

Sistemas Operativos.

Tema 1

Arquitectura Básica de los Computadores

<http://www.ditec.um.es/so>

Departamento de Ingeniería y Tecnología de Computadores
Universidad de Murcia

Índice

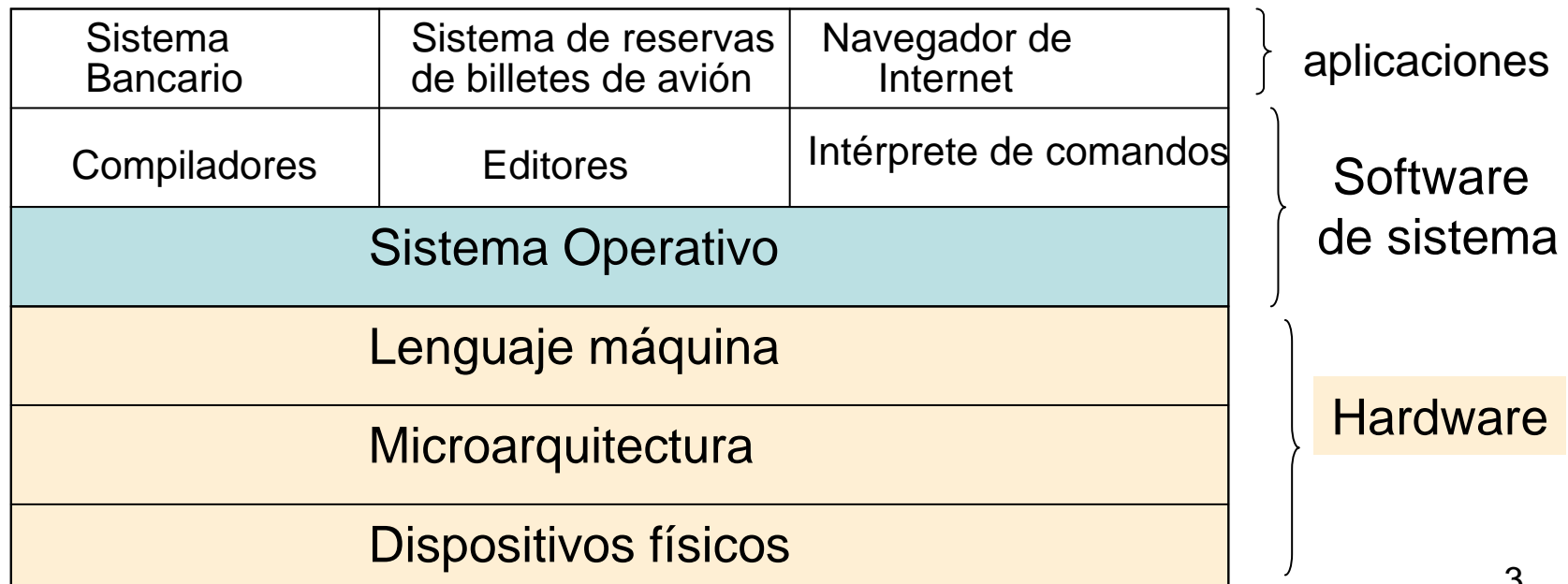
1. Estructura y Funcionamiento General
2. Procesador
3. Memoria
4. Entrada/Salida
5. Protección

Bibliografía básica: Tanenbaum[P1-3, C1.4]

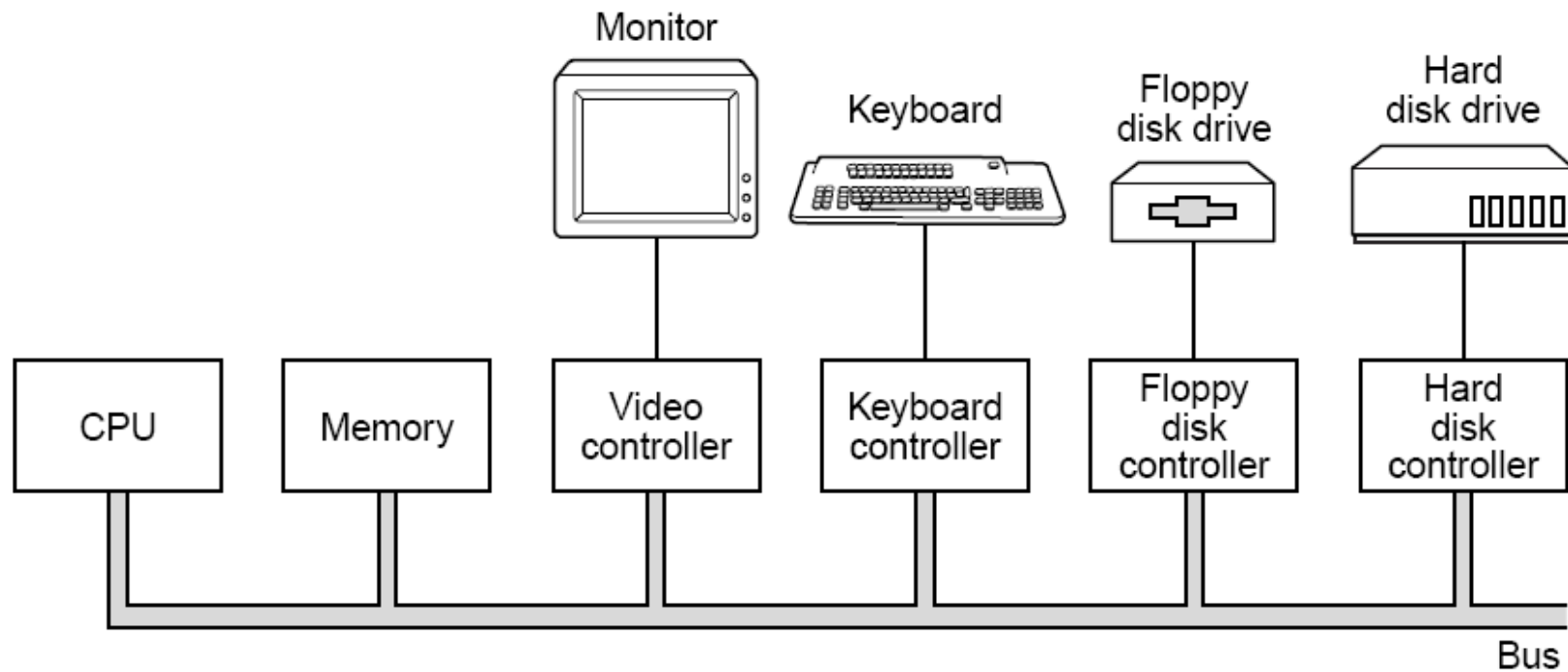
Bibliografía complementaria: Silberschatz[C2], Carretero[C1], Stallings[C1]

1 Estructura y Funcionamiento General

- Un sistema de cómputo moderno es un sistema complejo
- Para administrar todos estos dispositivos y proporcionar una interfaz sencilla de hardware \Rightarrow capa software:
sistema operativo



Estructura y Funcionamiento General (i)



Visión simplificada de un ordenador

2 Procesador

- Ciclo básico de la CPU
 1. Leer instrucción de memoria
 2. Decodificarla para determinar su tipo y operandos
 3. Ejecutarla
 4. Calcular la posición de la siguiente instrucción
 5. Chequear si hay una activación eléctrica en el pin de petición de interrupción y en caso afirmativo calcular la posición de la primera instrucción de la subrutina de atención (se almacena tb. la dirección de la inst. actual)
 6. Volver al paso 1

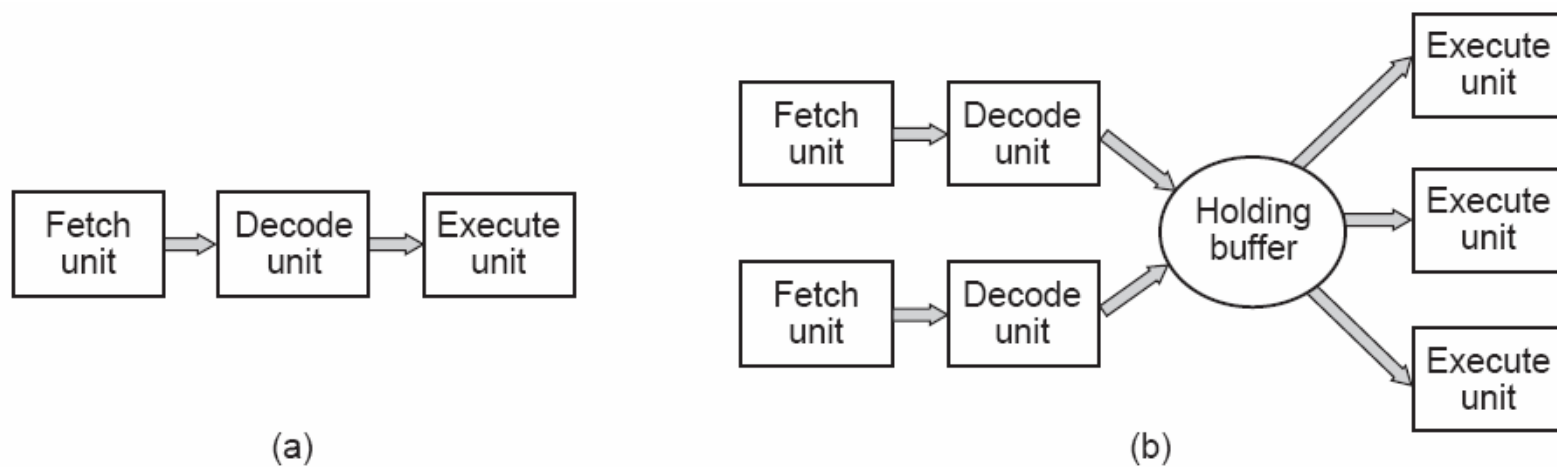
Procesador (i)

- Registros generales de datos
- Registros especiales
 - Contador de programa
 - Apuntador de pila
 - Palabra de estado del programa

El contenido de los registros determina el **contexto de ejecución** de un programa en un instante dado

Procesador (ii)

- Para mejorar el desempeño de las CPUs \Rightarrow ejecutar varias instrucciones al mismo tiempo
- Varios mecanismos:



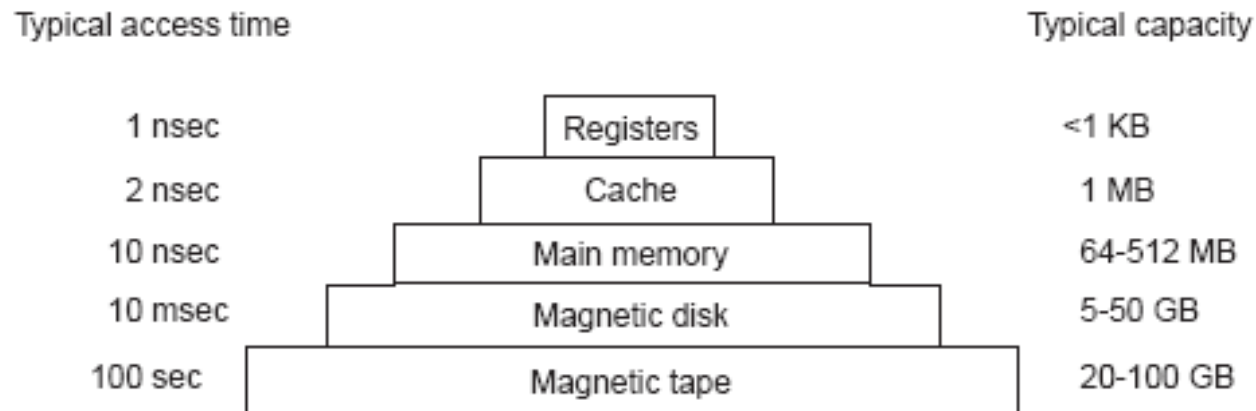
- Complican la construcción de compiladores y sistemas operativos

2 Procesador (iii)

- Dos modos de funcionamiento
 - Modo núcleo
 - Permite ejecutar todas las instrucciones y acceder a todo el hardware
 - Modo en el que se ejecuta el **sistemas operativo**
 - Modo usuario
 - Sólo se permiten ejecutar un subconjunto de instrucciones y no se permite acceder a todo el hardware (E/S, memoria..)
 - Modo en el que se ejecutan las aplicaciones
 - Mediante **llamadas al sistema** se puede acceder a los servicios del SO
 - Pasar a modo núcleo: **interrupciones software** (trap, int,..) o **hardware** (división por cero, dispositivos de E/S)

3 Memoria

- Estructura jerárquica (gracias a la *localidad de la referencia* desde la cpu se ve mucha memoria y muy rápida)



- Cada nivel es un subconjunto del nivel inferior \Rightarrow Hay información que no se encuentra en un nivel \Rightarrow **Aciertos/fallos y algoritmos de reemplazo**
- Modificación en un nivel \Rightarrow **problemas de coherencia**
 \Rightarrow propagación de modificación a niveles inferiores

Memoria (i)

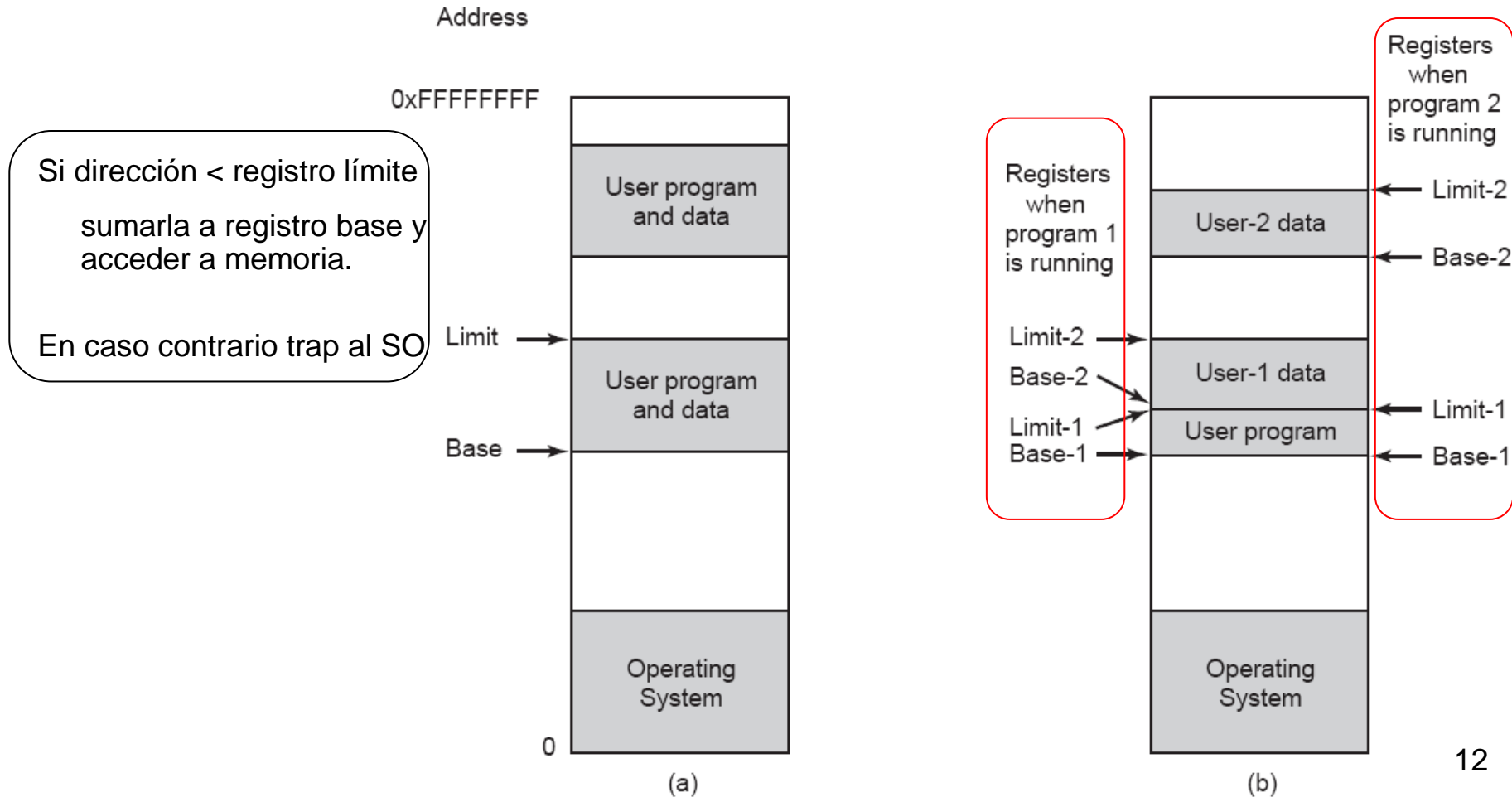
- Otros tipos de memoria:
 - ROM: memoria no volátil.
 - Código de arranque
 - Rutinas básicas de acceso a dispositivos, etc..
 - EEPROM y flash: no volátiles y lentas
 - rutinas de dispositivos y de arranque actualizables
 - discos duros flash
 - Llaveros usb y tarjetas de cámaras de fotos, etc..
 - CMOS: volátil alimentada por batería
 - Fecha y hora
 - Parámetros de configuración

Memoria. Memoria principal

- La memoria principal es el elemento más importante de la jerarquía de memoria que debe administrar el SO
- 2 problemas a resolver por el SO:
 - Proteger a unos programas de otros cuando se cargan varios en memoria.
 - Poder ubicar un programa en cualquier posición de memoria
- 2 soluciones
 - Registro base y límite
 - Utilizar memoria virtual

Memoria. Registros base y límite

- **Registro límite** → tamaño máximo del programa y los datos
- **Registro base** → posición de inicio del programa en memoria



Memoria. Memoria virtual

- **Dirección virtual** → generada por el programa
- **Dirección física** → accedida en memoria
- **MMU** → Unidad de administración de memoria (*memory management unit*):
 - Verifica las direcciones generadas por el programa
 - Convierte las direcciones virtuales en físicas
- El manejo de la MMU es función del SO (en intel-86 viene implementado por hardware)
- Cambiar de contexto → modificar configuración de la MMU

4 Dispositivos de E/S



- Controladora
 - Dispositivo electrónico que controla físicamente al dispositivo
 - Se le pasan órdenes mediante instrucciones máquina de E/S que acceden a los **registros de la controladora.**
 - Estos registros conforman una interfaz sencilla con el dispositivo
- Manejador de dispositivo (device driver)
 - Parte del SO que se comunica con la controladora: da órdenes y procesa respuestas
 - Se ejecuta en modo kernel
 - Puede
 - estar integrado en el kernel
 - cargarse en tiempo de arranque
 - cargarse en tiempo de ejecución cuando lo necesite (sin necesidad de iniciar el SO)

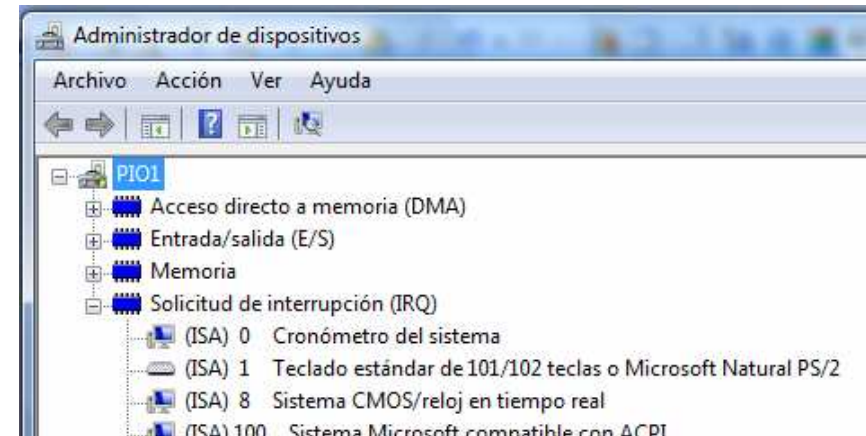
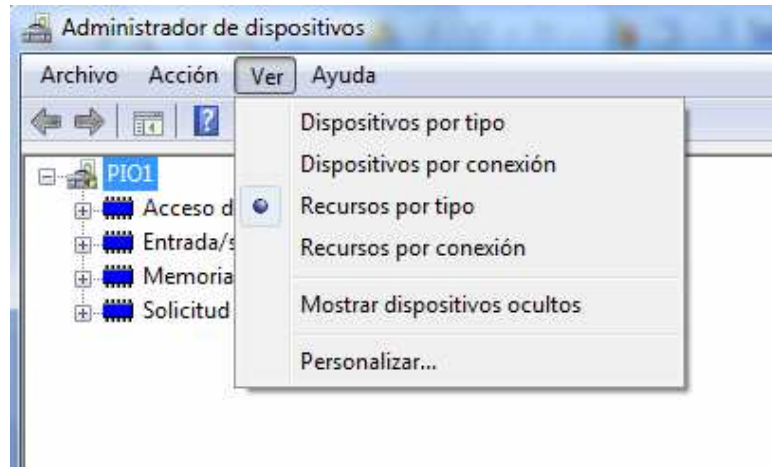
Dispositivos de E/S (sigue)

- Manejador recibe una petición del SO
 - Escribe la petición en los registros de la controladora
 - Lee de los registros el resultado de la operación
- ¿Cómo se accede a esos registros?
 - Se corresponden con el **espacio de direcciones** de la memoria principal
 - Se leen/escriben como si fueran palabras de memoria
 - No se necesitan instrucciones especiales de E/S
 - El mecanismo de protección de memoria se puede aprovechar para proteger HW
 - Se colocan en un **espacio de puertos de E/S especial**
 - Cada registro tiene una dirección de puerto
 - Los registros se leen/escriben mediante instrucciones IN y OUT especiales en modo kernel
 - No se ocupa parte del espacio de direcciones de memoria

Dispositivos de E/S (sigue)

Ejemplo:

Interfaz controladoras de dispositivos en ordenadores compatibles i86



Cada controladora que se inserta necesita para su control

- 0,1 o varios Canales DMA
- 0,1 o varias direcciones de puerto de E/S
- 0,1 o varias direcciones de memoria de E/S
- 0,1 o varios hilos de interrupción

Dispositivos de E/S. Espera activa

- Llamada al sistema → llamada al manejador del dispositivo
- Manejador pide a la controladora la operación de E/S
- Manejador de la controladora: ¿ha terminado mi petición?
 - No → continúa esperando
 - Sí:
 - Manejador de dispositivo coloca los datos donde se necesitan
 - Regresa
 - SO devuelve el control al proceso invocador

Esto se conoce como espera activa

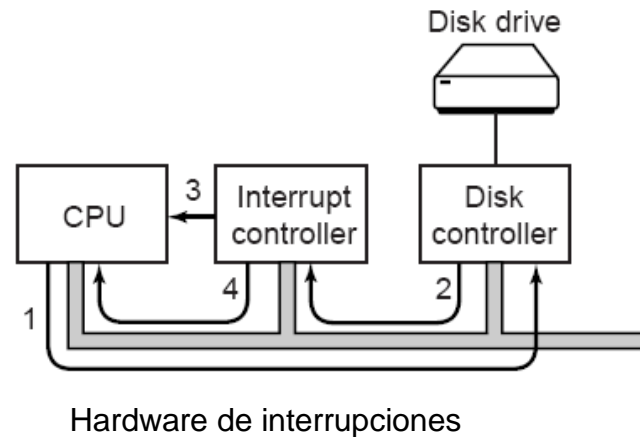
CPU ocupada dentro de un bucle escrutando a que cambie el contenido del registro de estado de la controladora del dispositivo.

Desperdicio de CPU

Dispositivos de E/S. Interrupciones

- Manejador de dispositivo
 - Pide a la controladora la operación de E/S
 - Le pide también generar una INTERRUPCIÓN al terminar
 - Regresa
- SO
 - Bloquea al proceso invocador (no le pasa la CPU)
 - Hace otras cosas
 - Al finalizar la transferencia → **interrupción** generada por la controladora
- CPU acepta la interrupción
 - Pasa a modo kernel y salta al manejador de interrupciones del dispositivo
 - N° de dispositivo → índice de una zona de memoria (tabla de vectores de interrupciones) que contiene las direcciones de los manejadores de interrupciones
- El manejador de interrupciones
 - Pregunta al dispositivo su estado
 - Desbloquea al proceso invocador (anota que se le puede pasar la cpu)
 - Al finalizar devuelve el control al programa de usuario que se estaba ejecutando (que **no tiene por qué ser el que solicitó la operación de E/S**) 18

Dispositivos de E/S. Interrupciones (i)



1. CPU ordena al controlador del disco que efectúe una operación de E/S y pasa a ejecutar otro proceso
2. Una vez finalizada la transferencia, el controlador demanda interrupción a la cpu a través del controlador de interrupciones
3. El controlador de interrupciones pide interrupción a la CPU enviándole a continuación el número de vector correspondiente (4)

La CPU llega al manejador de interrupciones por activación eléctrica de cable (**llega por hardware**) y regresa al proceso interrumpido mediante instrucción de retorno (**regresa por software**). Recordad el ciclo básico de la cpu

Dispositivos de E/S. Interrupciones (ii)

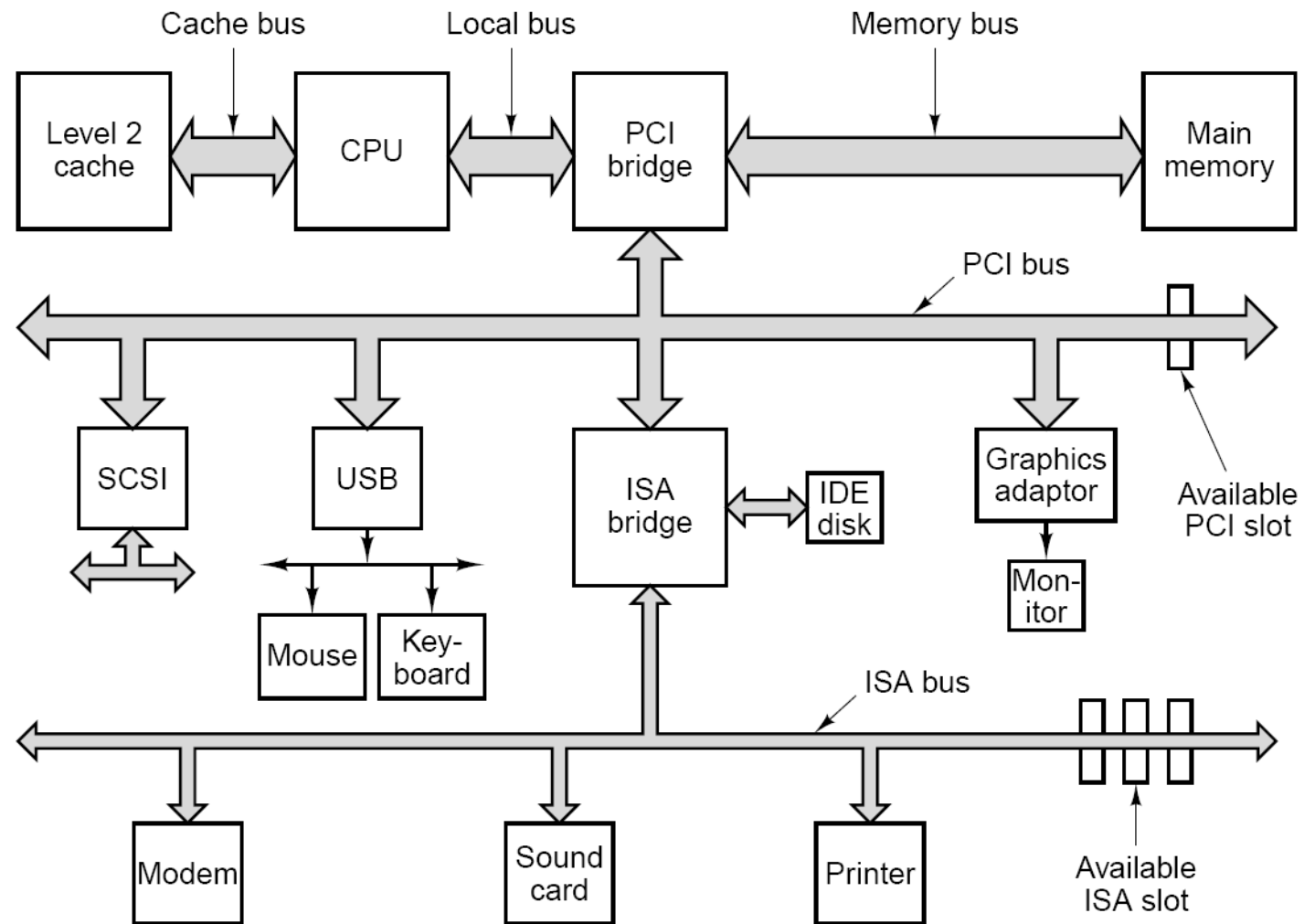
- ¿Qué pasa si llega una interrupción mientras se trata otra?
¿Paramos la interrupción actual? ¿Perdemos la nueva?
- Mecanismo en la CPU para inhabilitar interrupciones:
 - Empieza a tratar una interrupción → inhabilita interrupciones
 - Si llega una nueva → dispositivo seguirá aplicando la interrupción
 - CPU no se interrumpe mientras estén inhabilitadas
 - Termina el tratamiento → habilita de nuevo la interrupciones
 - CPU se puede volver a interrumpir
- ¿Qué pasa si varios dispositivos esperan a la CPU para su interrupción?
 - Controladora de interrupciones decide cuál atender primero
 - Prioridades estáticas asignadas a los dispositivos

Dispositivos de E/S. DMA: Acceso Directo a Memoria

- Chip que controla el flujo de bits entre la memoria principal y alguna controladora
- CPU no interviene en las transferencias memoria ↔ controladora. La CPU mientras, ejecuta otro proceso

(recordad que la CPU accede continuamente a memoria – busca instrucción y operandos, y deposita resultados-. El bus de memoria está siendo utilizado por 2 dispositivos al mismo tiempo. Unos ciclos del bus serán utilizados por el DMA y otros por la cpu)
- Al finalizar, el DMA pide interrupción a la CPU
 - La CPU salta por hardware al manejador de interrupciones y desbloquea desde allí al proceso invocador que ha demandado la transferencia de E/S (anota que se le puede pasar la cpu puesto que los datos ya están en memoria)
- Funcionamiento
 - El SO localiza un *buffer* de memoria
 - Programa la controladora DMA y la controladora específica
 - El DMA realiza la transferencia *buffer* de la controladora ↔ *buffer* de memoria
 - Controladora de DMA avisa a CPU mediante una interrupción
 - CPU trata esa interrupción
- Dispositivos muy rápidos (discos duros, tarjetas de sonido, ...)

Dispositivos de E/S. Buses



5 Protección

Varios procesos en ejecución \Rightarrow Protección

- Protección del HW \Rightarrow Operación en modo dual:
 - Modo núcleo: instrucciones de E/S, configuración MMU,...
 - Modo usuario: instrucciones de acceso al HW ilegales
- Protección de memoria \Rightarrow Registros base y límite, ...:
 - Protege a unos programas de otros y al SO
- Protección de la CPU \Rightarrow Interrupciones periódicas (cronómetros o relojes,...):
 - Interrupciones \rightarrow ejecución del sistema operativo
 - Impiden que un proceso se apropie de la CPU
- Protección total del SO y el HW \Rightarrow **uso simultáneo de los tres mecanismos**