

Estructuras

2

Sistemas Operativos y Distribuidos

Dr. Javier Echaiz

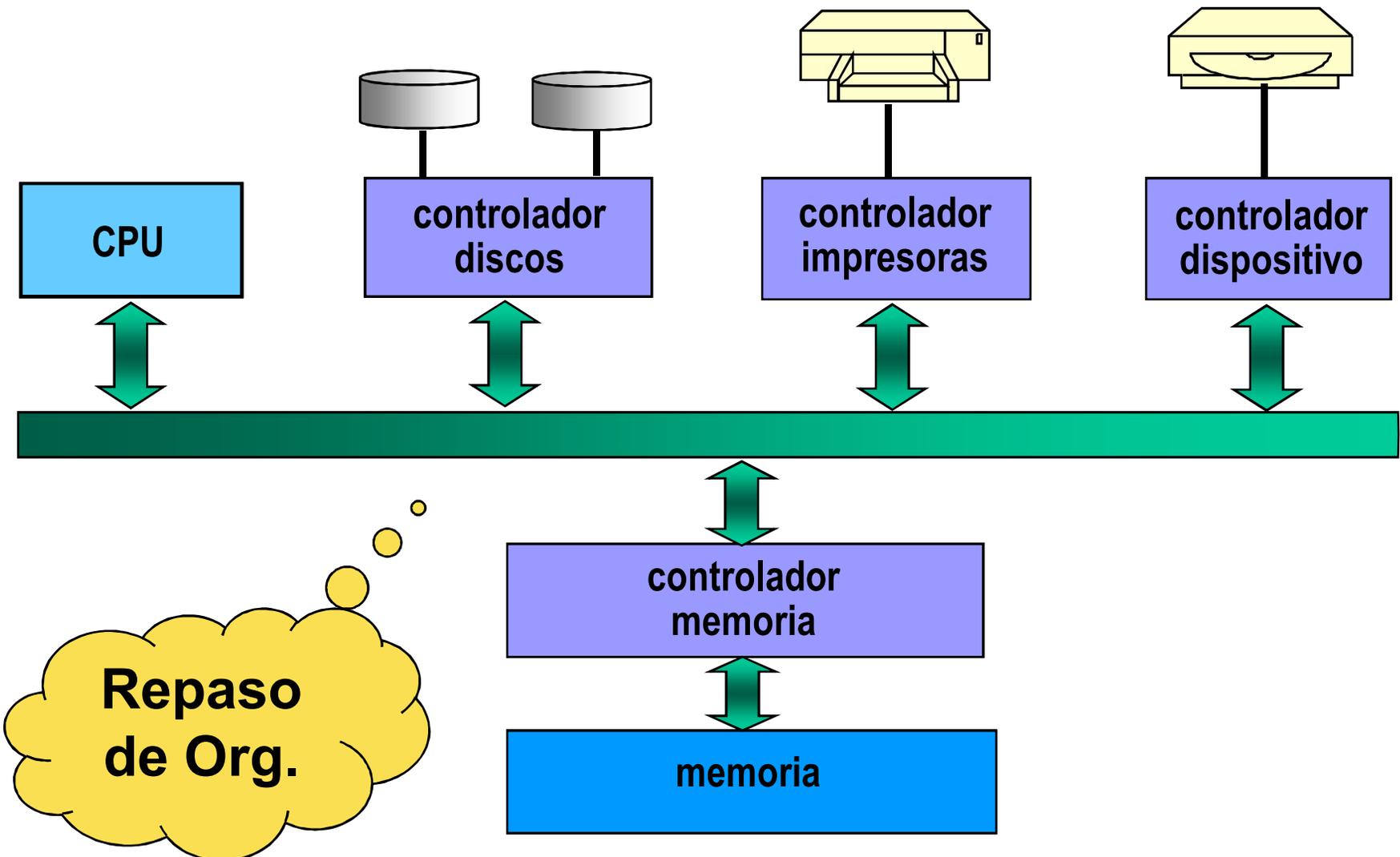
D.C.I.C. – U.N.S.

<http://cs.uns.edu.ar/~jechaiz>

je@cs.uns.edu.ar



Arquitectura del Sistemas de cómputo



Operación del Sistema de Cómputo

- Los dispositivos de E/S y el CPU pueden ejecutar concurrentemente.
- Cada controlador de dispositivos está encargado de un tipo particular de dispositivo.
- Cada controlador de dispositivos tiene un buffer local.
- El CPU mueve datos desde/hacia la memoria principal hacia/desde los buffers locales.
- La E/S es desde el dispositivo al buffer local del controlador.
- El controlador de dispositivo informa al CPU que ha finalizado su operación por medio de una *interrupción*.

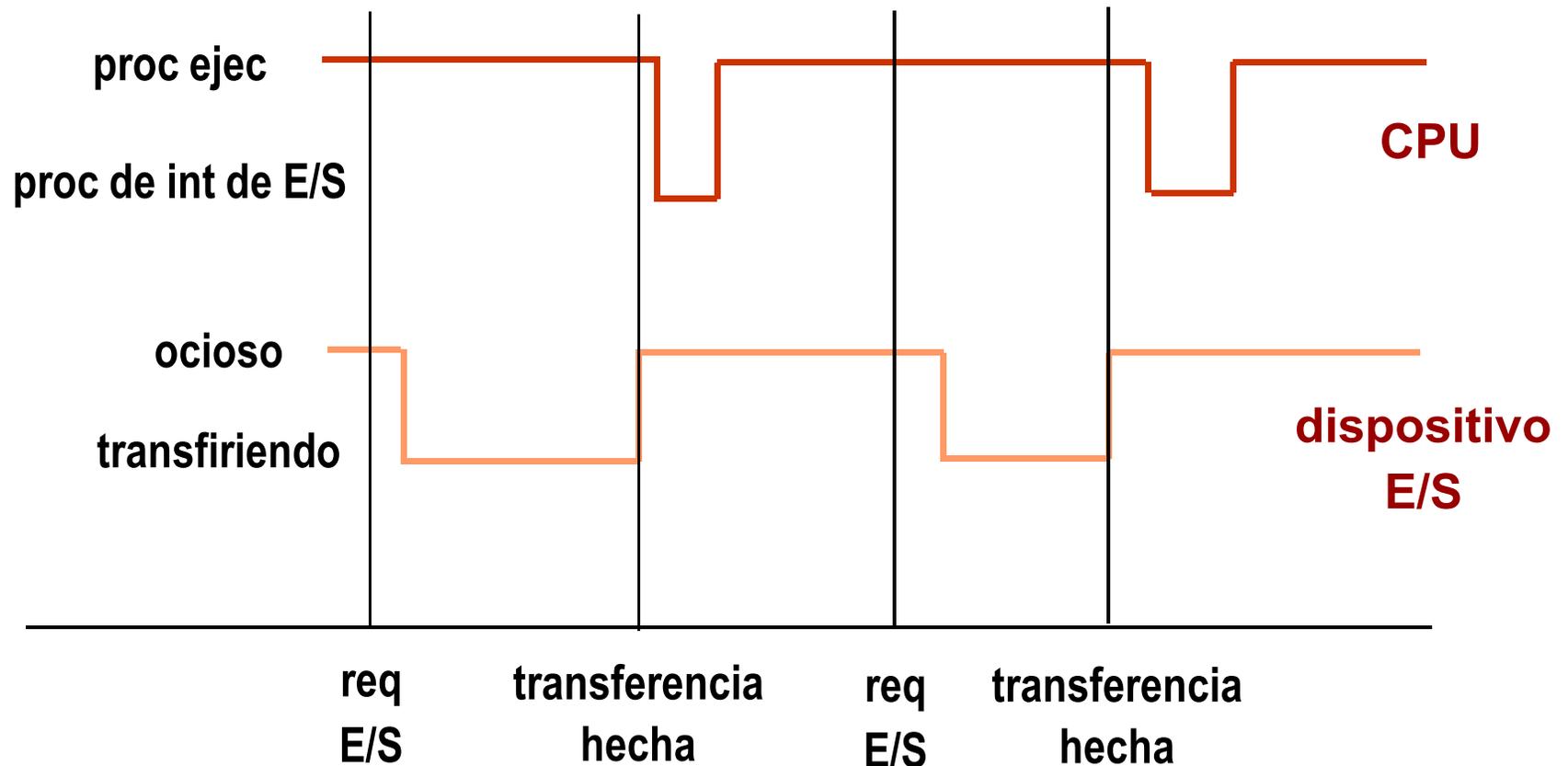
Funcionamiento de las Interrupciones

- Una **interrupción** transfiere el control a la rutina de servicio de la misma, generalmente por medio del **vector de interrupción**, que contiene las direcciones de todas las rutinas de servicio.
- La arquitectura de la interrupción debe salvar la dirección de la instrucción interrumpida.
- Las interrupciones entrantes son **deshabilitadas** mientras otra interrupción está siendo procesada para prevenir una **pérdida de interrupción**.
- Un **trap** es una interrupción generada por el software causada por un error o por un requerimiento de usuario.
- Un sistema operativo es manejado por las **interrupciones**.

Manejo de Interrupciones

- El sistema operativo preserva el estado de la CPU almacenando los registros y el contador de programa.
- Determina que tipo de interrupción ha ocurrido:
 - *polling*
 - Sistema de interrupción *vectorado*
- Segmentos de código separados determinan que tipo de acción deberían llevarse a cabo para cada tipo de interrupción.

Línea de tiempo de Interrupción para un proceso simple realizando una salida



Estructura de E/S

Forma Sincrónica

- Luego que la E/S comienza, el control retorna al programa del usuario solo cuando la E/S se haya completado.
 - una instrucción de espera mantiene ociosa la CPU hasta la próxima interrupción.
 - lazo de espera (contención para el acceso a memoria).
 - A lo sumo una E/S requerida es atendida a la vez, no hay procesamiento simultáneo de E/S.

Estructura de E/S

Forma Asíncrona

- Luego que la E/S comienza, el control retorna al programa del usuario sin esperar que la E/S se haya completado.
 - *Llamada al Sistema* – requiere al SO que permita al usuario esperar por la terminación de la E/S.
 - *Tabla de estado de los dispositivos*: contiene una entrada por cada dispositivo de E/S indicando su tipo, dirección, y estado.
 - El SO pone un índice en la tabla de dispositivos de E/S para determinar el estado del dispositivo y modificar la entrada a la tabla para incluir la interrupción.

Comparación esquemas E/S

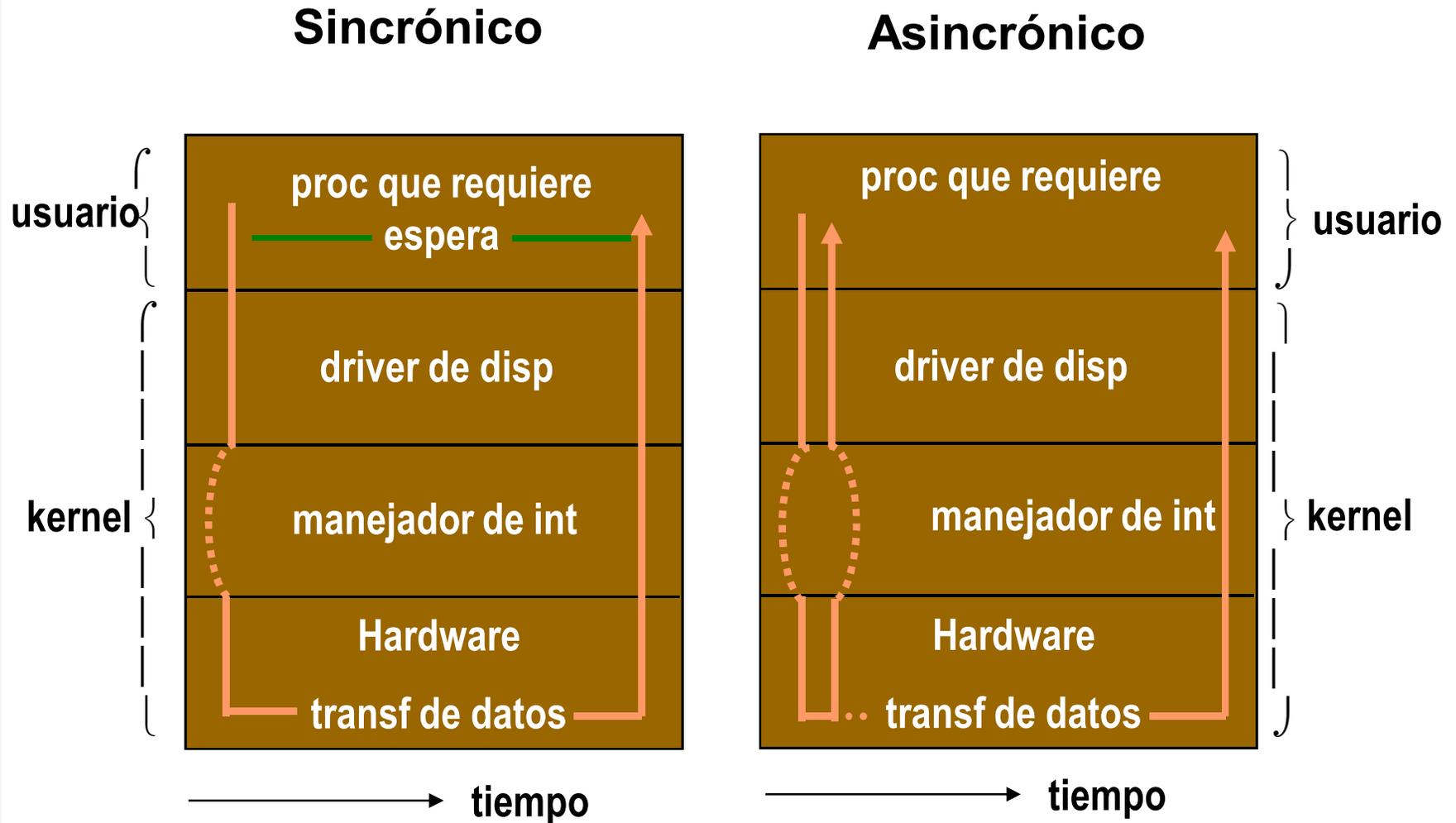
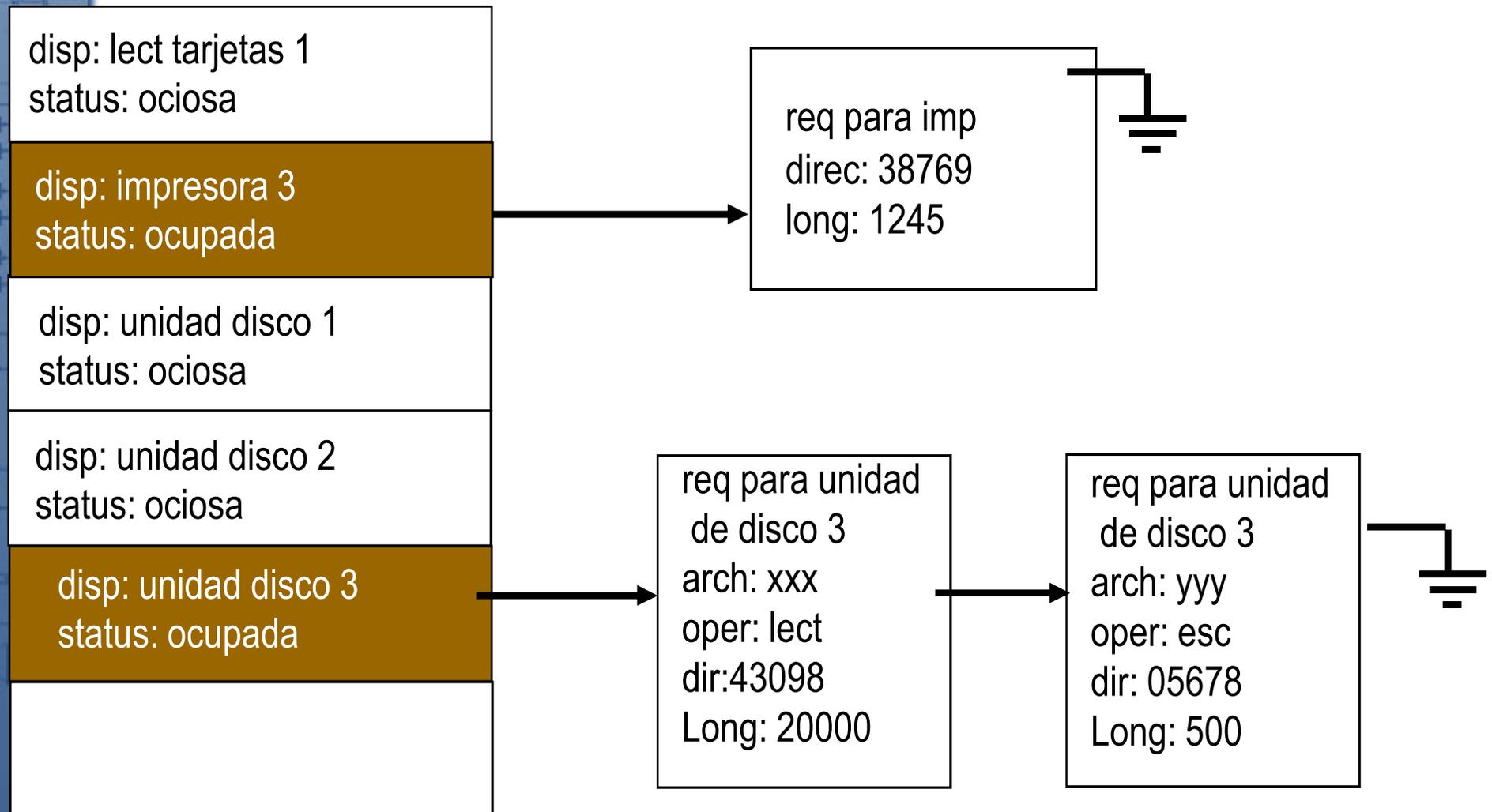
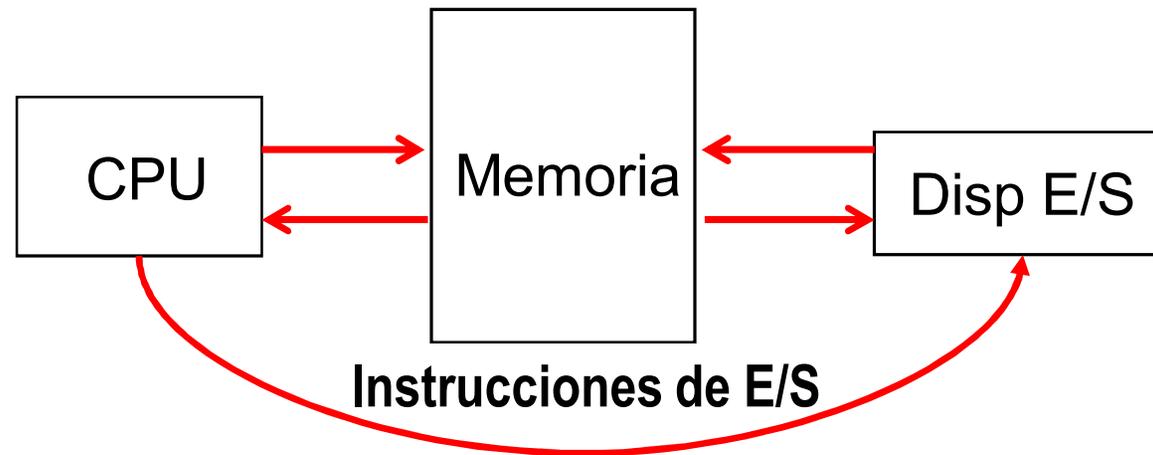


Tabla de estado de dispositivos



Estructura de Acceso Directo a Memoria (DMA)

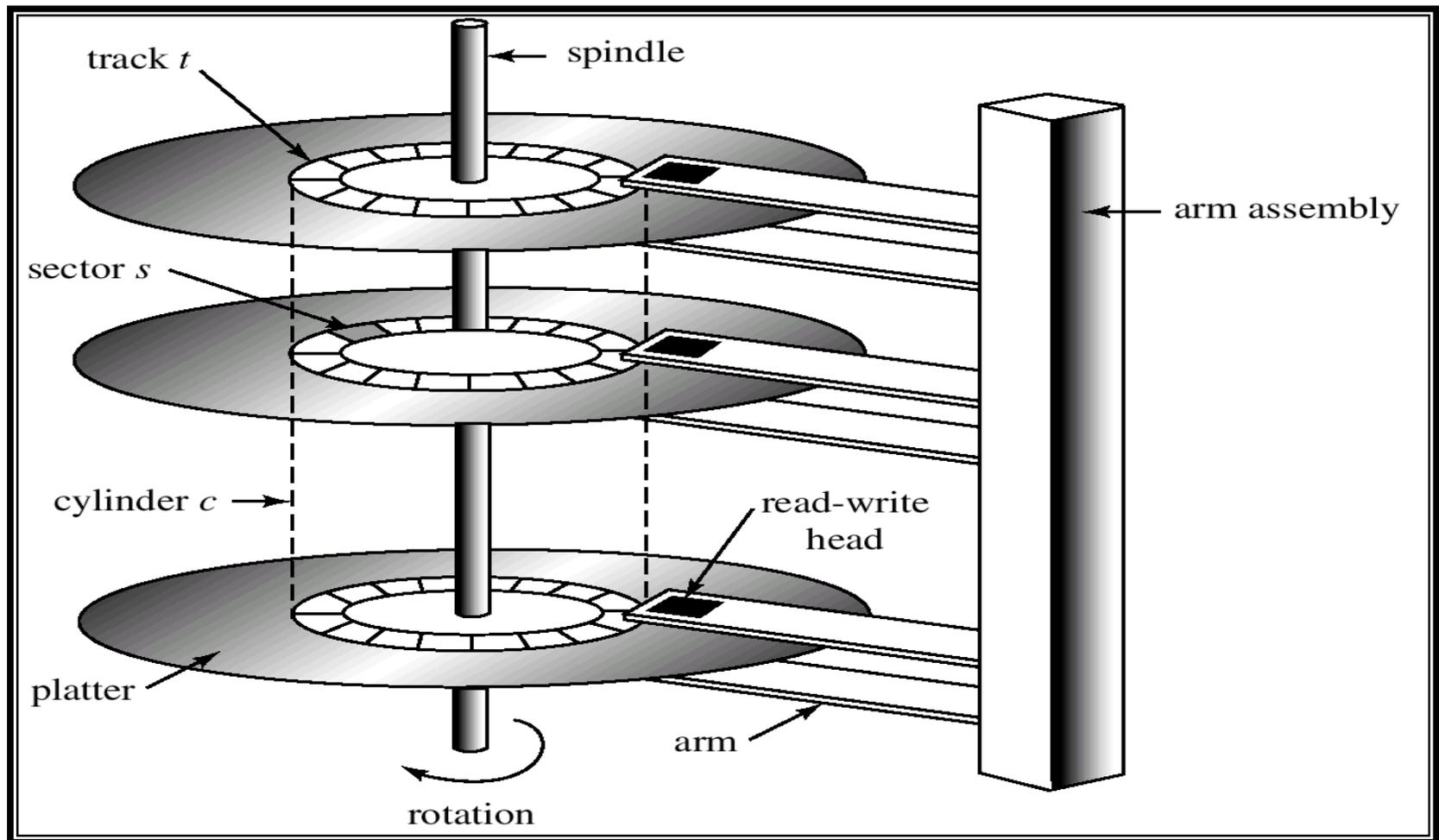


- Usado por dispositivos de E/S de alta velocidad para transmitir información a velocidades similares a la de la memoria.
- El controlador de dispositivos transfiere bloques de datos desde el buffer de almacenamiento directamente a la memoria principal sin la intervención de la CPU.
- Solo una interrupción es generada por bloque, y no una por byte.

Estructura de almacenamiento

- Memoria Principal– único medio de almacenamiento que la CPU puede acceder directamente.
- Almacenamiento Secundario– extensión de la memoria principal que provee una gran capacidad de almacenamiento no volátil.
- Discos Magnéticos–
 - La superficie del disco está lógicamente dividida en *tracks (pistas)*, los cuales están subdivididas en *sectores*.
 - El *controlador de disco* determina la interacción lógica entre el dispositivo y la computadora.

Mecanismo del Movimiento de Cabezas en el Disco

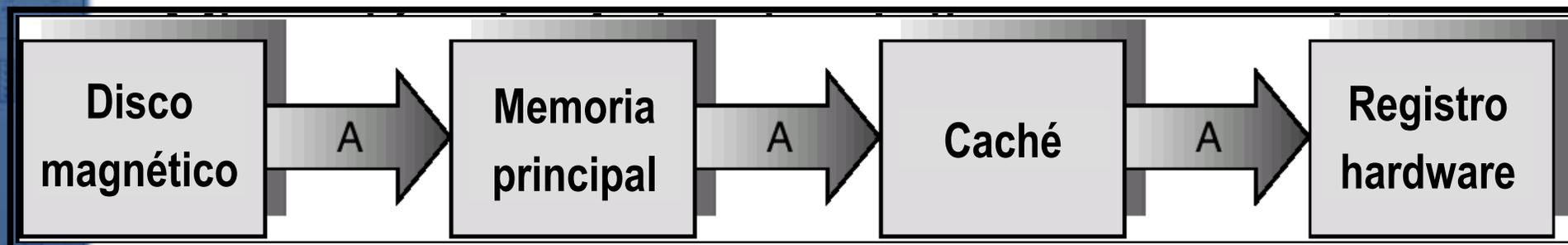


Jerarquía de almacenamiento

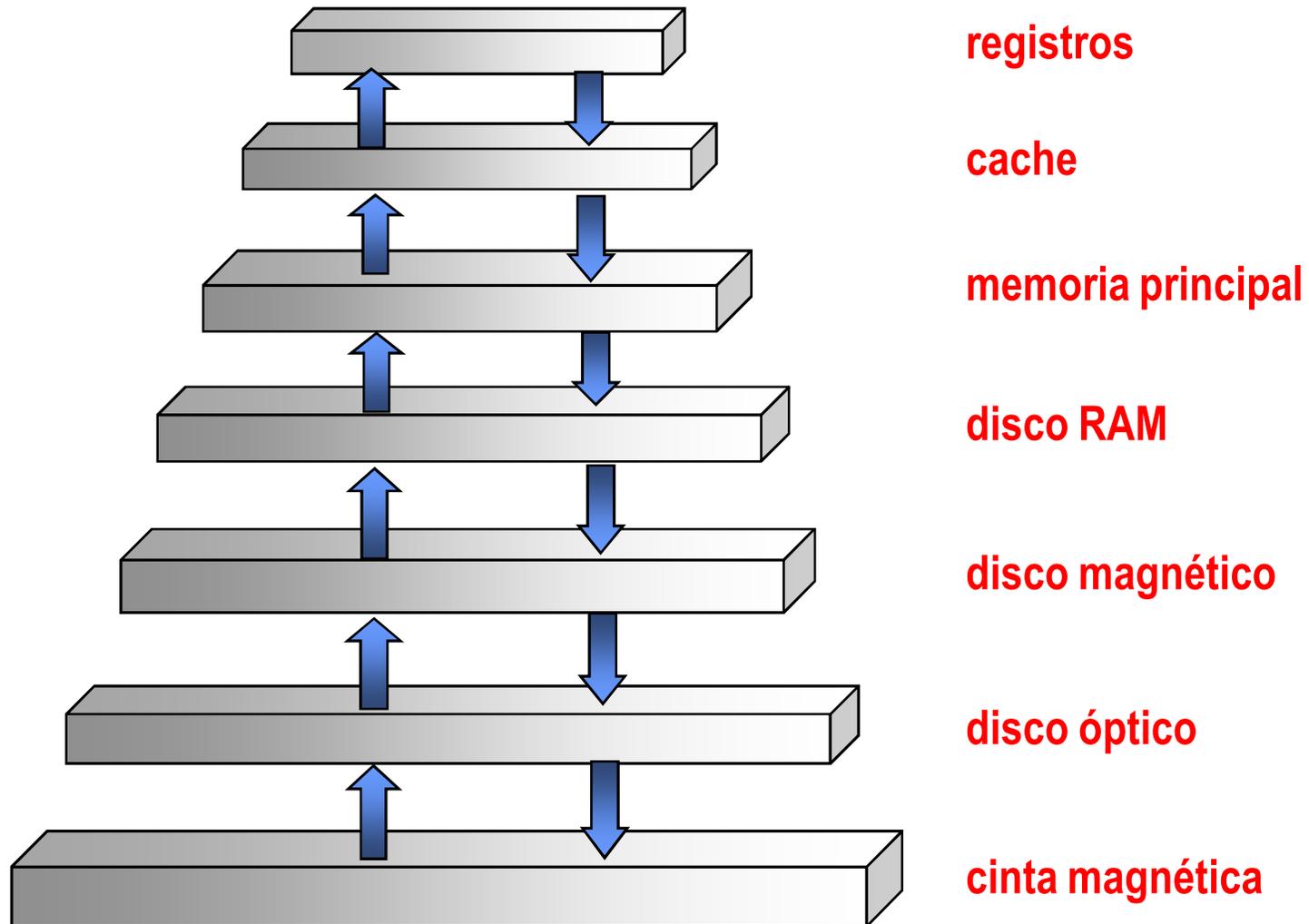
- Los sistemas de almacenamiento está organizado en jerarquía.
 - Velocidad.
 - Costo.
 - Volatilidad.
- *Caching* – información copiada en un sistema de almacenamiento rápido; la memoria principal puede verse como el último *cache* para el almacenamiento secundario.

Caching

- Uso de memoria de alta velocidad para mantener información recientemente accedida.
- Se requiere una política de *manejo de cache*.
- El *caching* introduce otro nivel en la jerarquía de almacenamiento. Esto requiere que datos que están simultáneamente en mas de un nivel sean *consistentes*.



Jerarquía de Dispositivos de Almacenamiento



Protección por Hardware

- Modo Dual de Operación.
- Protección de E/S.
- Protección de Memoria.
- Protección CPU.

Modo Dual de Operación (1)

- Un sistema de recursos compartidos requiere que el SO asegure que un programa incorrecto no pueda causar que otros programas ejecuten incorrectamente.
- Provee soporte de hardware para diferenciar entre, al menos, dos modos de operación.
 1. **Modo Usuario** – la ejecución se hace en beneficio del usuario.
 2. **Modo Monitor** (también *modo supervisor* o *modo sistema*) – la ejecución se hace en beneficio del sistema operativo.

Modo Dual de Operación (2)

- Se agrega un *bit de modo* al hardware de la computadora para indicar el modo corriente : monitor (0) o usuario (1).
- Cuando ocurre una interrupción o *fault* (falta), el hardware cambia a modo monitor.



- Las *instrucciones privilegiadas* pueden ejecutarse sólo en modo monitor.

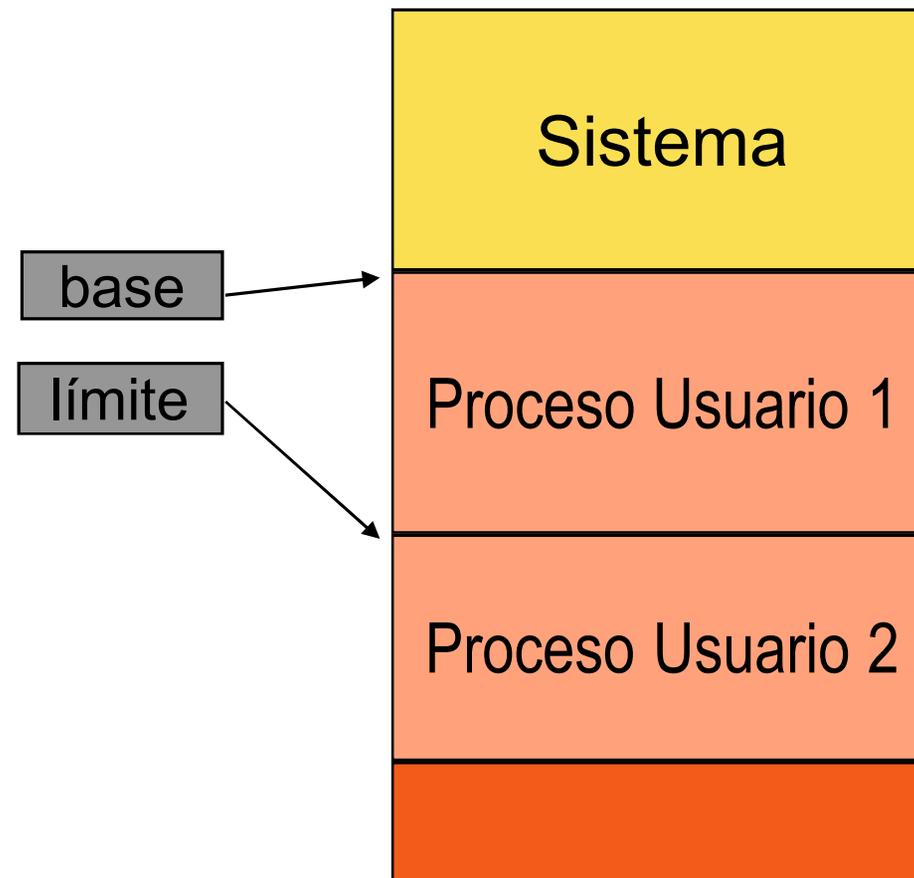
Protección de E/S

- Todas las instrucciones de E/S son instrucciones privilegiadas.
- Se debe asegurar que un programa de usuario no pueda ganar nunca el control de la computadora en modo monitor (e.g. un programa de usuario que, como parte de su ejecución, almacena una nueva dirección en el vector de interrupción).

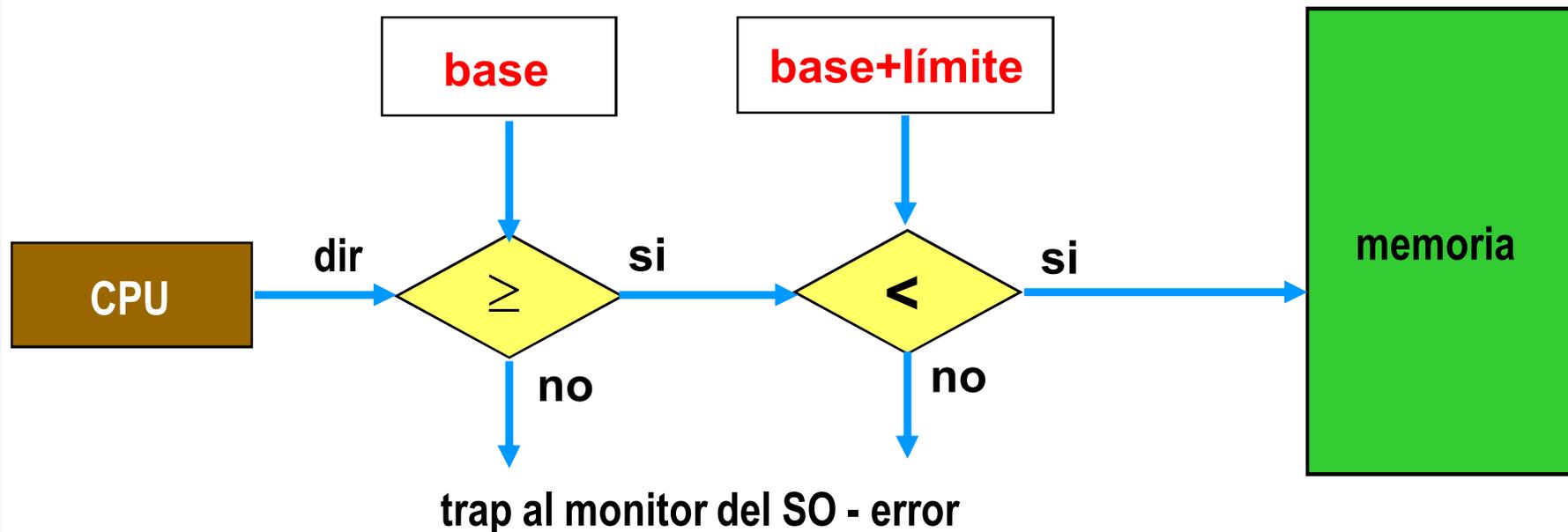
Protección de Memoria

- Debe proveer protección de memoria, al menos para el vector de interrupción y las rutinas de servicio de interrupciones.
- Para tener protección de memoria, se agregan registros que determinan el rango de las direcciones legales a las que un programa puede acceder:
 - **registro base** – mantiene la dirección física legal más chica (más baja).
 - **registro límite** – contiene el tamaño del rango
- Se protege la memoria fuera del rango definido.

Los registros base y límite definen un espacio lógico de direcciones



Protección por Hardware



- Cuando ejecuta en modo monitor, el SO tiene acceso irrestricto al monitor y a la memoria del usuario.
- Las instrucciones de carga para los registros *base* y *límite* son instrucciones privilegiadas (obvio).

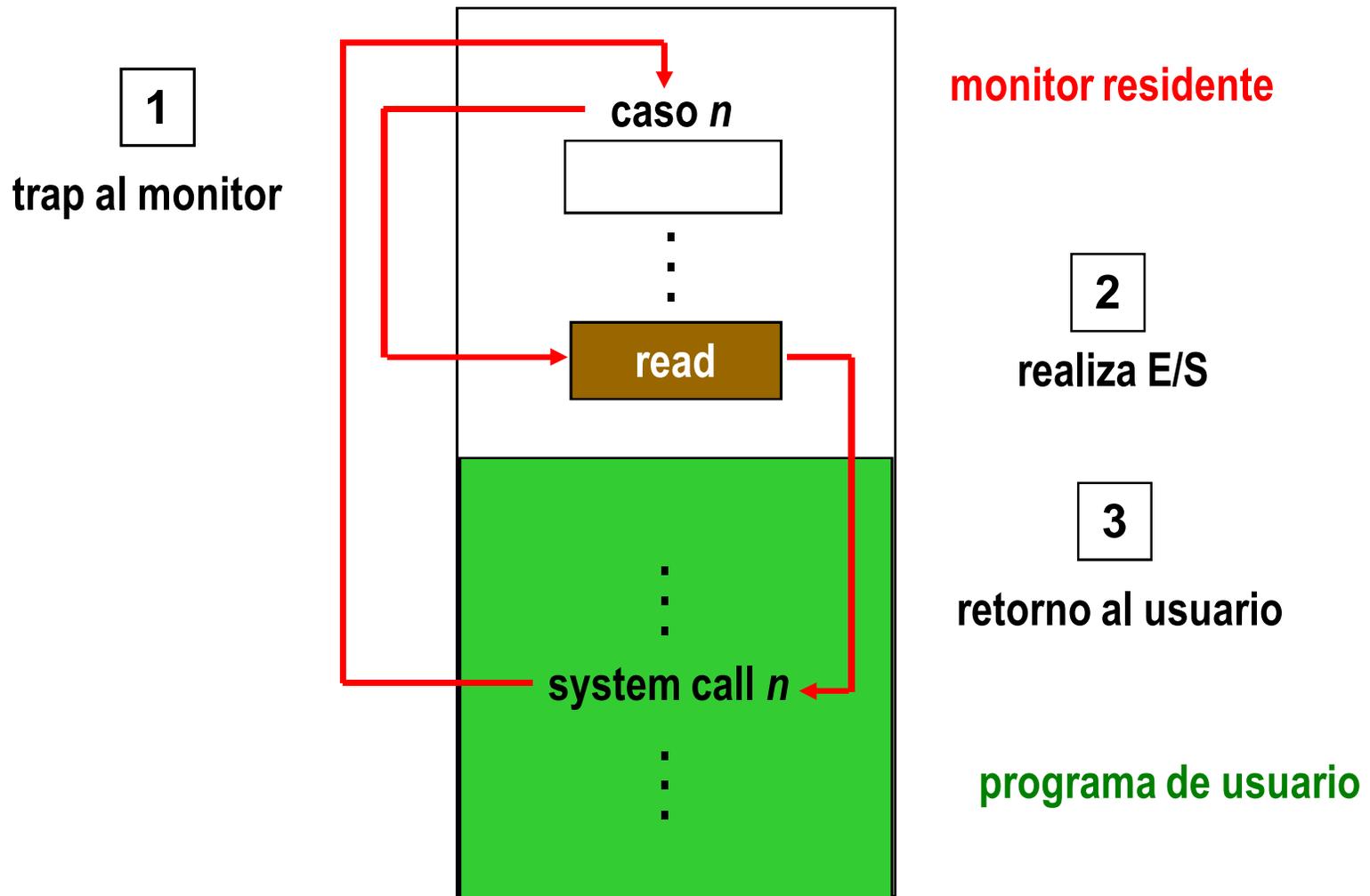
Protección de CPU

- *Timer* – interrumpe a la computadora luego de un período específico para asegurar que el SO mantenga el control.
 - El *timer* es decrementado cada *tick* de reloj.
 - Cuando el *timer* alcanza el valor 0, ocurre una interrupción.
- El *timer* es comunmente usado para implementar el tiempo compartido.
- El *timer* también es usado para computar el tiempo corriente.
- La carga del *timer* es una instrucción privilegiada (obvio).

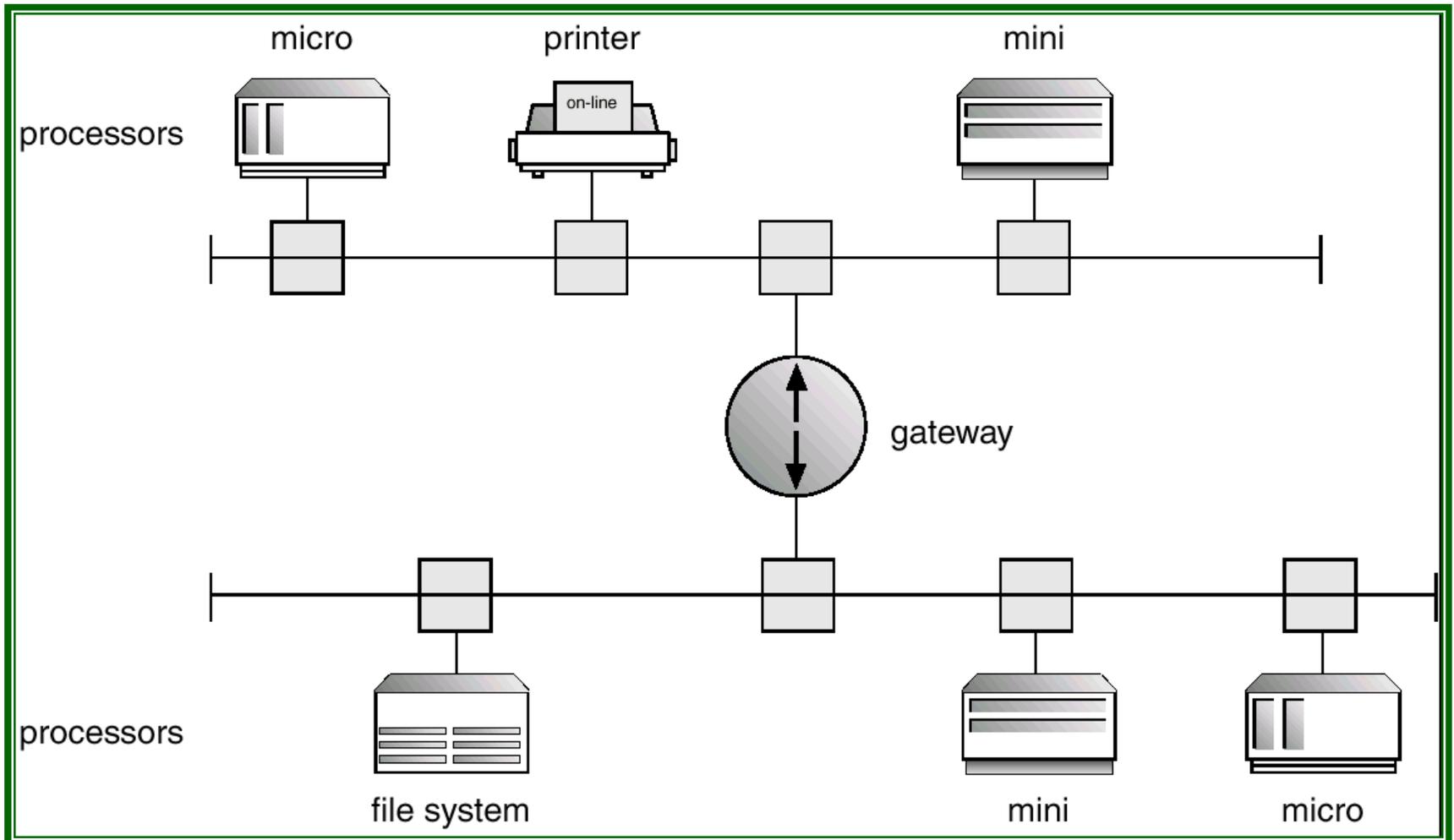
Arquitectura General del Sistema

- Dado que las instrucciones de E/S son privilegiadas, ¿cómo hace el usuario para ejecutar una E/S?
- **Llamada al sistema (system call)** – es el método usado por un proceso para requerir una acción del SO.
 - Usualmente toma la forma de un *trap* a una locación específica en el vector de interrupción.
 - El control pasa por medio del vector de interrupción a la rutina de servicio en el SO, y el bit de modo es puesto en modo monitor.
 - El monitor verifica que los parámetros son correctos y legales, ejecuta el requerimiento, y retorna el control a la siguiente instrucción de la llamada al sistema.

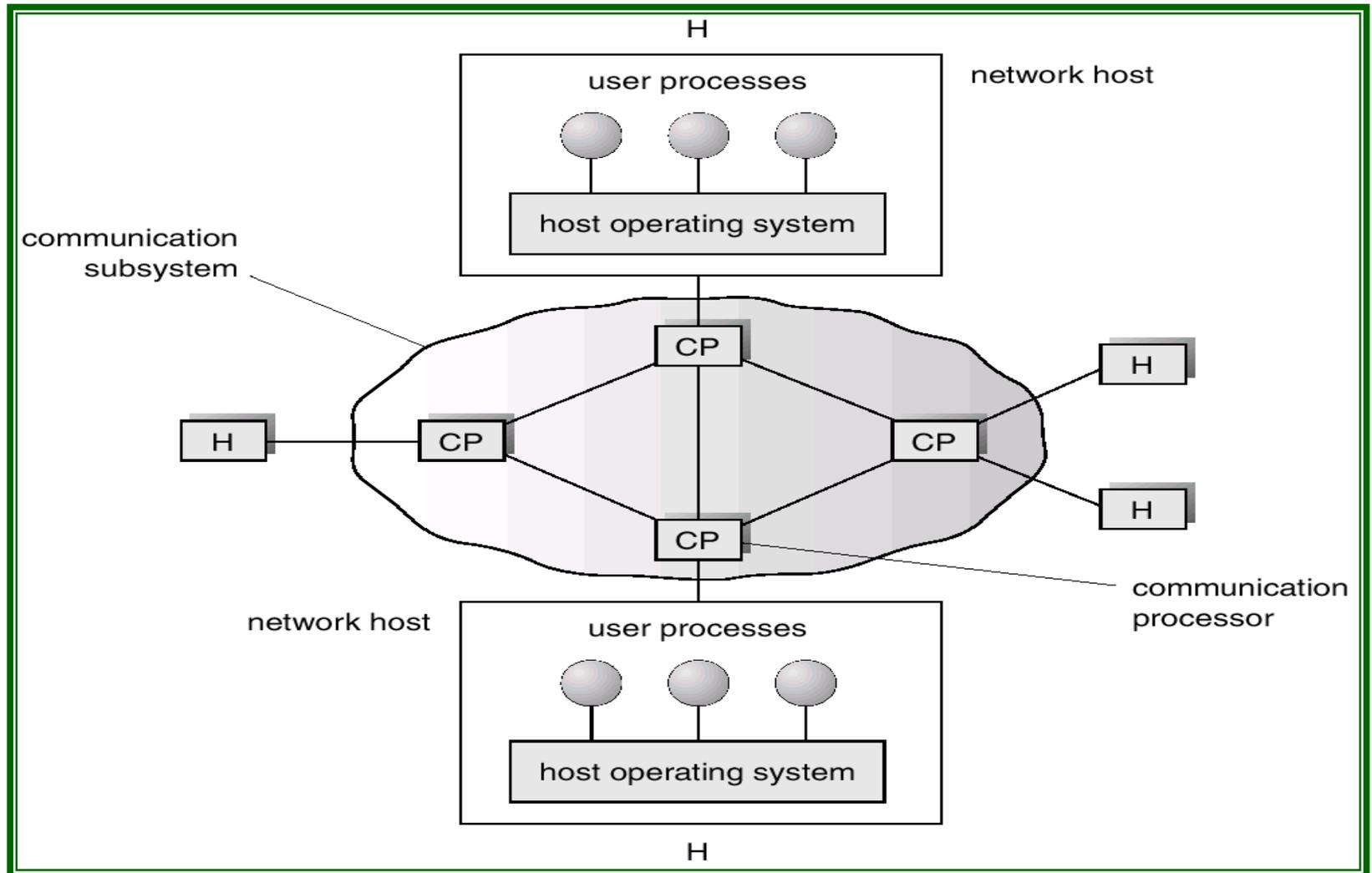
E/S mediante *system call*



Redes de Área Local (LAN: *Local Area Network*)



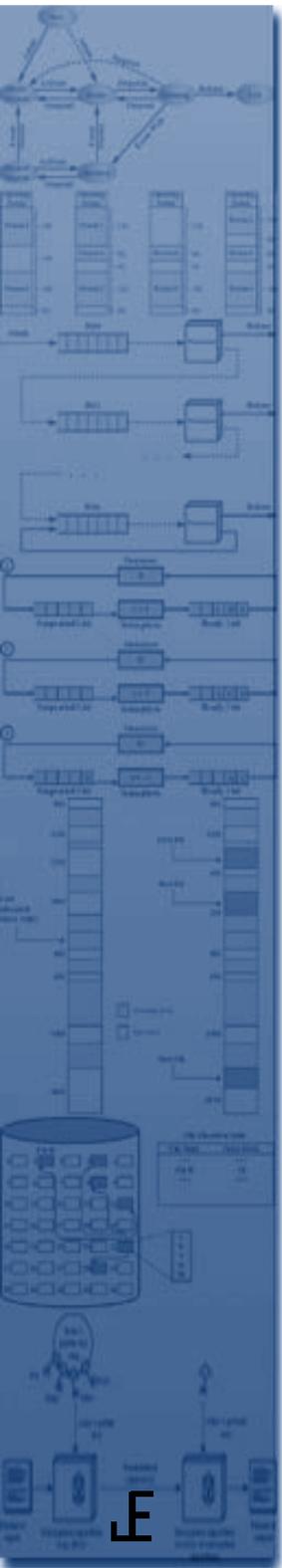
Redes de Área Amplia (WAN: *Wide Area Network*)



Más sobre Redes?

- Ver [http://en.wikipedia.org/wiki/Computer network](http://en.wikipedia.org/wiki/Computer_network)
 - Pequeña intro a redes, obviamente estos temas se expanden en el curso específico de Redes!

Estructura de los Sistemas Operativos



Administración de Procesos

- **Un *proceso* es un programa en ejecución.** Un proceso necesita ciertos recursos, incluyendo tiempo de CPU, memoria, archivos, y dispositivos de E/S, para llevar a cabo su tarea.
- El sistema operativo es responsable por las siguientes actividades relacionadas con la administración de procesos.
 - Creación y eliminación de procesos.
 - Suspensión y reactivación de procesos.
 - Provisión de mecanismos para:
 - sincronización de procesos.
 - comunicación de procesos.

Administración de Memoria Principal

- La memoria es un gran arreglo de palabras o bytes, cada una con su propia dirección. Es un repositorio de datos rápidamente accesibles compartidos por la CPU y los dispositivos de E/S.
- La memoria principal es un dispositivo de almacenamiento volátil. Pierde su contenido en caso de falla en el sistema, e.g. corte eléctrico.
- El sistema operativo es responsable de las siguientes actividades relacionadas con la administración de memoria:
 - Lleva control de qué partes de la memoria están siendo usadas y por quién.
 - Decide que procesos cargar cuando hay espacio de memoria disponible.
 - Ocupa y desocupa espacio de memoria cuando necesite.

Administración de Almacenamiento Secundario

- Dado que la memoria principal (*almacenamiento primario*) es volátil y demasiado pequeño para acomodar todos los datos y programas permanentemente, el sistema de la computadora debe proveer *almacenamiento secundario* para respaldar la memoria principal.
- La mayoría (todos!) de los sistemas de cómputo modernos usan discos como medio de almacenamiento en línea, tanto para datos como para programas.
- El sistema operativo es responsable de las siguientes actividades relacionadas con la administración de discos:
 - Administración del espacio libre
 - Asignación del almacenamiento
 - Planificación del disco (no hoy en día)

Sistema de Administración de E/S

- El sistema de E/S consiste de:
 - Un sistema de *caching* (buffers).
 - Una interfaz general de *driver* de dispositivo.
 - *Drivers* para dispositivos específicos de hardware.

Administración de Archivos

- Un archivo es una colección de información relacionada definida por su creador. Comúnmente, los archivos representan programas (sea en su forma fuente u objeto) y datos.
- El sistema operativo es responsable de las siguientes actividades relacionadas con la administración de archivos:
 - Creación y destrucción de archivos (`unlink(2)`).
 - Creación y destrucción de directorios (`unlinkat(2)`).
 - Soporte de primitivas para el manejo de archivos y directorios (`mkdir(2)`).
 - Mapeo de archivos sobre el almacenamiento secundario.
 - Respaldo sobre medios de almacenamientos estables.

Sistemas de Protección

- *Protección* se refiere a los mecanismos para controlar el acceso por programas, procesos, o usuarios del sistema y usuarios de recursos.
- El mecanismo de protección debe:
 - Distinguir entre uso autorizado y no autorizado.
 - Especificar el control a imponer.
 - Proveer medios de imposición.
 - Implementar mediación completa.

Sistemas de Intérpretes de Comandos

- Muchos comandos son pasados al sistema operativo por medio de sentencias de control que llevan a:
 - Creación y manejo de procesos.
 - Manejo de E/S.
 - Administración de almacenamiento secundario.
 - Manejo de la memoria principal.
 - Acceso al sistema de archivos.
 - Protección.
 - Redes.

Sistemas de Intérpretes de Comandos

- El programa que lee e interpreta sentencias de control es llamado de diferentes maneras:
 - Intérprete de tarjetas de control (old!)
 - Intérprete de líneas de comando
 - **Shell** (en SO UNIX like)

Su función es tomar y ejecutar la sentencia de comando siguiente.

Servicios del Sistema Operativo

- Ejecución de Programas – capacidad del sistema para cargar programas en memoria y ejecutarlos.
- Operaciones E/S – dado que los programas de usuario no pueden ejecutar operaciones de E/S directamente, el sistema operativo debe proveer alguna forma de realizar E/S.
- Manipulación del sistema de archivos – capacidad de los programas para leer, escribir, crear y borrar archivos.
- Comunicaciones – intercambio de información entre procesos ejecutando en la misma computadora o en diferente sistema (vía red). Implementadas vía *memoria compartida* o *pasaje de mensajes*.
- Detección de Errores – asegura la correcta computación detectando errores en la CPU y hardware de memoria, en dispositivos de E/S, o en programas de usuario.

Funciones Adicionales del Sistema Operativo

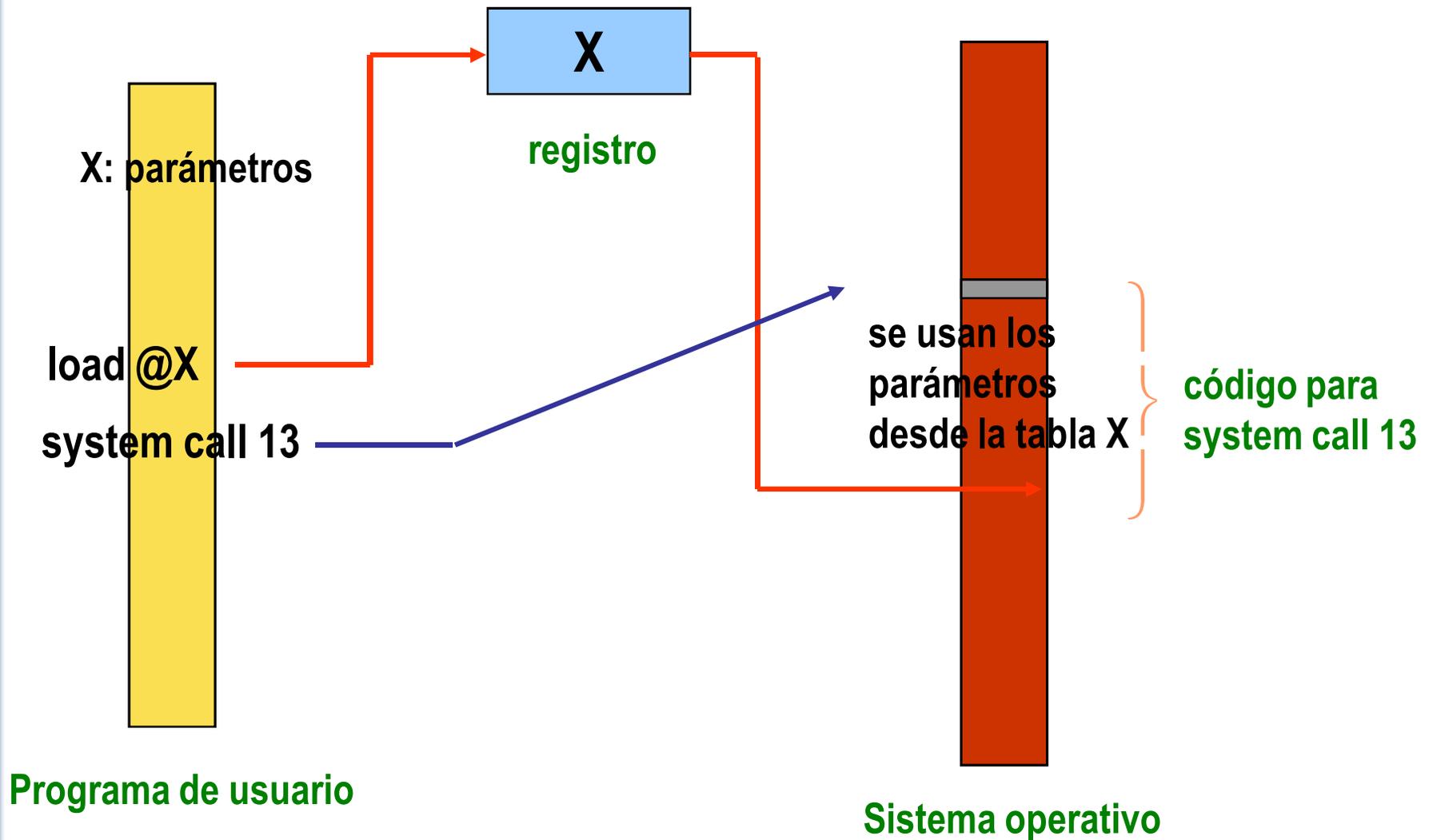
Las funciones adicionales existen no sólo para ayuda del usuario sino también para asegurar la eficiencia de las operaciones del sistema.

- Gestión de recursos – asigna recursos a múltiples usuarios o múltiples tareas ejecutando al mismo tiempo.
- Contabilidad – tiene en cuenta y registra que clase de recursos y cuanto usa cada usuario para facturación o (usualmente) estadística.
- Protección – asegura que todos los accesos al sistema están controlados.

Llamadas al Sistema (*System Calls*)

- Las **llamadas al sistema** proveen una interfaz entre el programa ejecutando y el sistema operativo.
 - Generalmente disponible como instrucciones assembler.
 - Lenguajes, definidos en reemplazo del lenguaje ensamblador para programadores de sistemas, que permiten hacer llamadas a sistema en forma directa
 - e.g., la mayoría de las implementaciones de lenguaje C.
- Se usan en general tres métodos para pasar parámetros entre programas en ejecución y el sistema operativo.
 - Pasaje de parámetros en *registros*.
 - Se almacenan los parámetros en una tabla en memoria, y la dirección de la misma es pasada como parámetro en un registro.
 - El programa *push* (almacena) los parámetros en un *stack*, y son *pop* (retirados) del stack por el sistema operativo.

Pasaje de Parámetros por Tabla



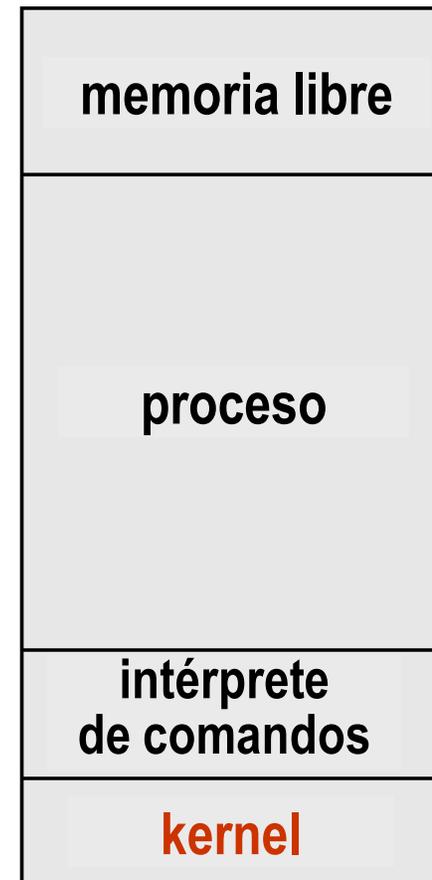
Ejecución MS-DOS

Inicio



(a)

Programa ejecutando



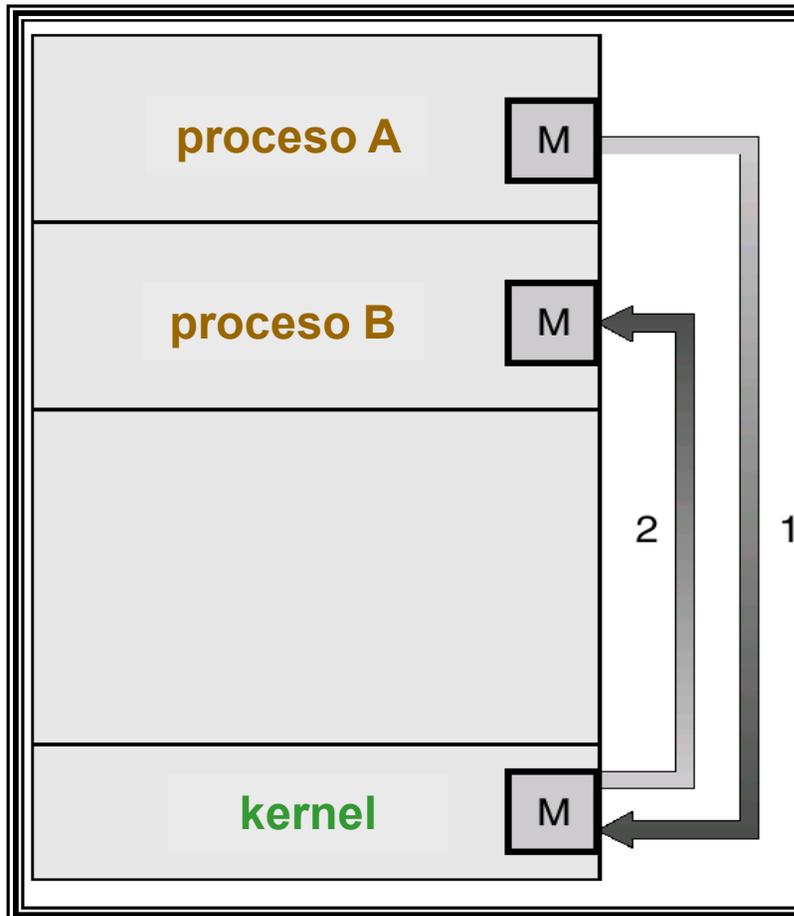
(b)

UNIX Ejecutando Múltiples Programas

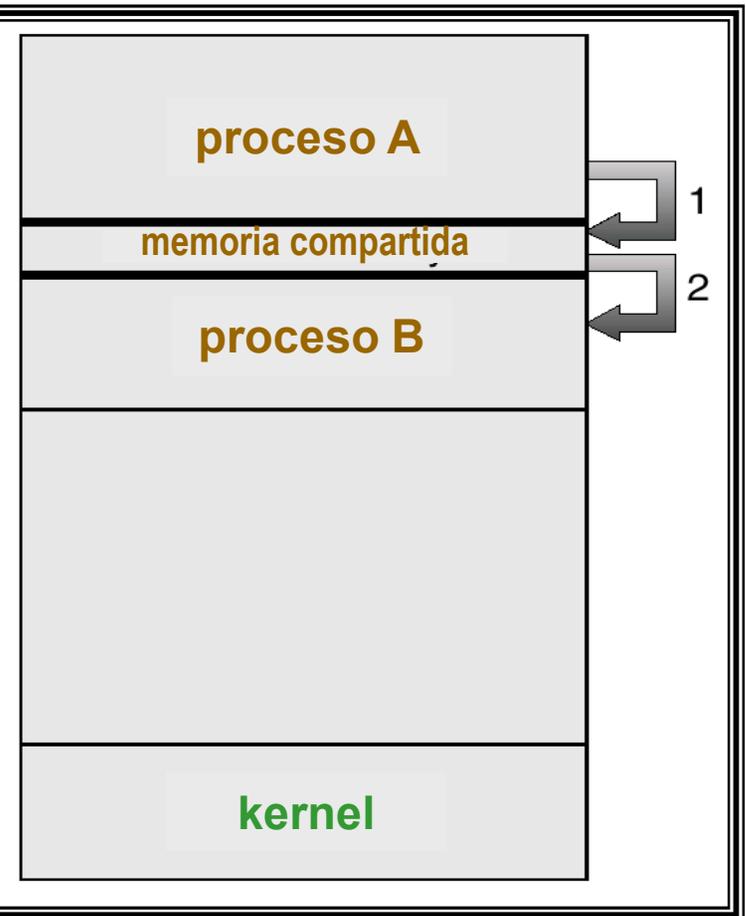


Modelos de Comunicación

Pasaje de mensajes



Memoria compartida

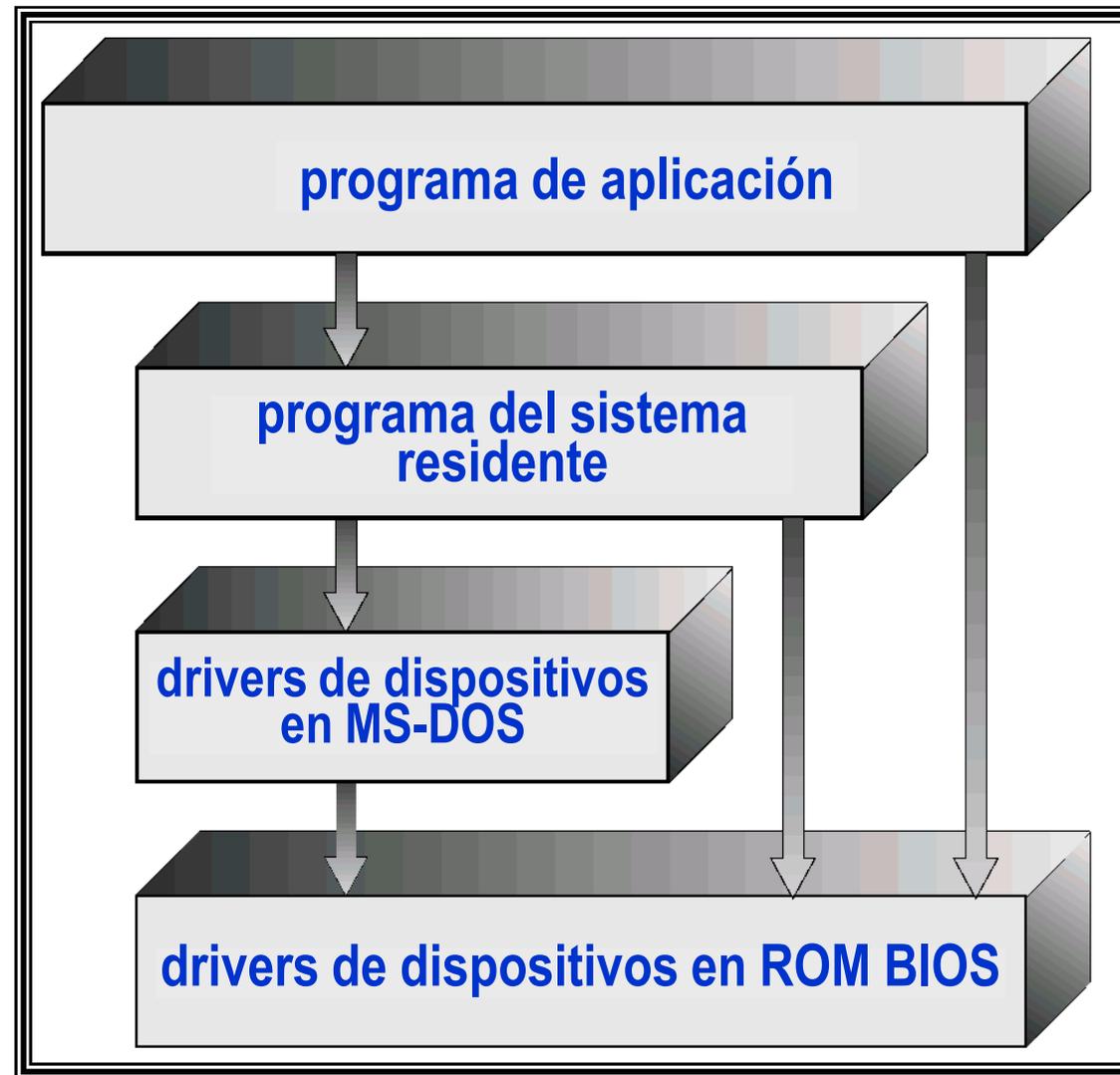


Programas de Sistema

- Los programas de sistema proveen un medio conveniente para el desarrollo de programas y ejecución. Pueden ser divididos en:
 - ◆ Manipulación de archivos
 - ◆ Información de estado
 - ◆ Modificación de archivos
 - ◆ Soporte de lenguajes de programación
 - ◆ Carga de programas y ejecución
 - ◆ Comunicaciones
 - ◆ Programas de aplicación

La visión que tienen la mayoría de los usuarios del sistema operativo está dada por los programas de sistema y no por las llamadas a sistema (system calls).

Capas de la Estructura MS-DOS



Estructura de UNIX

USUARIOS

shells y comandos
compiladores e intérpretes
librerías de sistema

interfaz de system-call al kernel

signals
manejo terminal
sistema I/O caracter
drivers terminal

sistema de archivos
swapping
sistema I/O bloque
drivers disco y cinta

planificación CPU
reemplazo de páginas
demanda de páginas
memoria virtual

interfaz kernel al kernel

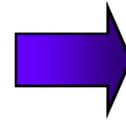
controladores terminal
terminales

controladores disp
discos y cintas

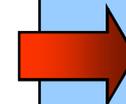
controladores memoria
memoria física

Un Sistema Operativo por Capas

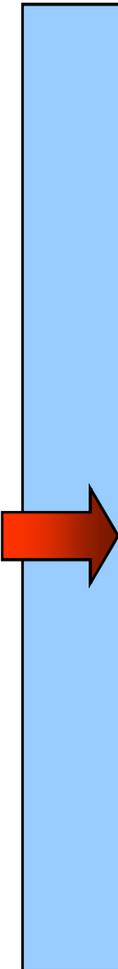
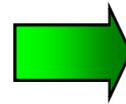
nuevas
operaciones



operaciones
ocultas



operaciones
existentes

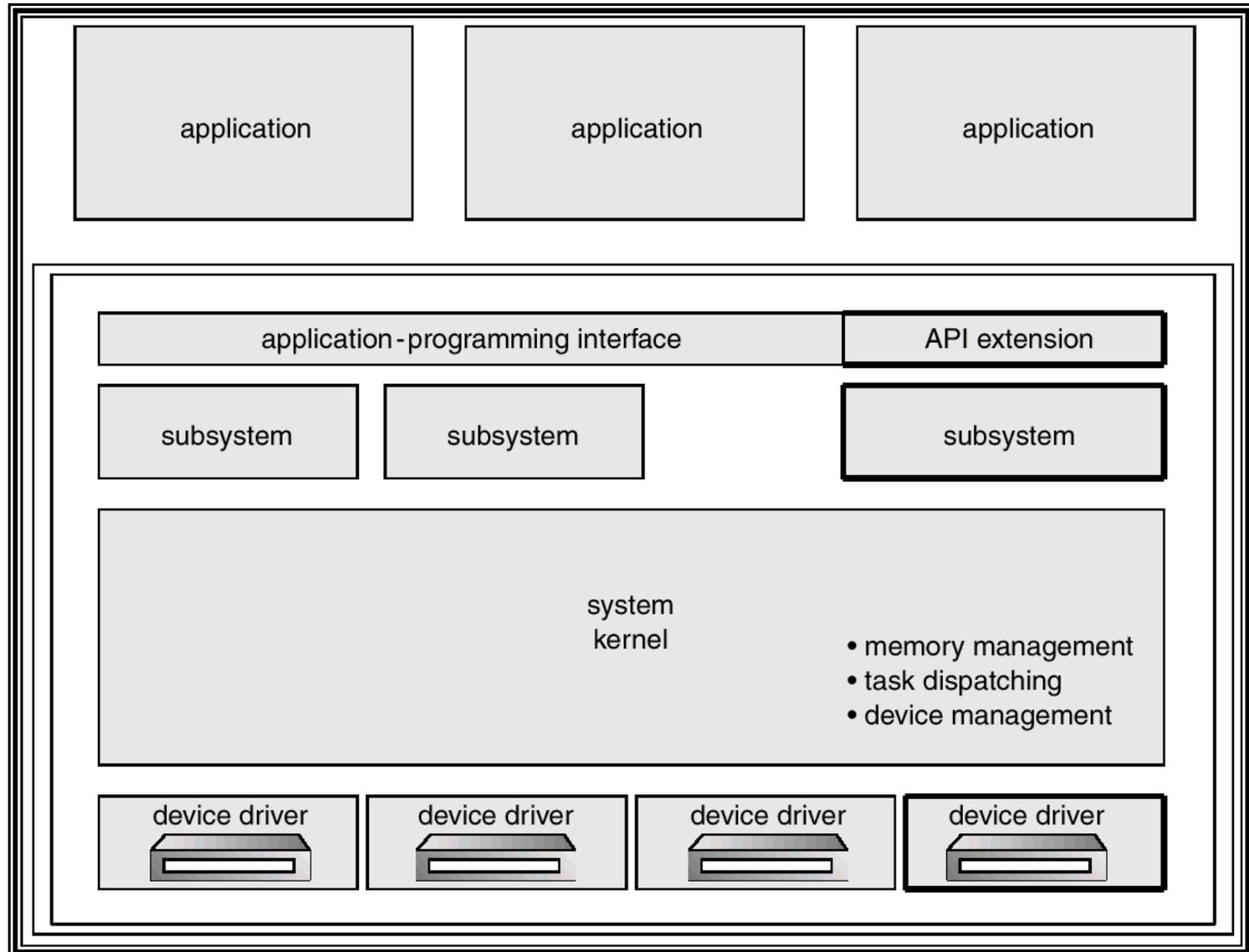


Capa *M*



Capa *M-1*

Estructura en Capas de OS/2



Estructura de Microkernels

- Núcleo pequeño de sistema operativo
- Contiene sólo operaciones esenciales de las funciones del sistema operativo
- Muchos de los servicios tradicionalmente incluidos en el sistema operativo son ahora subsistemas externos:
 - Drivers de dispositivos
 - Sistemas de archivos
 - Manejo de memoria virtual
 - Sistema de ventanas
 - Servicios de seguridad

Microkernels

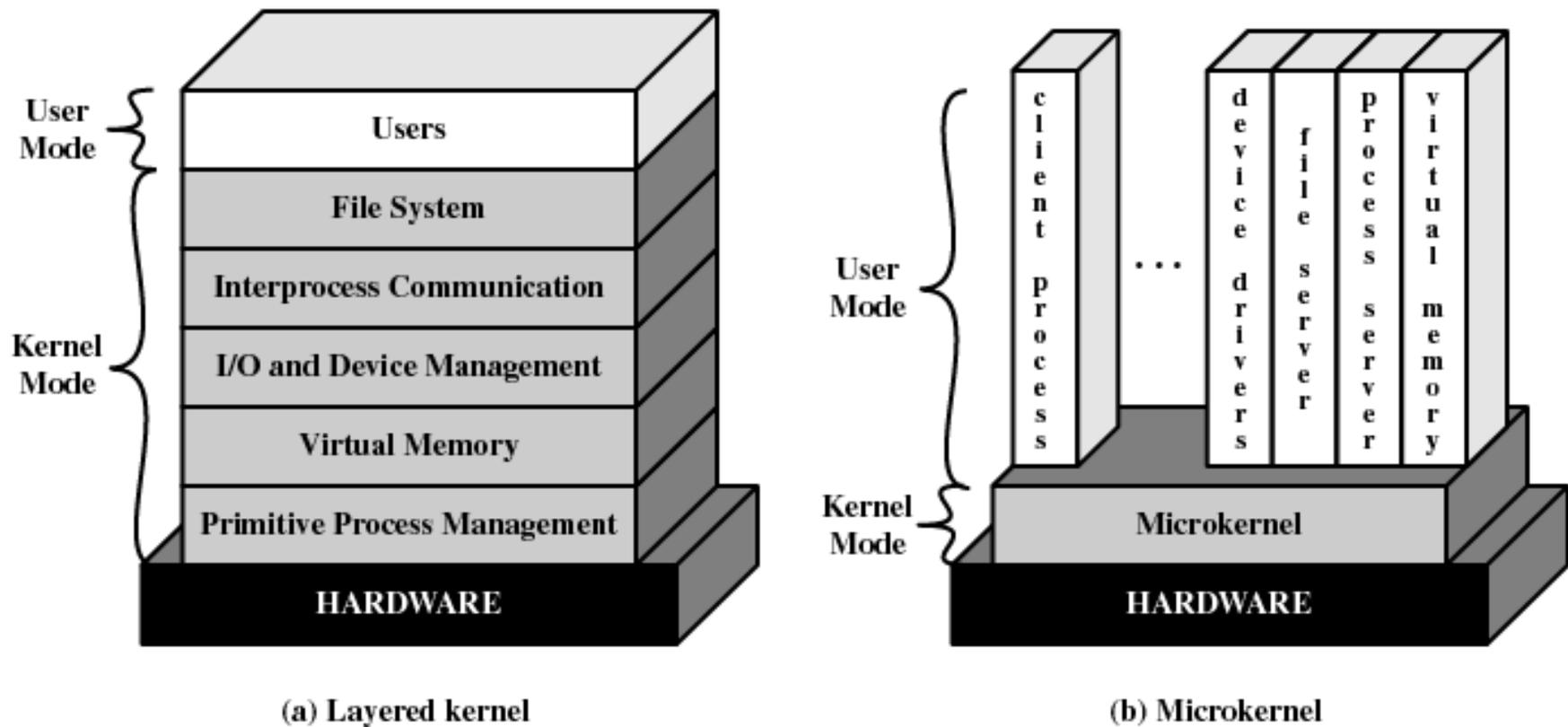


Figure 4.10 Kernel Architecture

Beneficios de una Organización Microkernel

- Interfaz uniforme en requerimientos hechos por un proceso.
 - No distingue entre servicios nivel kernel y nivel usuario.
 - Todos los servicios están provistos por medio de pasajes de mensajes.
- Extensibilidad.
 - Permite al agregado de nuevos servicios.
- Flexibilidad.
 - Nuevas características adicionales.
 - Características existentes pueden ser eliminadas.

Beneficios de una Organización Microkernel

- **Portabilidad**
 - Los cambios necesarios para portar el sistema a un nuevo procesador es cambiado en el microkernel - no en los otros servicios
- **Confiabilidad**
 - Diseño modular
 - Un microkernel puede ser rigurosamente verificado
- **Soporte de sistema distribuido**
 - Los mensajes son enviados sin conocer cuál es la máquina destino
- **Sistema operativo orientado a objetos**
 - Los componentes son objetos con interfaces claramente definidas que pueden ser interconectadas para formar el software

Diseño de Microkernel

- Manejo de memoria bajo nivel.
 - Mapea cada página virtual al frame de página física.
- Comunicación Interprocesos (IPC).
- Administración de E/S e interrupciones.

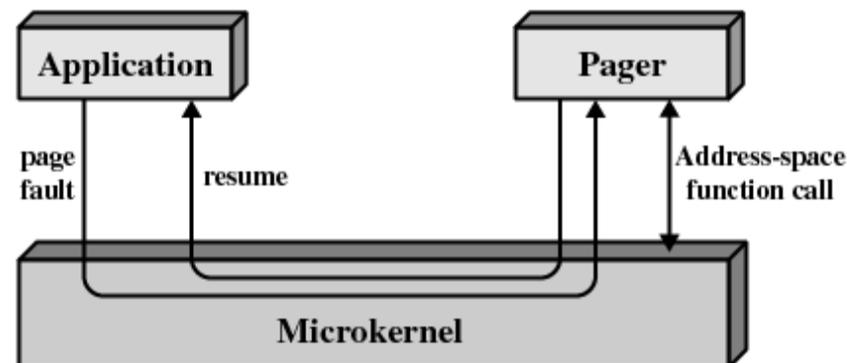
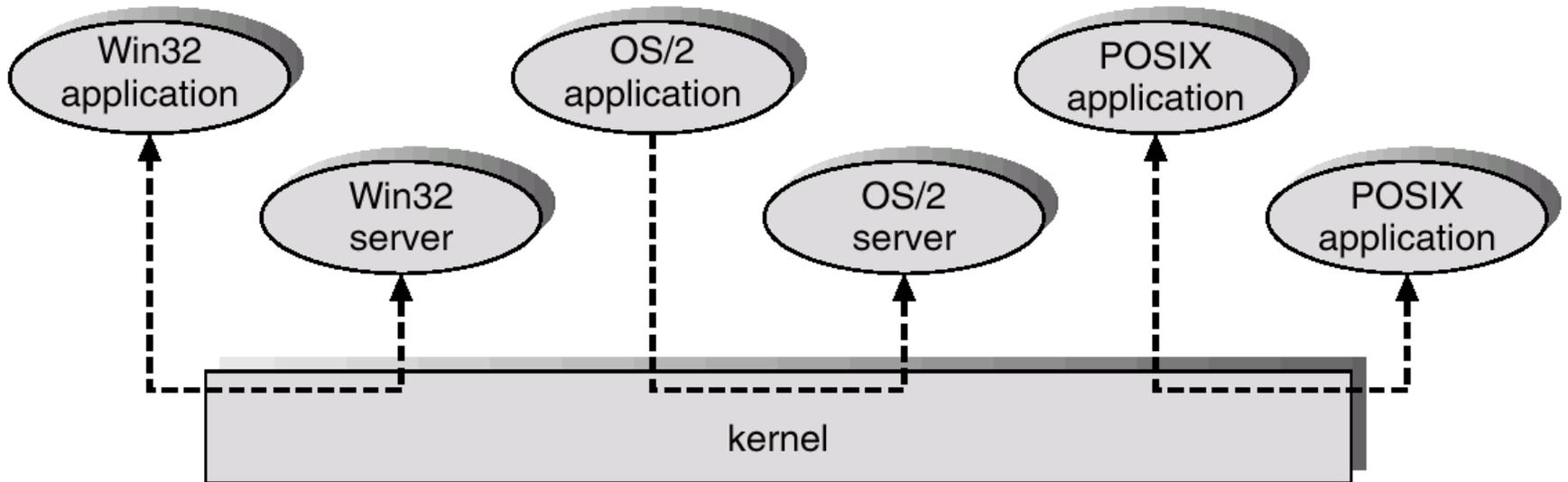


Figure 4.11 Page Fault Processing

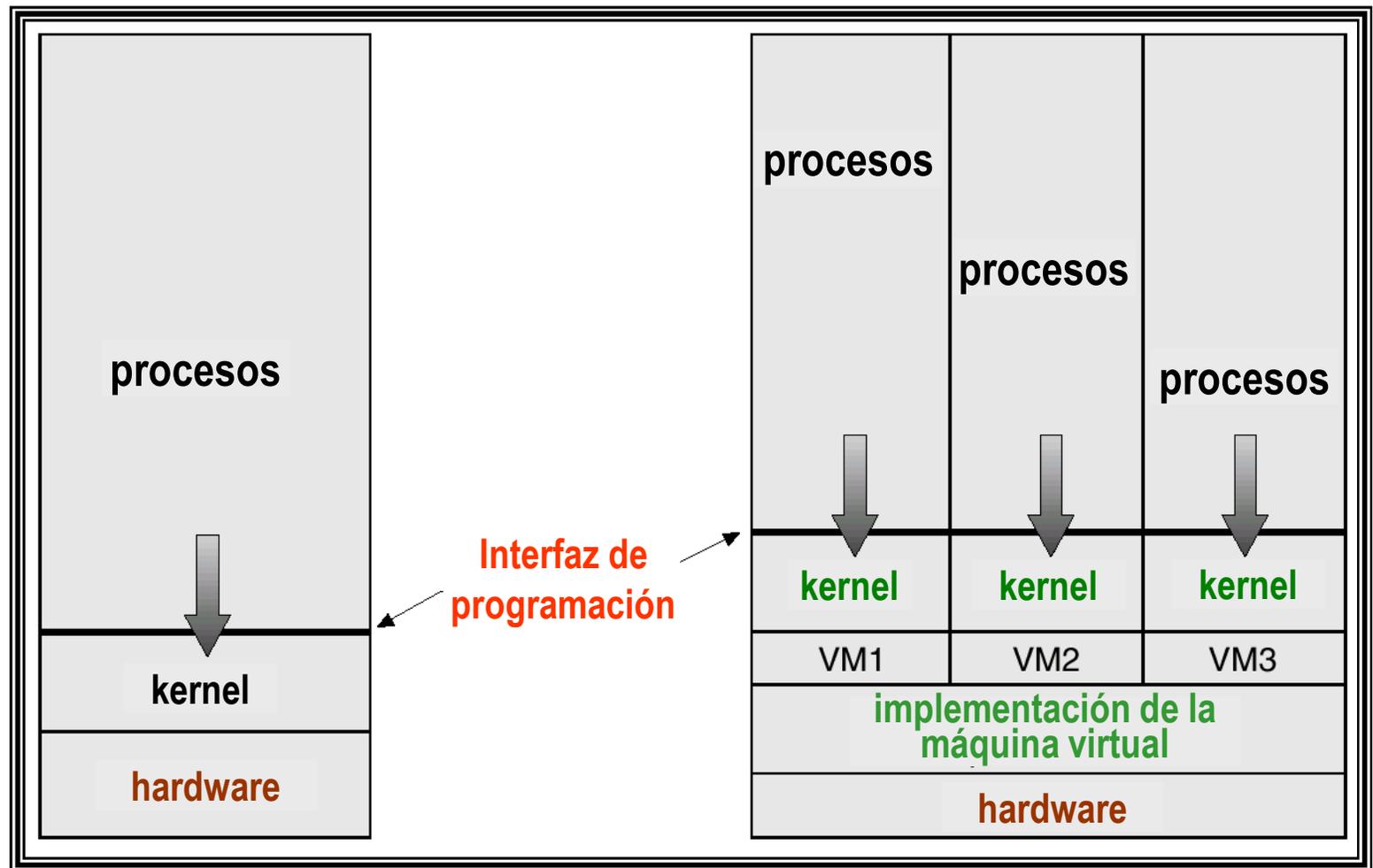
Estructura Cliente-Servidor M\$ Windows NT/2000/XP/Vista/Seven



Modelos de Sistema

Máquina real (no virtual)

Máquina virtual



Virtualización

- Hypervisor vs. Paravirtualización.
- Virtualización a nivel del sistema operativo.
- Exokernel / kernels híbridos / nanokernel... Mucho más que kernel monolítico vs. microkernel.
- Volveremos (temas hot!).

Objetivos de Diseño del Sistema

- **Objetivos del Usuario** – el sistema operativo debería ser conveniente para usar, fácil de aprender, confiable, seguro y rápido.
- **Objetivos del Sistema** – el sistema operativo debería ser fácil de diseñar, implementar y mantener y a su vez flexible, confiable, libre de errores y eficiente.

Mecanismos y Políticas

- Los *mecanismos* determinan cómo hacer algo, las *políticas* deciden qué debe ser hecho.
- La separación entre políticas y mecanismos es un principio muy importante, permite máxima flexibilidad si es que las decisiones políticas tienen que ser cambiadas posteriormente.

Implementación del Sistema

- Tradicionalmente se escribía en lenguaje ensamblador, actualmente son escritos en lenguajes de alto nivel (C, no Java!).
- El código escrito en lenguaje de alto nivel:
 - puede ser escrito muy rápido.
 - es más compacto.
 - es mas fácil de entender y depurar.
- Un sistema operativo es más fácil de *portar* (moverlo a otro hardware) si está escrito en un lenguaje de alto nivel.

Coming
Next

