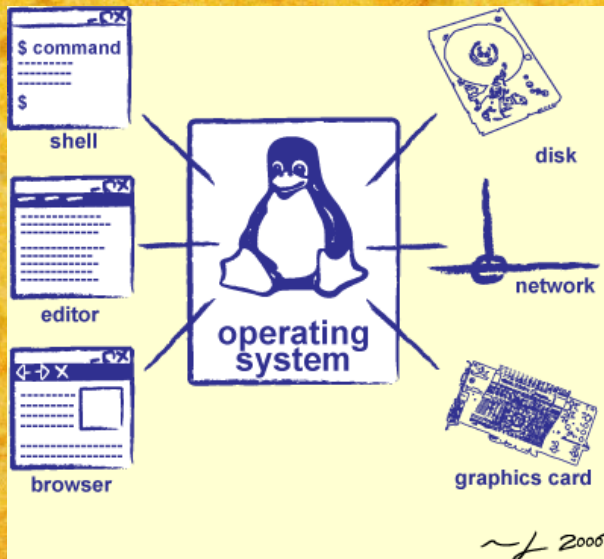


# 1.- Introducción



# Capítulo 1: Introducción

- Definición
- Organización de Computadoras
- Arquitectura de Computadoras
- Estructura del Sistema Operativo
- Operaciones del Sistema Operativo
- Procesos
- Memoria
- Almacenamiento

# Objetivos

- Intro a los componentes principales de SOs.
- Proporcionar un entendimiento básico de cómo funciona una computadora en general

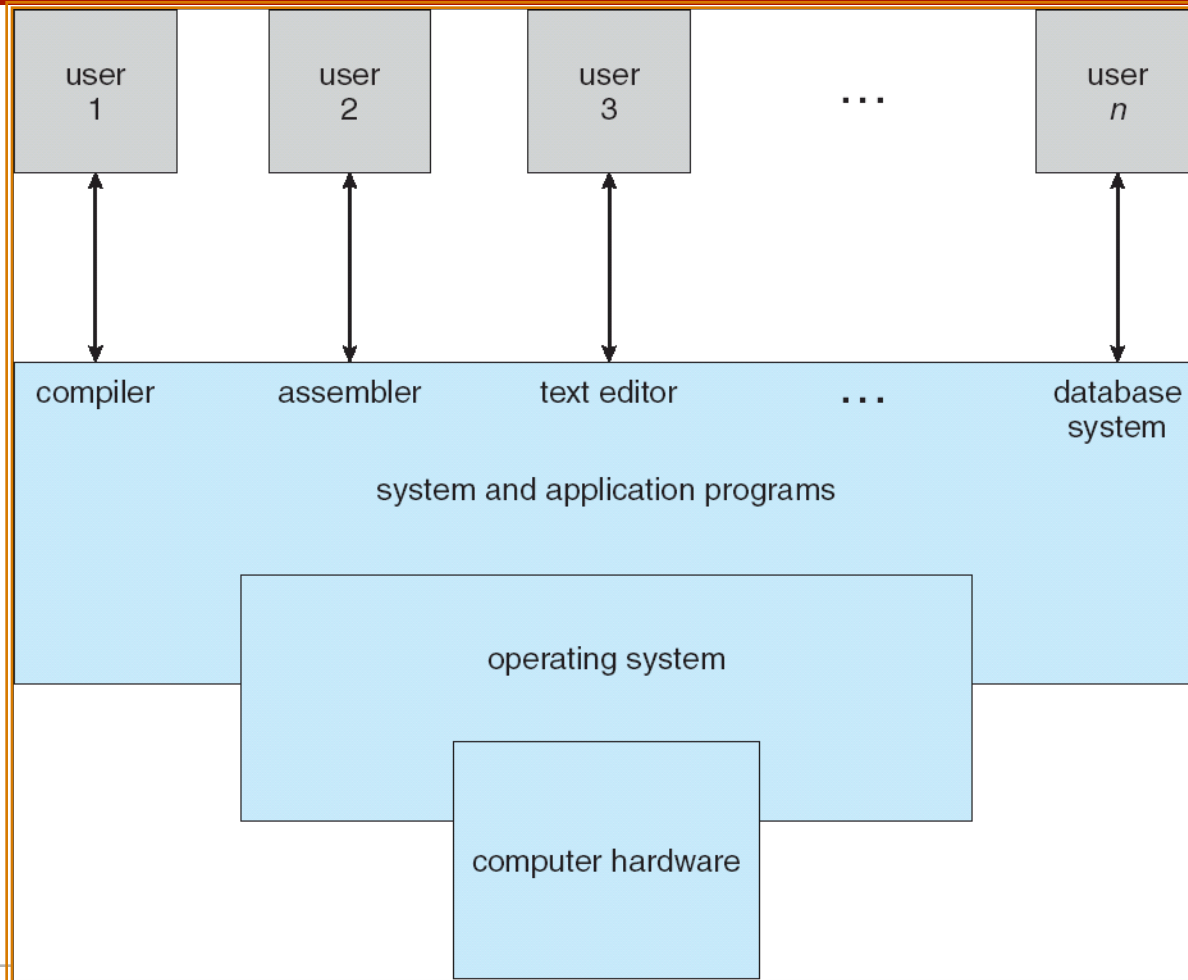
# ¿Qué es un SO?

- Programa intermediario usuario / hardware.
- Metas:
  - Ejecutar programas – facilitar solución de problemas
  - Hacer “usable” el sistema (computadora)
- Usar el hardware eficientemente

# Estructura de Computadoras

- Componentes de una computadora
  - Hardware – recurso básico de cómputo
    - CPU, memoria, dispositivos de I/O
  - Sistema Operativo
    - Controla y coordina el uso de hardware aplicaciones y usuarios
  - Aplicaciones (programas) – define como usar los recursos del sistema para reasolver los problemas de cómputo del usuario
    - Word, compiladores, web browsers, bases de datos
  - Usuarios
    - Humanos, máquinas, otras computadoras

# 4 Componentes de una Computadora



# SO – Definición

- SO es un **administrador de recursos**
  - Administra todos los recursos
  - Resuelve conflictos en requisiciones
  - Asegura un uso justo y eficiente
- SO es un **programa de control**
  - Controla la ejecución de programas para prevenir errores y uso impropio de los recursos

# SO – Definición (Cont.)

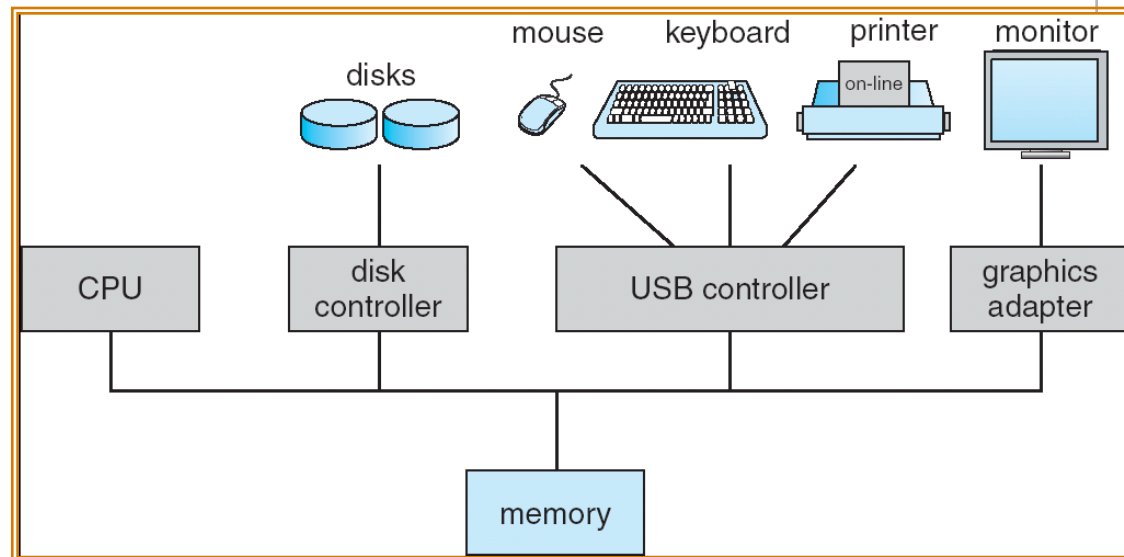
- No hay una definición aceptada universalmente
- “El software que te entrega el vendedor cuando compras una computadora”
- “El programa que está corriendo todo el tiempo en la computadora” es el **kernel**.
- El resto es un programa del sistema (viene con el SO) o un programa de aplicación

# Al inicio (Startup)

- **Programa bootstrap** se carga al encender o “rebotar”
  - Típicamente se almacena en ROM o EPROM.
  - Se conoce como **firmware**
  - Inicializa todo aspecto del sistema
  - Carga el kernel del SO y comienza su ejecución

# Organización de la Computadora

- Operación del sistema
  - 1+ CPUs y dispositivos se conectan a través de bus (comparten memoria)
  - Ejecución concurrente de CPUs y dispositivos, compitiendo por ciclos de memoria



# Operación de la Computadora

- Dispositivos I/O y CPU en ejecución concurrente
- Cada controlador se encarga de un dispositivo particular
- Cada controlador tiene un buffer local
- El CPU mueve datos de/a memoria principal a/de buffers locales
- I/O se realiza de un dispositivo al buffer local de un controlador
- El controlador le informa al CPU que ha terminado mediante una *interrupción*.

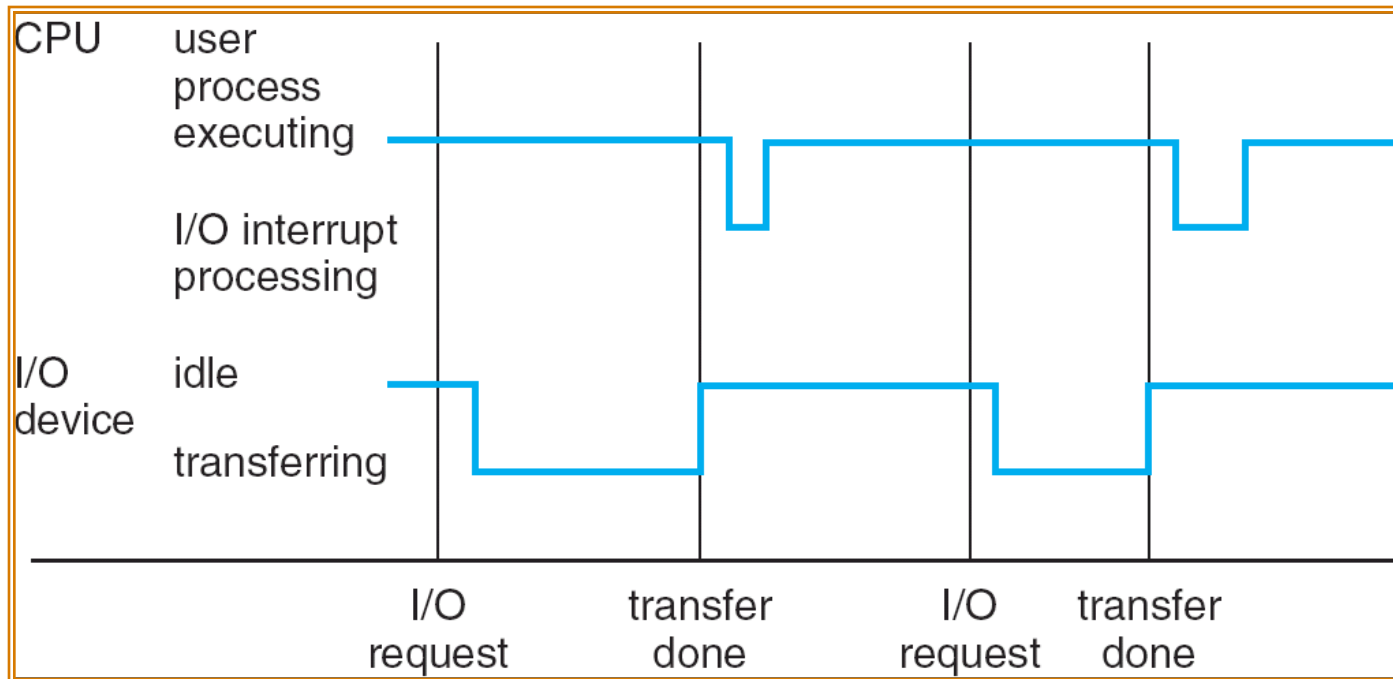
# Funciones de las Interrupciones

- Transferir control a la rutina de atención a la interrupción, a través del *vector de interrupción*, el cual contiene la dirección de la rutina de atención
- La arquitectura debe guardar la dirección de la instrucción interrumpida
- Las interrupciones *deshabilitan* mientras se procesa otra interrupción, para evitar *interrupciones perdidas*
- Un *trap* es una interrupción generada por software, causada por un error o requisición del usuario
- El SO es un sistema basado en interrupciones (interrupt driven)

# Manejo de Interrupciones

- El SO preserva el estado del CPU almacenado registros y el PC (program counter).
- Determina qué tipo de interrupción ha ocurrido:
  - *Polling (bussy waiting)*
  - *Systema vectorizado* de interrupciones
- Segmentos de código separados determinan qué acción se debe tomar para cada tipo de interrupción

# Diagrama de tiempo de Interrupciones



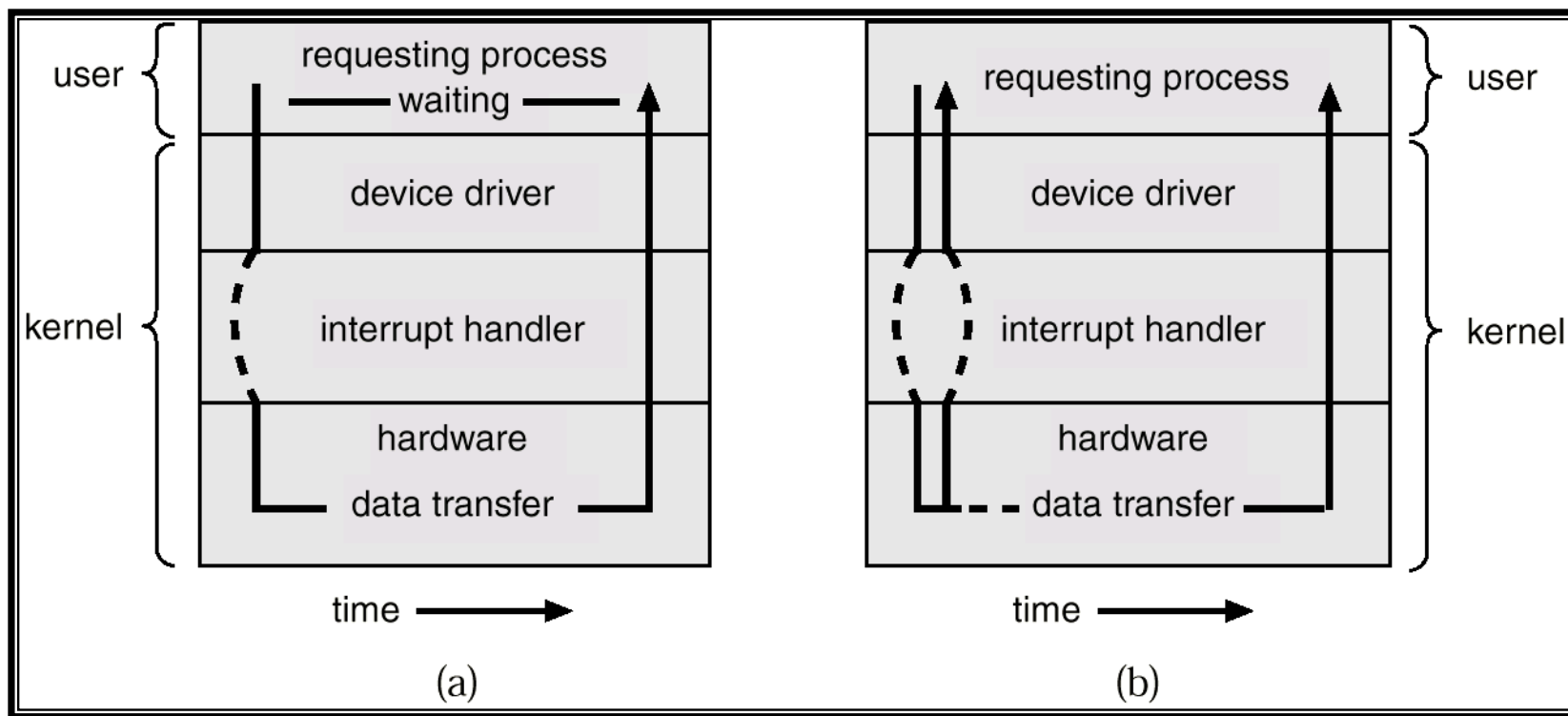
# Estructura de I/O

- Después de que la I/O comienza, el control no se regresa al programa sino hasta que el I/O termina
  - La instrucción `Wait` detiene al CPU hasta que hay una interrupción
  - `Wait loop` (esperar para acceder memoria)
  - A lo mas una requisición de I/O se atiende a la vez
- Después de que la I/O comienza, el control se regresa al programa sin esperar a que el I/O termine
  - *System call* – pide al SO que permita al usuario esperar a que termine el I/O
  - *Device-status table* – contiene una entrada para cada dispositivo de I/O indicando tipo, dirección y estado
  - El SO accesa la tabla de dispositivos para determinar el estado del dispositivo y modificarla par indicar que se encuentra en interrupción

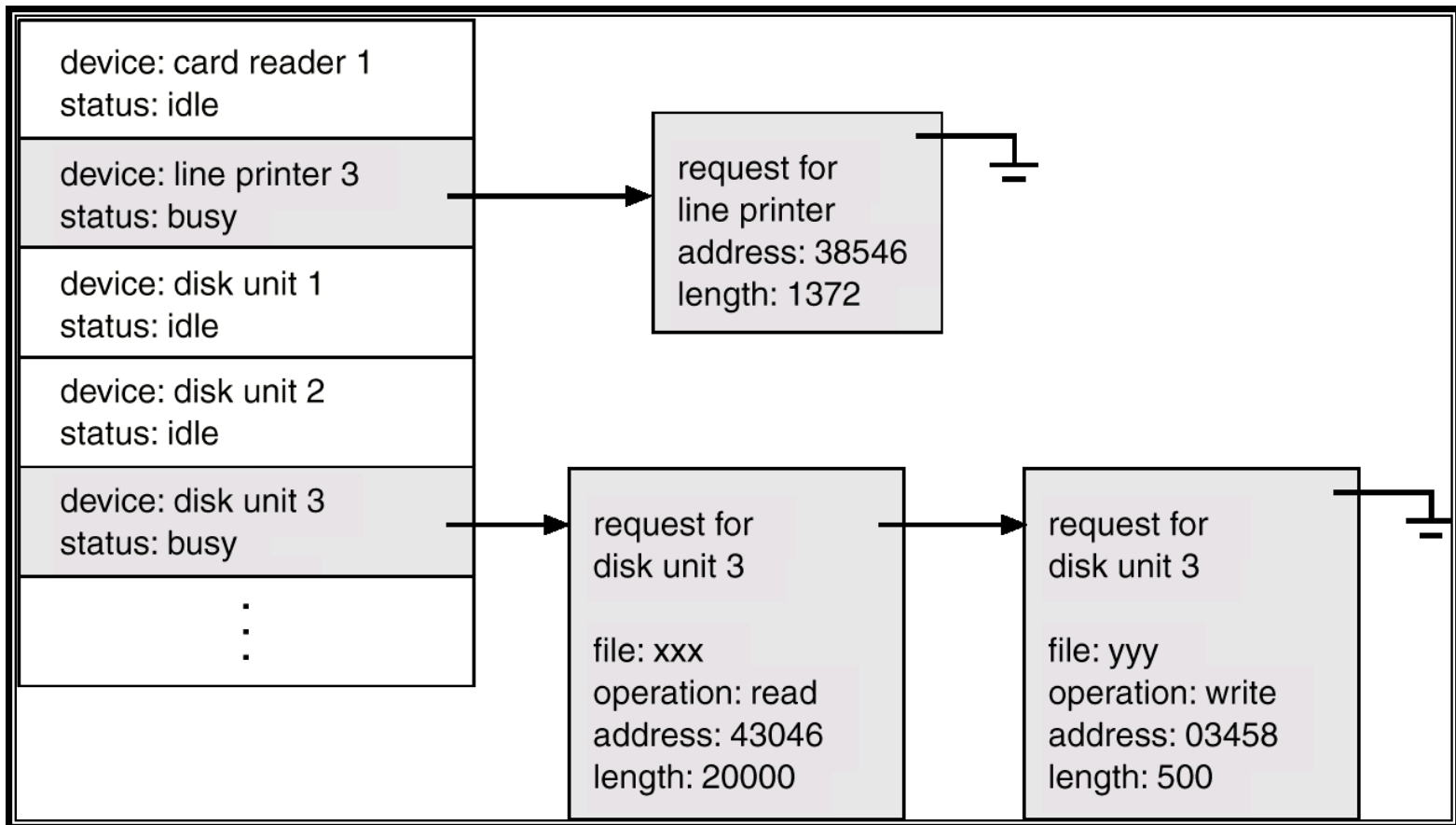
# Dos métodos de I/O

Síncrono

Asíncrono



# Device-Status Table



# Estructura DMA (Direct Memory Access)

- Usada para dispositivos de I/O de alta velocidad, que transmiten información a velocidades cerca de la de la memoria
- El controlador transfiere bloques de datos del buffer directamente a memoria principal sin intervención del CPU
- Se genera una interrupción por bloque, y no una por byte

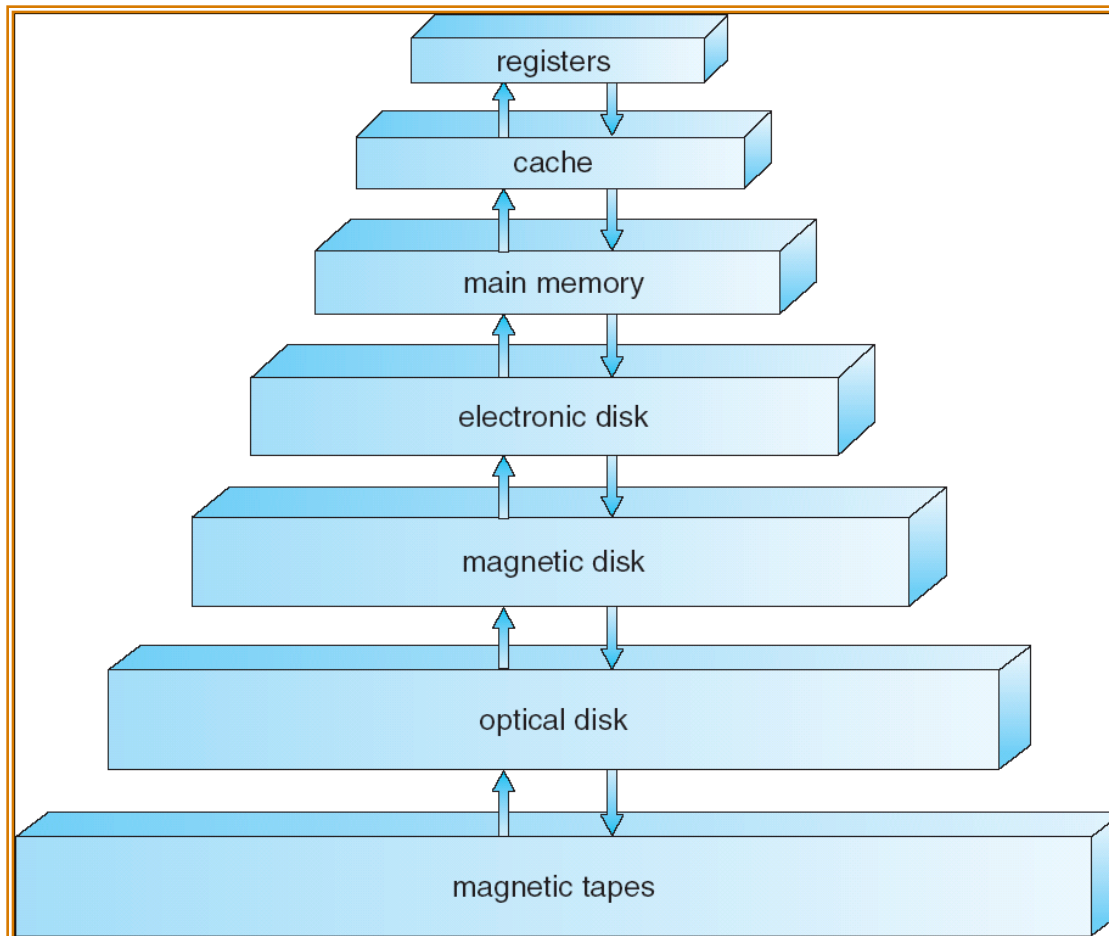
# Estructura de Almacenamiento

- Memoria principal – única memoria que el CPU accesa directamente
- Memoria secundaria – extensión de la memoria principal que proporciona almacenamiento masivo no volátil
- Discos magnéticos – plátos rígidos de metal o vidrio cubiertos de material magnético grabable
  - La superficie se divide (lógicamente) en *tracks*, divididos en *sectores*
  - El *controlador de disco* determina la interacción lógica entre el dispositivo y la computadora

# Almacenamiento

- Los sistemas de almacenamiento se organizan jerárquicamente
  - Velocidad
  - Costo
  - Volatilidad
- *Caching* – copia de información a un sistema de almacenamiento más rápido. La memoria principal puede ser vista como el *cache* de la memoria secundaria

# Jerarquía de Almacenamiento



# Cache

- Principio importante, realizado a varios niveles en una computadora (en hardware, SO, software)
- La información en uso se copia de un almacenamiento más lento a uno más rápido, temporalmente
- El medio más rápido (cache) se checa primero para determinar si la información se encuentra ahí
  - SI – la información se usa directamente del cache (rápido)
  - NO – Los datos se copian al cache y se usan de ahí
- El cache es más pequeño que la memoria principal
  - El manejo del cache es una decisión importante de diseño
  - Tamaño y políticas de reemplazo del Cache

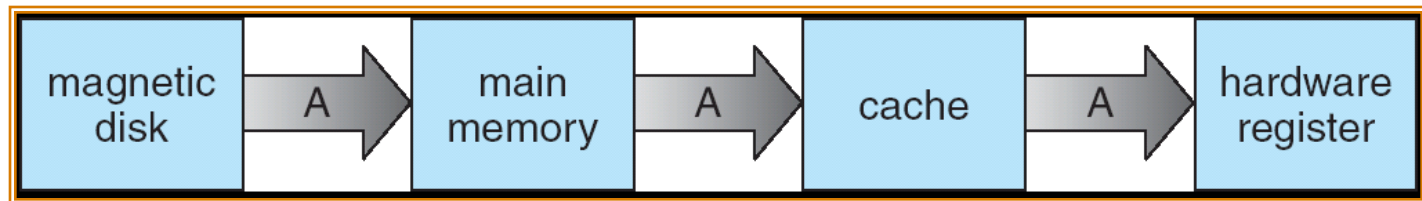
# Desempeño de los Niveles de Almacenamiento

- Los movimientos entre niveles de almacenamiento pueden ser explícitos o implícitos

Level	1	2	3	4
Name	registers	cache	main memory	disk storage
Typical size	< 1 KB	> 16 MB	> 16 GB	> 100 GB
Implementation technology	custom memory with multiple ports, CMOS	on-chip or off-chip CMOS SRAM	CMOS DRAM	magnetic disk
Access time (ns)	0.25 – 0.5	0.5 – 25	80 – 250	5,000.000
Bandwidth (MB/sec)	20,000 – 100,000	5000 – 10,000	1000 – 5000	20 – 150
Managed by	compiler	hardware	operating system	operating system
Backed by	cache	main memory	disk	CD or tape

# Movimiento de un Integer A de Disco a un Registro

- Se debe usar el valor más reciente, no importa donde se encuentre en la jerarquía de almacenamiento

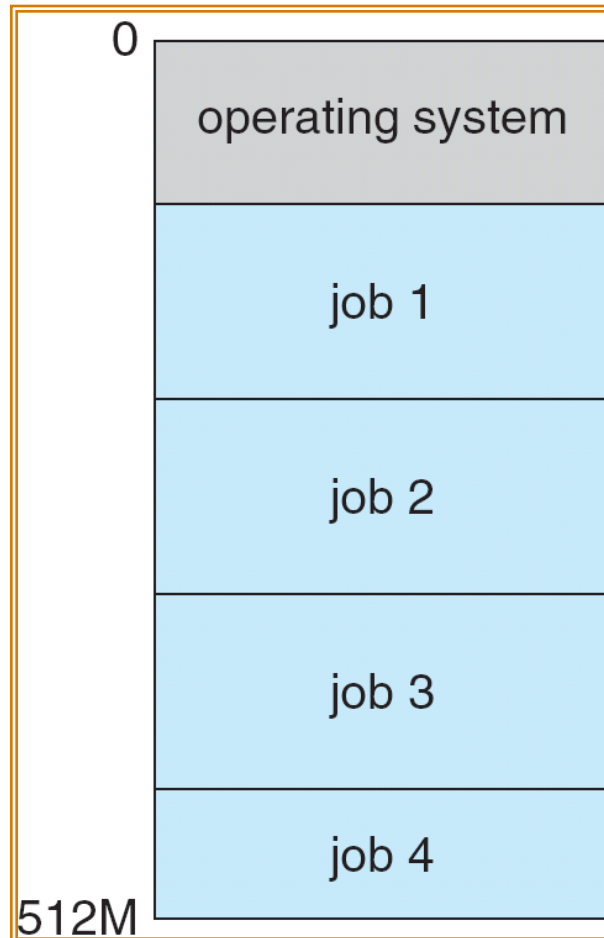


- Si se tienen multiprocesadores se debe proporcionar coherencia de cache en hardware (todos los CPUs tienen el valor más reciente del dato en su cache)
- La situación es aún más compleja en medios distribuidos
  - Pueden existir varias copias de un dato
  - La solución no se estudiará en este curso

# Estructura de un SO

- **Multiprogramming** = eficiencia
  - Un solo usuario no puede mantener al CPU y a los dispositivos I/O ocupados todo el tiempo
  - La multiprogramación organiza tareas (jobs) de manera que el CPU siempre tiene algo que hacer
  - Un subconjunto de esas tareas se mantiene en memoria
  - **job scheduling** = ¿qué tarea ejecutar?
  - Cuando el programa tiene que esperar (I/O, por ejemplo), el SO cambia a otro programa
  - **Timesharing (multitasking)** es una extensión lógica, donde el CPU cambia de jobs tan rápido que los usuarios pueden interactuar con cada programa mientras se ejecuta, creando el cómputo **interactivo**
  - El tiempo de respuesta debe ser  $< 1$  segundo
  - Cada usuario tiene al menos un programa en memoria y ejecutándose  $\Rightarrow$  **proceso**
  - Si varios jobs se encuentran listos para ejecución al mismo tiempo  $\Rightarrow$  **CPU scheduling**
  - Si los procesos no caben en memoria, se meten y sacan mediante **swapping**
  - La memoria **virtual** permite la ejecución de procesos que no se encuentran completamente en memoria

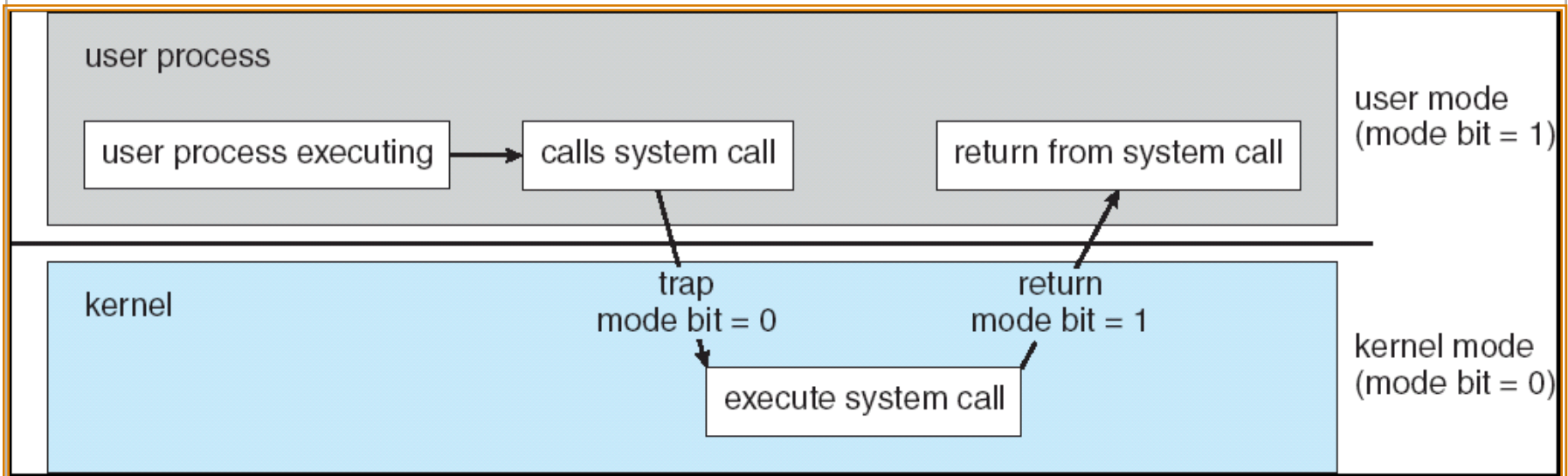
# Mapa de Memoria para un Sistema Multiprogramado



# Operaciones del SO

- Interrupciones ocasionadas por hardware
- Errores de software o requisiciones crean **excepciones** o **traps**
  - División por cero, requerir un servicio del SO
- Otros problemas incluyen ciclos infinitos, o procesos que modifican la memoria de otros o del SO
- Modo **Dual** de operación le permite al SO protegerse y proteger los otros componentes del sistema
  - **User mode** y **kernel mode**
  - El hardware proporciona un **Mode bit**
    - Distingue cuando el sistema ejecuta código del usuario o del kernel
    - Algunas instrucciones son **privilegiadas**, solo se ejecutan en kernel mode
    - Las llamadas al SO cambia el modo a kernel mode, el return lo regresa a user mode

# Transición de User a Kernel Mode



# Transición de User a Kernel Mode

- Un timer previene ciclos infinitos / procesos que consumen muchos recursos (hogs)
  - Pon una interrupción después de cierto tiempo
  - OS decrementa un contador
  - Si contador=0 se genera una interrupción
  - Se prepara antes de despachar un proceso, para retomar el control o terminar un programa que se pasa de tiempo

# Administración de Procesos

- Proceso = programa en ejecución. Unidad de trabajo del sistema.
- Programa = *entidad pasiva*, proceso = *entidad activa*.
- Los procesos requieren recursos para realizar sus tareas
  - CPU, memoria, I/O, archivos
  - Datos de inicialización
- Para la terminación de procesos el SO reclama recursos reusables
- Los procesos de un solo thread tienen un PC (**program counter**)
  - Un proceso ejecuta instrucciones secuencialmente
- Los procesos multi-threaded tienen un PC por thread
- Los sistemas tienen muchos procesos (de usuario y sistema) ejecutándose concurrentemente en uno o más CPUs
  - Concurrencia = multiplexar los CPUs entre procesos / threads

# Administración de Procesos

El SO es responsable de:

- Crear y destruir procesos (usuario y sistema)
- Suspende y resume procesos
- Proporcionar mecanismos de sincronización
- Proporcionar mecanismos de comunicación
- Proporcionar mecanismos para tratar con deadlocks

# Administración de Memoria

- Todos los datos están en memoria antes y después de procesarlos
- Todas las instrucciones deben estar en memoria para ser ejecutadas
- El administrador de memoria determina qué reside en memoria para
  - Optimizar el uso de CPU y la respuesta de la computadora
- Actividades de AM
  - Registrar que partes de la memoria están siendo usadas y por quien
  - Decidir qué procesos (o partes de) y datos ingresar y sacar de memoria
  - Alojarse y desalojar memoria de acuerdo a las necesidades

# Administración de Almacenamiento

- Vista uniforme, lógica de los medios de almacenamiento
  - Abstrae las propiedades físicas del medio a la unidad lógica - **archivo**
  - Cada medio es controlado por su propio dispositivo (i.e., disco, cinta, etc.)
    - Propiedades de los medios: velocidad de acceso, capacidad, velocidad de transferencia, método de acceso (secuencial o aleatorio)
- Administración de archivos
  - Archivos y directorios
  - Quién puede acceder qué
  - Actividades del SO
    - Crear y borrar archivos y directorios
    - Primitivas para manipular archivos y directorios
    - Mapear archivos al medio de almacenamiento
    - Respaldo de archivos a medios no volátiles

# Administración de Almacenamiento Masivo

- Los discos almacenan datos que no caben en memoria o que deben ser almacenados “permanentemente”
- La velocidad de la computadora depende del sistema de discos y sus algoritmos
- Actividades del SO
  - Manejo de espacio libre
  - Asignar espacio
  - Despacho de disco

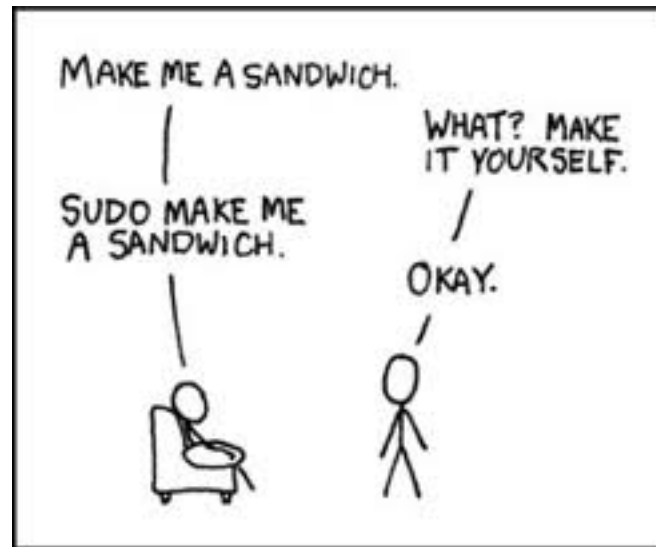
# Subsistema I/O

- Ocultarle al usuario los particulares de los dispositivos
- El subsistema de I/O:
  - Administra la memoria de I/O.
  - Buffering – almacenamiento temporal de datos mientras se transfieren
  - Caching – almacenar partes de los datos en un medio más rápido
  - Spooling – traslape de la salida de un job con la entrada del otro
  - Interfaz de drivers

# Protección y Seguridad

- **Protección** – mecanismo de control de acceso a los recursos
- **Seguridad** – defensa contra ataques externos
  - DOS (denial-of-service), worms, virus, identity theft, etc.
- SO distingue entre usuarios, para determinar privilegios (quien puede hacer que)
  - **User IDs** – incluyen nombre y número de usuario
  - User ID asociado con cada archivo, procesos, etc, para control de acceso
  - Identificador de grupo (**group ID**) permite que un grupo de usuarios tengan privilegios sobre un grupo de objetos
  - **Privilege escalation** le permite al usuario cambiar a un ID con mas derechos (si se le permite)

# Protección y Seguridad



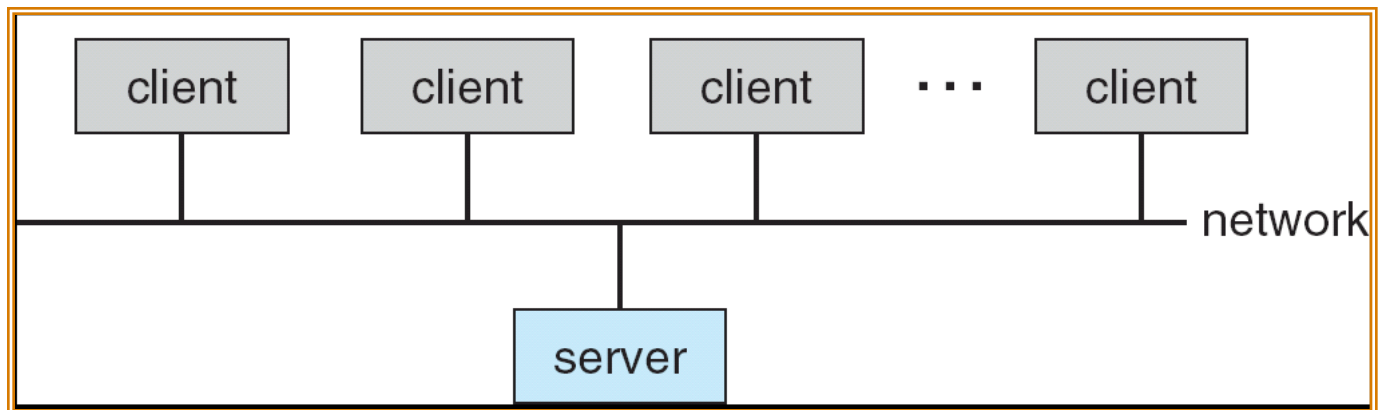
# Medios Ambientes de Cómputo

- Computadora tradicional
  - Oficina
    - PCs conectados a una red, terminales conectadas a un mainframe, minicomputadoras con sistemas batch y/o timesharing
    - Recientemente se tienen sistemas de red con acceso remoto a recursos compartidos
  - Redes caseras
    - Antes computadora aislada, luego modems ...
    - Ahora en red, con firewall, etc.

# Medios Ambientes de Cómputo (Cont.)

## ■ Esquema Cliente-Servidor

- Dumb terminals substituidas por smart PCs
- Muchos sistemas son **servidores**, respondiendo a requisiciones generadas por **clientes**
  - ▶ **Compute-server** proporciona una interfaz al cliente para solicitar servicios (v.g. database)
  - ▶ **File-server** proporciona una interfaz para almacenar archivos



# Cómputo Peer-to-Peer

- Otro modelo de sistemas distribuidos
- P2P no distingue entre cliente y servidor
  - Todos los nodos son peers
  - Actuando como cliente, servidor, o ambos
  - Cada nodo debe anotarse en una red P2P
    - Registra sus servicios en un servidor central de búsqueda
    - Solicita servicios por medio de *broadcast* y responde requisiciones de servicio mediante un *discovery protocol*
  - Ejemplos: *Napster*, *Gnutella*, etc.

# Cómputo Basado en Web

- La Web es ubicua
- Las PCs son los dispositivos comunes
- Mas y mas dispositivos tienen acceso a la red
- Ahora se cuenta con nuevos dispositivos para manejar el tráfico web entre servidores similares:  
**load balancers**
- Los SOs como Windows 95, típicamente clientes, han cambiado a Linux y Windows XP, los cuales pueden ser clientes o servidores