaptar velocidades de transmisión
- **Registro de ordenes:**
 - De designación de modo: p.e. entrada, o salida, algoritmo de chequeo de errores, etc.
 - Operacionales de E/S: gobiernan la temporización y el mecanismo de las transferencias reales de datos, dando acciones como activar/desactivar, leer y escribir.
 - **Registro de estado**, brinda información hacia la CPU: preparado o no, buffer vacío o lleno, e indica errores detectados.

Módulo de Entrada - Salida

- El módulo se conecta al resto de máquina por medio de un conjunto de líneas.
- Un módulo de E/S que se encarga de la mayoría de los detalles del procesamiento presentando a la CPU una interfaz de alto nivel, se le denomina canal de E/S o procesador de E/S; mientras que un módulo que sea bastante simple y requiera un control detallado normalmente se denomina controlador de E/S o controlador de dispositivo.

Técnicas de comunicación de E/S

- **E/S programada:** El procesador emite una orden de E/S de parte de un proceso a un módulo de E/S; el proceso espera a que termine la operación antes de seguir.
- **E/S dirigida por interrupciones:** El procesador emite una orden de E/S de parte de un proceso, continua la ejecución de otras instrucciones y es interrumpido por el módulo de E/S cuando este ha completado su trabajo. Durante este tiempo se consulta por el estado del módulo de E/S, degradándose el rendimiento del sistema.
- **Acceso directo a memoria (DMA):** se produce intercambio de datos entre la memoria principal y un módulo de E/S sin que participe el procesador principal. El módulo debe ser inteligente para reemplazar al procesador, permitiéndolo seguir trabajando mejorando el rendimiento del sistema, al no tener tiempos ociosos del procesador.

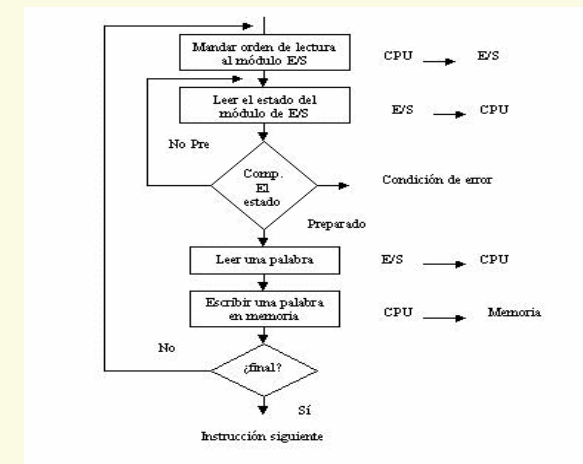
E/S programada

- Desde el punto de vista de las órdenes:
 - **Control:** se utiliza para activar el periférico e indicarle que hacer.
 - **Test:** se utiliza para comprobar las condiciones de estado asociadas con el módulo de E/S y sus periféricos.
 - **Lectura:** hace que el módulo de E/S capte un dato de un periférico y lo sitúe en un buffer interno
 - **Escritura:** hace que el módulo de E/S tome un dato (byte o palabra) del bus de datos y posteriormente lo trasmita al periférico

E/S programada

- Desde el punto de vista de las instrucciones:
 - **Las E/S asignadas en memoria,** existe un único espacio de direcciones para las posiciones de memoria y los dispositivos de E/S. En este tipo de direccionamiento se necesita una sola línea de lectura y una sola línea de escritura en el bus
 - **E/S aislada,** el bus puede disponer de líneas de lectura y escritura en memoria junto con líneas para ordenes de entrada y salida. En este caso, las líneas de órdenes especifican si la dirección se refiere a una posición de memoria o a un dispositivo de E/S. Puesto que el espacio de direcciones de E/S esta aislado del de memoria.

E/S PROGRAMADA



E/S dirigidas por interrupciones

- El mecanismo de interrupción hace posible que el módulo provoque la detención temporal de la actividad que realiza el procesador para prestarle servicio a los sucesos de E/S cuando ocurran. Este mecanismo es asistido por el hardware para sincronizar el procesador con los sucesos de E/S (asíncronos). Luego de dar servicio a un dispositivo ejecutando su rutina de servicio de interrupción asociada, el procesador reanuda la actividad previa desde el punto de la interrupción.
- Esta forma de administrar la E/S resulta ventajosa respecto de la programada, porque evita el ciclo de espera activa.

Clases de interrupciones

- De programa: las genera alguna condición que ocurre como resultado de una ejecución, sobreflujo, división sobre cero o llamada ilegal.
- De temporizador: por un timer dentro del procesador a intervalos regulares.
- De E/S generadas por un controlador de E/S para indicar la terminación normal de una operación o diferentes condiciones de error.
- De falla de hardware: debidas a fallas, como en la paridad de memoria.

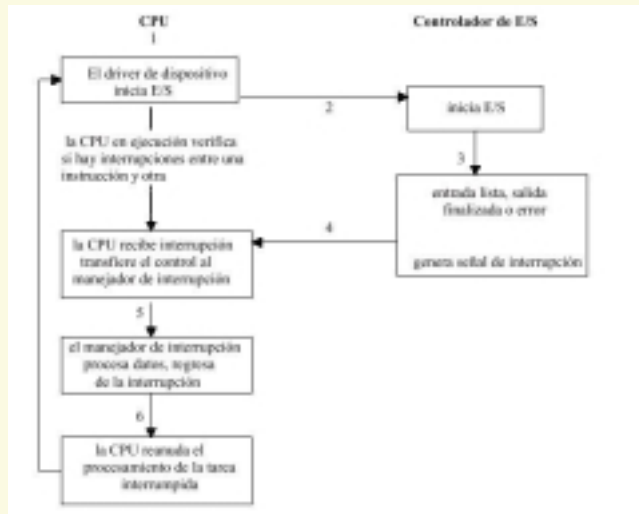
Eventos luego de una operación de E/S

- El dispositivo envía una señal de interrupción al procesador.
- El procesador termina la ejecución de la instrucción en curso antes de responder.
- El procesador comprueba si hay interrupciones, determina que hay una, y envía una señal de reconocimiento al dispositivo que origino la interrupción.
- El procesador para transferir el control a la rutina de interrupción, guarda la información necesaria para continuar con el programa en curso en el punto en que se interrumpió, en la pila del sistema.
- Después el procesador carga el contador de programa con la posición de inicio del programa de gestión de la interrupción solicitada.

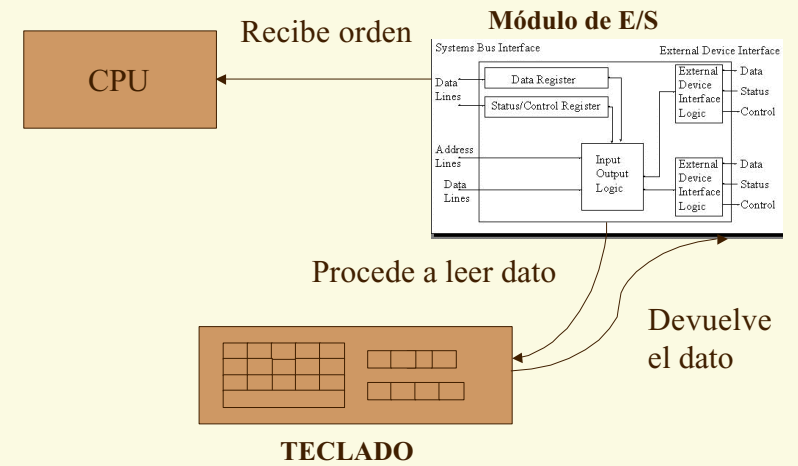
Eventos luego de una operación de E/S

- Se guardarán los contenidos de los registros del procesador puesto que estos registros pueden ser utilizados por la rutina de interrupción. Usualmente, la rutina de gestión de interrupción empezará almacenando en la pila los contenidos de todos los registros.
- La rutina de gestión de la interrupción puede continuar ahora procesando la interrupción. Esto incluirá el examen de la información de estado relativa a la operación de E/S o a cualquier otro evento que causará la interrupción.
- Cuando el procesamiento de la interrupción ha terminado, los valores de los registros almacenados se recuperan de la pila y se vuelven a almacenar en los registros.
- El paso final es recuperar los valores de PSW y del contador del programa desde la pila. Como resultado, la siguiente instrucción que se ejecute pertenecerá al programa previamente interrumpido

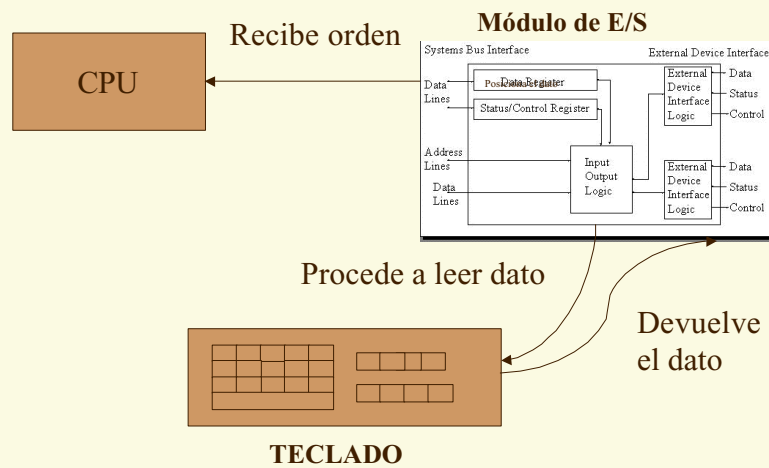
E/S dirigidas por interrupciones



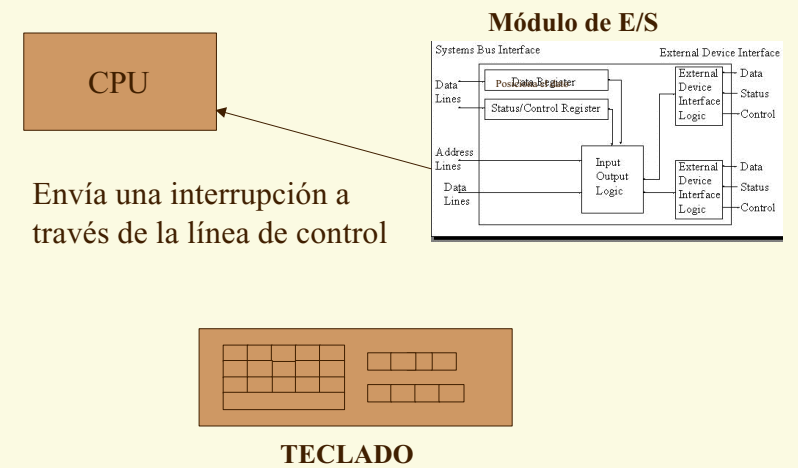
E/S dirigidas por interrupciones



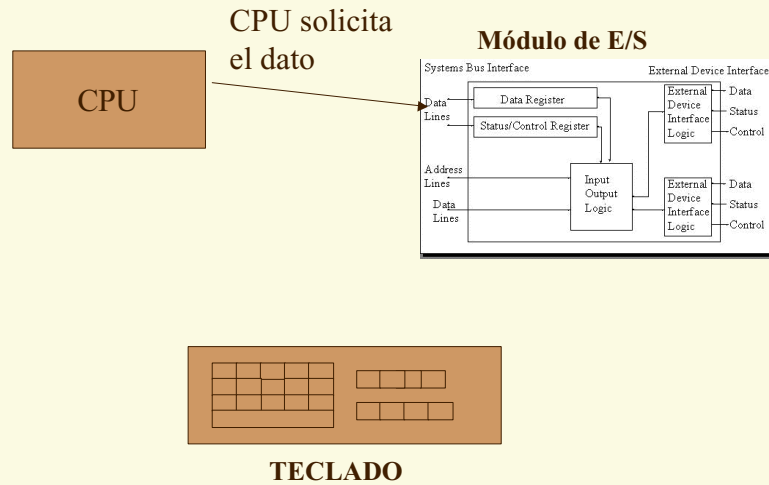
E/S dirigidas por interrupciones



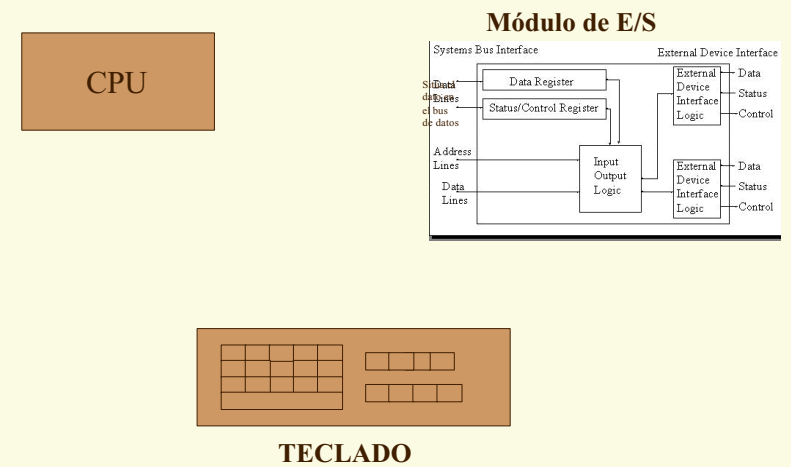
E/S dirigidas por interrupciones



E/S dirigidas por interrupciones



E/S dirigidas por interrupciones



Técnicas para identificar el módulo de E/S que solicita interrupción, y asignar prioridades.

- **Múltiples líneas de interrupción**, al utilizar varias líneas, es probable que a cada una se conecten varios módulos de E/S, se debe utilizar alguna de las restantes técnicas.
- **Consulta por software**, cuando la CPU detecta una interrupción, se produce una bifurcación a una rutina de servicios de interrupción que se encarga de consultar a cada módulo de E/S para determinar el módulo que ha provocado la interrupción, con la correspondiente consumo de tiempo.
- **Conexión en cadena**, por hardware vectorizado. De los módulos de E/S que proporciona, de hecho, una consulta hardware.
- **Arbitraje de bus** vectorizado, sólo un módulo de E/S debe disponer del control del bus antes de poder activar la línea de petición de interrupción. Al detectarla la CPU responde mediante la línea de reconocimiento de interrupción. Después, el módulo que solicitó la interrupción sitúa su vector en las líneas de datos.

E/S mediante Acceso Directo a Memoria

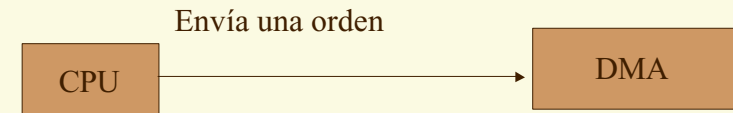
La unidad de DMA es capaz de imitar a la CPU, tomando el control del sistema E/S para transferir los datos con la memoria por el bus del Sistema, haciéndose cargo la unidad y no la CPU.

El modulo DMA usa el bus sólo cuando la CPU no lo necesita, o bien puede forzar a la CPU a que suspenda temporalmente su operación, conocido como **Robo de Ciclo**

E/S mediante Acceso Directo a Memoria



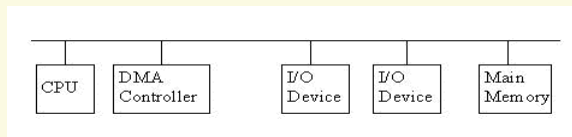
E/S mediante Acceso Directo a Memoria



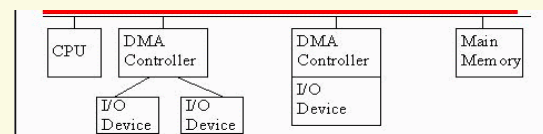
- Si se trata de una lectura o una escritura.
- La dirección del dispositivo de E/S.
- La posición inicial de memoria.
- El número de palabras a leer o escribir.

E/S mediante Acceso Directo a Memoria

- Todos los módulos comparten el mismo bus del sistema

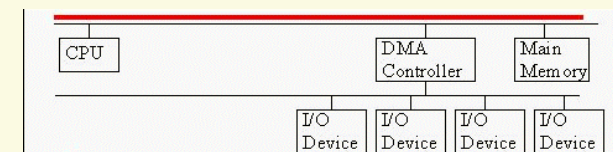


- Se integran las funciones de DMA y de E/S



E/S mediante Acceso Directo a Memoria

- Todos los módulos de E/S conectados a un módulo de DMA mediante un bus de E/S



Principios del software de E/S

- La idea básica es organizar el software en capas, donde la de mas abajo se ocupe de ocultar las características del hardware y las de más arriba presenten una interfase amigable con el usuario.
- Su función es establecer la vinculación entre los distintos módulos para realizar la transferencia de los datos en forma controlada
- El Sistema Operativo (SO), controla las Operaciones de E/S ya que:
 - Las interfaces del hardware requieren de un software complejo para controlarlo y usarlo.
 - Los Dispositivos Periféricos son recursos compartidos.
 - El SO provee una interfaz consistente, uniforme, y flexible para todos los dispositivos.

Estructura de capas

Los objetivos se logran de una forma comprensible y eficiente al estructurar el software en cuatro capas:

- Manejadores de interrupciones
- Manejadores de dispositivos
- Software de sistema operativo independiente de los dispositivos
- Software a nivel usuario

Manejadores de interrupción

- Las interrupciones deben ocultarse. La mejor manera de ocultarlas es tener todos los procesos que comienzan una E/S bloqueados hasta que termine y se haya producido una interrupción.
- Cuando ocurre una interrupción, el procedimiento de la interrupción hace lo que tenga que hacer para desbloquear el proceso que lo genero. Solamente debe generar un PAI (pedido de atención de interrupción) a la CPU, cuando se a completado la operación de E/S.

Manejadores de dispositivos

- Es un software formado por un conjunto de rutinas y tablas que instalados forman parte del SO
- Es un conjunto de programas que provee la interfase entre el SO y un determinado dispositivo periférico.
- Sirven para ejecutar y controlar todas las operaciones de E/S que se realizan sobre un periférico determinado.

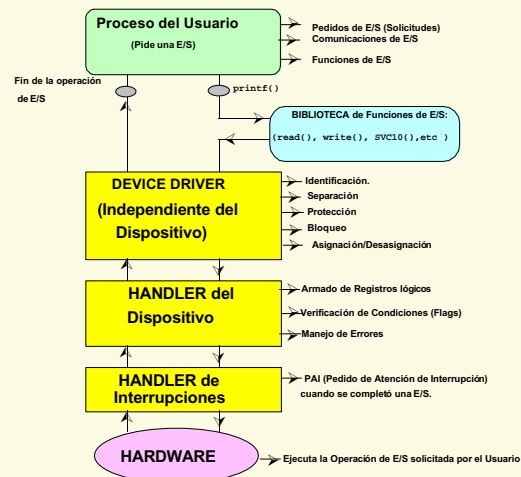
Software de E/S independiente del Dispositivo

- La diferencia que hay entre drivers y software de E/S es el sistema al que están subordinados.
- La función básica es hacer funciones de E/S que son comunes a todos los dispositivos y que proveen una interfase uniforme al software del nivel de usuario.
- Este se encarga de mapear los nombres de dispositivos simbólicas y transformarlos en los nombres correctos del driver, también se utiliza el Buffering.

Software de E/S del Espacio del Usuario

- Consiste en Bibliotecas Vinculadas
- Basado en dos procesos diferentes:
 - Procedimientos que lo único que hacen es darle los parámetros y llamar a una rutina del S.O.
 - Sistema de SPOOLING (Sist. Multiprogramado) trata de simular un periférico en línea mientras que no hay disponibilidad en el mismo. Crea un proceso especial DEAMON (Servidor de impresión). Para imprimir el proceso genera primero un archivo con los datos que desea imprimir y lo pone en el directorio de spooling. El Deamon que es el único que puede imprimir, imprime los archivos que se encuentren en el directorio de spooling.

Capas del Software de E/S



La transferencia de datos puede resultar:

- Exitosa: La información transferida llega exitosa a su destino.
- Falla parcial: la falla ocurre en la mitad de la transferencia y el bloque de destino tiene información errónea.
- Falla total: La falla ocurre lo suficientemente temprano durante la transferencia de modo que el bloque destino queda intacto.
- Interrupción de E/S: Se produce una interrupción voluntaria o no para que se complete o no la operación en curso.

Proceso de inicialización de E/S

- Es parte general de la E/S es parte del proceso general de inicialización (Booteo o Cold Start) de la máquina que realiza el SO.
- Prepara una serie de tablas de Device Drivers.
- Genera 4 pasos a seguir:
 - Reset
 - Selección del modo de Operación
 - Inicialización
 - Estado listo para operar, cargandose el Registro de Estado.

Conceptos claves en el diseño de Software

- Independencia de los dispositivos: Los programas deberían poder utilizarse sin la dependencia de un periférico para su ejecución.
- Uniformidad de los nombres: deberían de ser una cadena o un entero, y no depender del dispositivo.
- Manipulación de errores: Deberían tratarse tan cerca del hardware como sea posible. Si un controlador encuentra un error debería corregirlo.
- Transferencia sincrónica o asincrónica
- Dispositivos compartibles o dedicados

Funciones principales

- Definir las característica Lógicas-Físicas del dispositivo que controla.
- Inicializar los Registros y el modo de funcionamiento en el arranque (Booteo).
- Habilitar y deshabilitar el dispositivo para un dado proceso.
- Controlar los accesos según los permisos del usuario.
- Bloquear y desbloquear los datos durante la transferencia.

Tipos de Device Drivers

Existen dos tipos de device driver:

Block Device Driver: Los orientados a Bloques controlan los periféricos con accesos basados en una dirección de datos compuestos por varios caracteres, o sea que, manipulan varios bytes en una misma operación.

Character Device Driver: controlan aquellos periféricos que transmiten o reciben un solo caracter por vez en cada operación.

Software de Entrada

La mayoría de los sistemas ofrecen 2 modos

- Orientado a caracteres (RAW)
- Orientado a la línea (COOKED)

Software de Salida

- Se utiliza un Buffer de Salida asociado a cada terminal (Terminales RS-232-C)
- Se toman los caracteres uno a uno del espacio del usuario (Terminal de Memoria Mapeada)

FIN

Diseño de Sistemas Operativos

En el diseño de SO para la administración de E/S, hay que tener bien claro cuales son los Objetivo de Diseño y la estructura lógica de las funciones de E/S.

Diseño de Sistemas Operativos Objetivo de Diseño

Una serie de objetivos: dos de significativa importancia en la administración de E/S, son:

Generalidad: Manejo de forma uniforme, con el interés en la simplicidad y la exención de errores (objetivo difícil). Se recurre a un enfoque jerárquico y modular para el diseño de funciones de E/S. De este modo todos los niveles y procesos superiores al sistema operativo solo recurren a una serie de funciones generales (lectura, escritura, apertura, cierre, bloqueo, desbloqueo, etc).

Eficiencia: Es un objetivo importante ya que las operaciones en dispositivos de E/S son mucho mas lentas que las realizadas por el procesador o la memoria. Si bien todavía no se puede solucionar la rapidez de estos dispositivos, se puede mejorar a través del uso de la multiprogramación (donde algunos procesos esperan las operaciones de E/S y otros son ejecutados).

Diseño de Sistemas Operativos Estructura Lógica de las funciones de E/S

La estructura o modelo jerárquica:

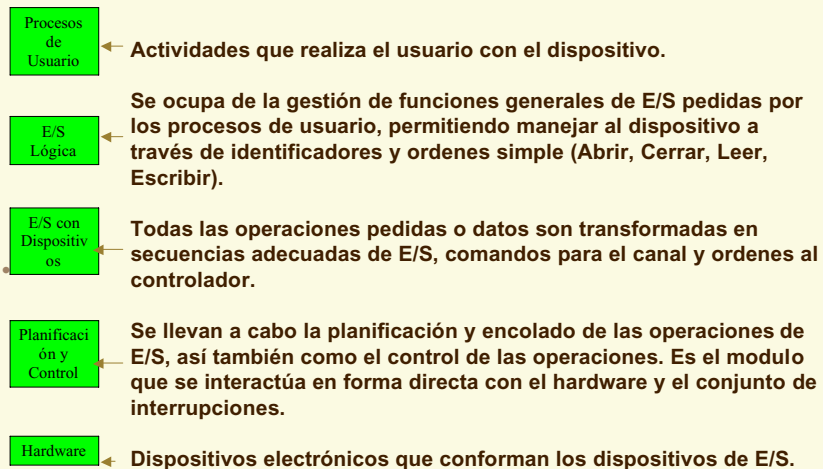
Las funciones del SO se separan de acuerdo a su complejidad, sus rangos característicos de tiempos y su nivel de abstracción.

En esta jerarquía cada nivel realiza una parte satisfaciendo el conjunto a todas las funciones necesarias del SO.

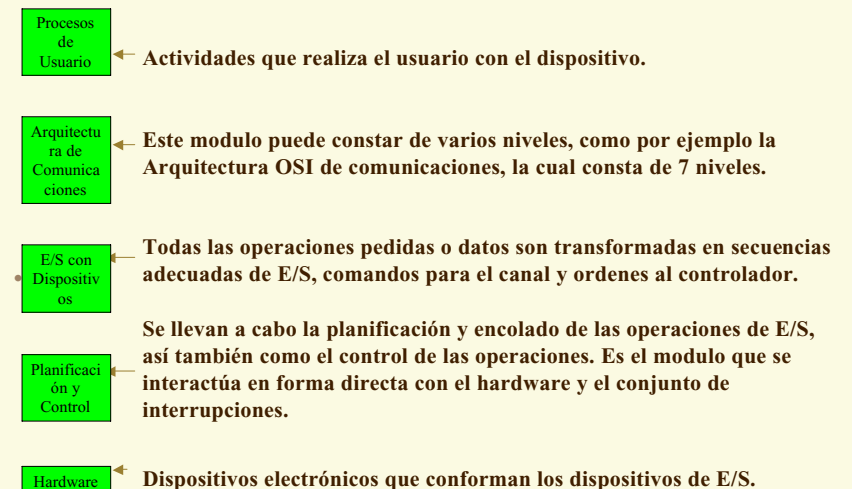
El orden van de niveles inferiores (HW) hasta niveles superiores (User) todo el problema se descompone en una serie de sub-problemas.

Tres tipos de jerarquías según el tipo de dispositivo y de su aplicación. El Dispositivo Periférico Local, el Puerto de Comunicaciones y Sistemas de Archivo.

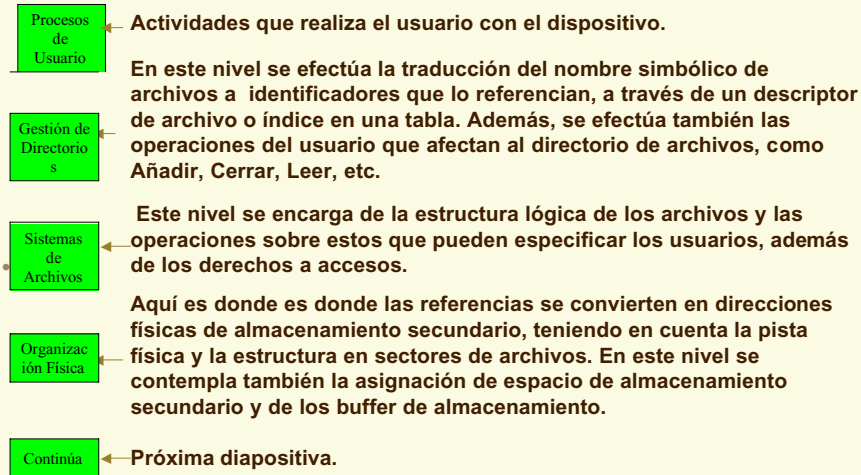
Diseño de Sistemas Operativos Dispositivo Periférico Local



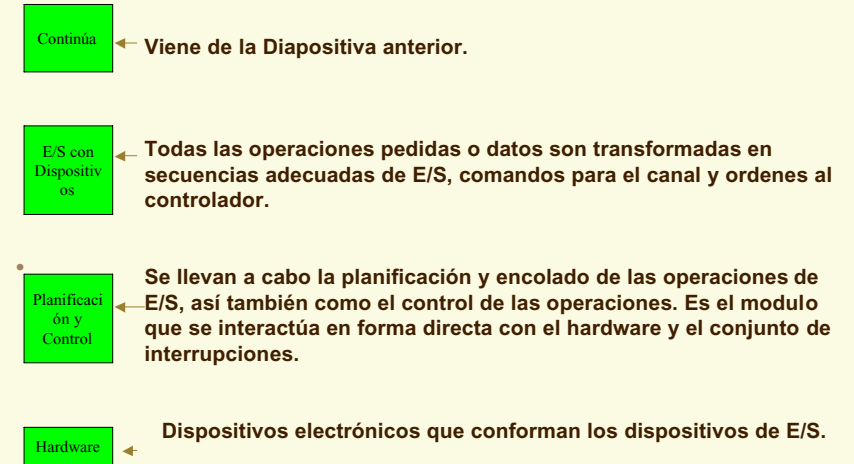
Diseño de Sistemas Operativos Puerto de Comunicaciones



Diseño de Sistemas Operativos Sistema de Archivos



Diseño de Sistemas Operativos Sistema de Archivos (Continuación)



Diseño de Sistemas Operativos Almacenamiento Intermedio (Buffering)

Es realizar la transferencia de entrada por adelantado a las peticiones y realizar las transferencias de salida un tiempo después de hacer la petición.

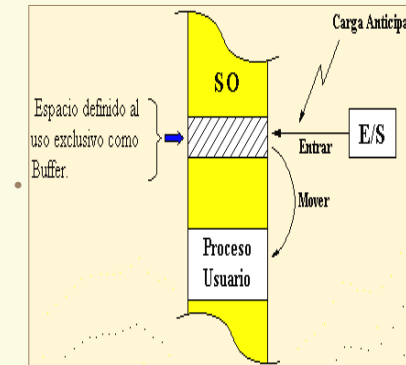
Se debe a que si se transfiere un bloque desde el área de un proceso de usuario hacia un modulo de E/S directamente, el proceso se bloqueará durante la transferencia y no podrá ser removido de la CPU.

Ejemplo: el buffer sencillo, el buffer doble y el buffer circular

Buffer Sencillo

Existe una porción de memoria que es para el uso exclusivo del dispositivo de E/S, en donde carga o lee la información.

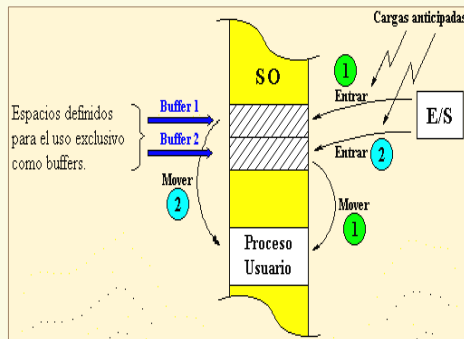
En el siguiente diagrama, se puede ver una carga anticipada en la memoria central



El dispositivo que es mas lento que el acceso a memoria, realiza una carga de información en el buffer (esta carga la realiza en algún momento cuando el procesador no utiliza los buses), después el proceso del usuario, cuando lo requiera, recurrirá a este buffer para la lectura de la información de E/S. Si se efectúa dos lecturas consecutivas del dispositivo de E/S, no da tiempo a la recarga del buffer, con lo cual, la segunda lectura se realiza directamente del dispositivo de E/S, y por ende esta es mas lenta.

Buffer Doble

Esencialmente es lo mismo para las transferencias de salida, solo que se almacenaran los datos del proceso usuario en estos buffer, y luego en algún momento el dispositivo los leerá



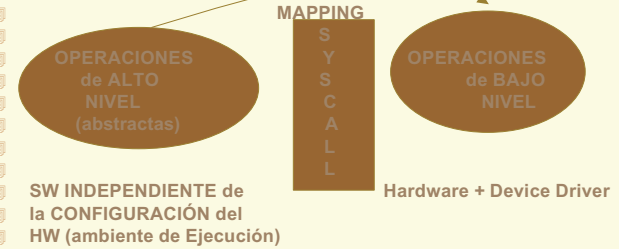
Permite la carga anticipada mientras el proceso puede estar leyendo el otro buffer, así para dos lecturas consecutivas del dispositivos de E/S, aparentará para el proceso como accesos rápidos a dicho dispositivo

Principios del Software de E/S

PRIMITIVAS de E/S del SISTEMA OPERATIVO

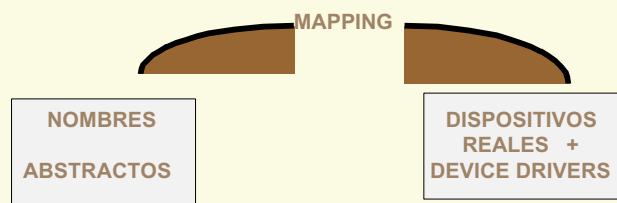
- Primitivas de E/S:
- * CHAR ⇒ getc(); putc().
- * BLOCK ⇒ read(); writwe().
- * USER ⇒ control(); random(); seek();
- * FILE ⇒ open(); close();
- * etc.

MAPPING de OPERACIONES de E/S:



Principios del Software de E/S

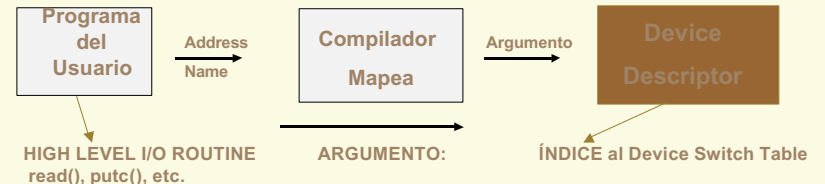
MAPPING de NOMBRES de E/S:



1. Cada S.O. tiene su forma de hacer el mapping:
2. Algunos requieren que el programador incorpore a los dispositivos cuando escribe su programa.
3. Otros requieren los servicios del SHELL para linkear el nombre de los dispositivos usados en el programa.
4. Otros hacen la vinculación dinámicamente a medida que los procesos en ejecución lo requieran.

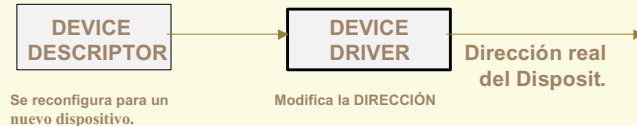
Principios del Software de E/S

1er Método:



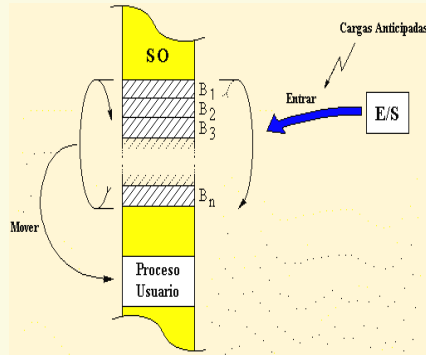
- Compromiso entre eficiencia y Binding.
- Los programas del User no contienen llamadas directas al Driver del Dispositivo ni la dirección al dispositivo

2do. Método:



Buffer Circular

Soportar una ráfaga de lectura o escritura en algún dispositivo de E/S, dando la sensación para el usuario que este dispositivo es mas rápido.



Para el caso particular de la lectura de un dispositivo de E/S, se posee una serie de búferes, desde B1 hasta Bn donde el dispositivo va cargando datos en cada uno de estos siguiendo un orden (por ejemplo desde B1 hasta Bn); y el proceso usuario que hace uso de este dispositivo, efectúa una ráfaga de lecturas desde el ultimo buffer cargado por el dispositivo de E/S, y continua en orden inverso a la carga que efectúa dicho dispositivo.