



---

# Capítulo 1: Introducción



# Capítulo 1: Introducción

---

- Historia de los sistemas operativos
- Línea de tiempo de la familia Unix
- Línea de tiempo de la familia windows
- Organización de computadoras

# Objetivo

---

Entender la necesidad de un S.O.

## 1.2 History of Operating Systems

- It all started with computer hardware in about 1940s.



ENIAC 1943

## 1.2 History of Operating Systems

- ENIAC (Electronic Numerical Integrator and Computer), at the U.S. Army's Aberdeen Proving Ground in Maryland.
  - built in the 1940s,
  - weighed 30 tons,
  - Was 3 ms high, 1 m deep, and 33 ms long
  - contained over 18,000 vacuum tubes that were cooled by 80 air blowers.

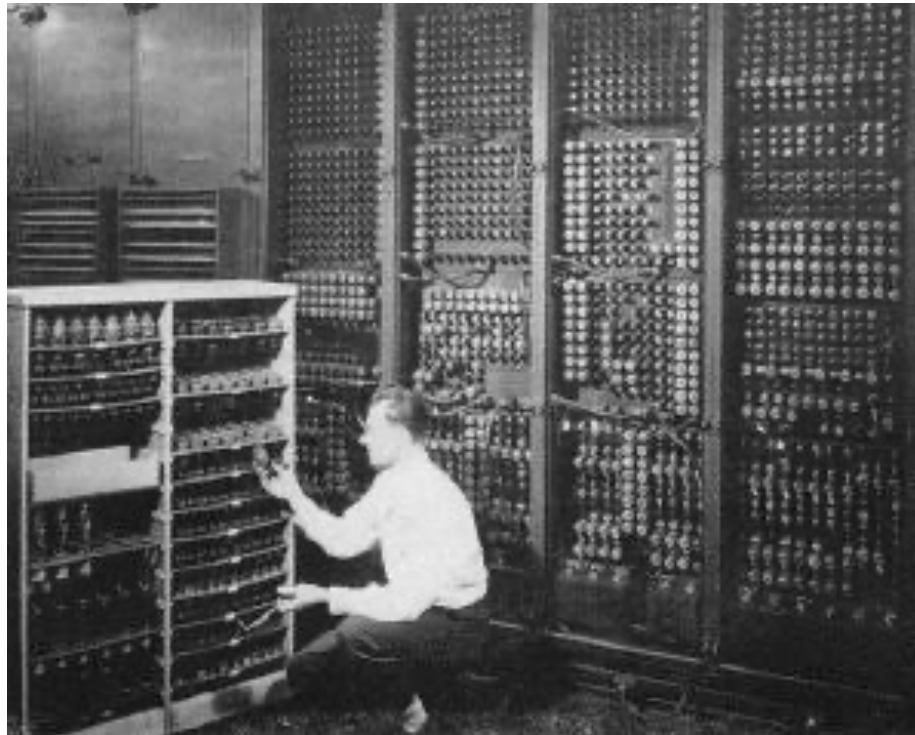
## 1.2 History of Operating Systems

- Computers were using vacuum tube technology



ENIAC's vacuum tubes

## 1.2 History of Operating Systems

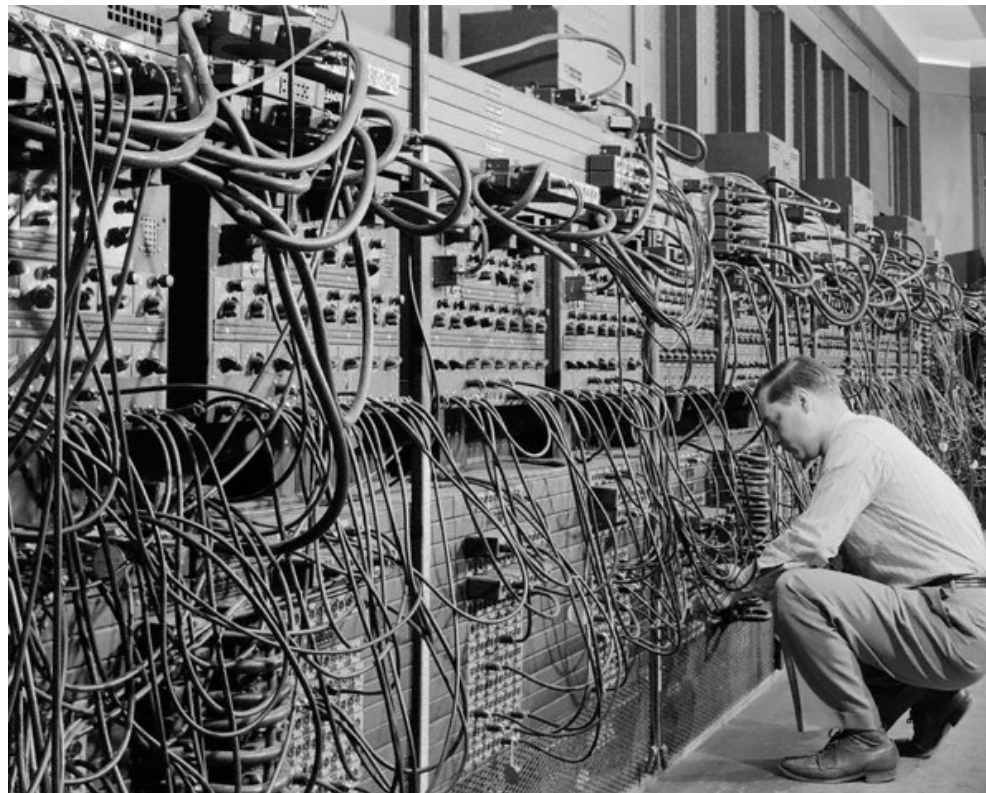


ENIAC's backside



## 1.2 History of Operating Systems

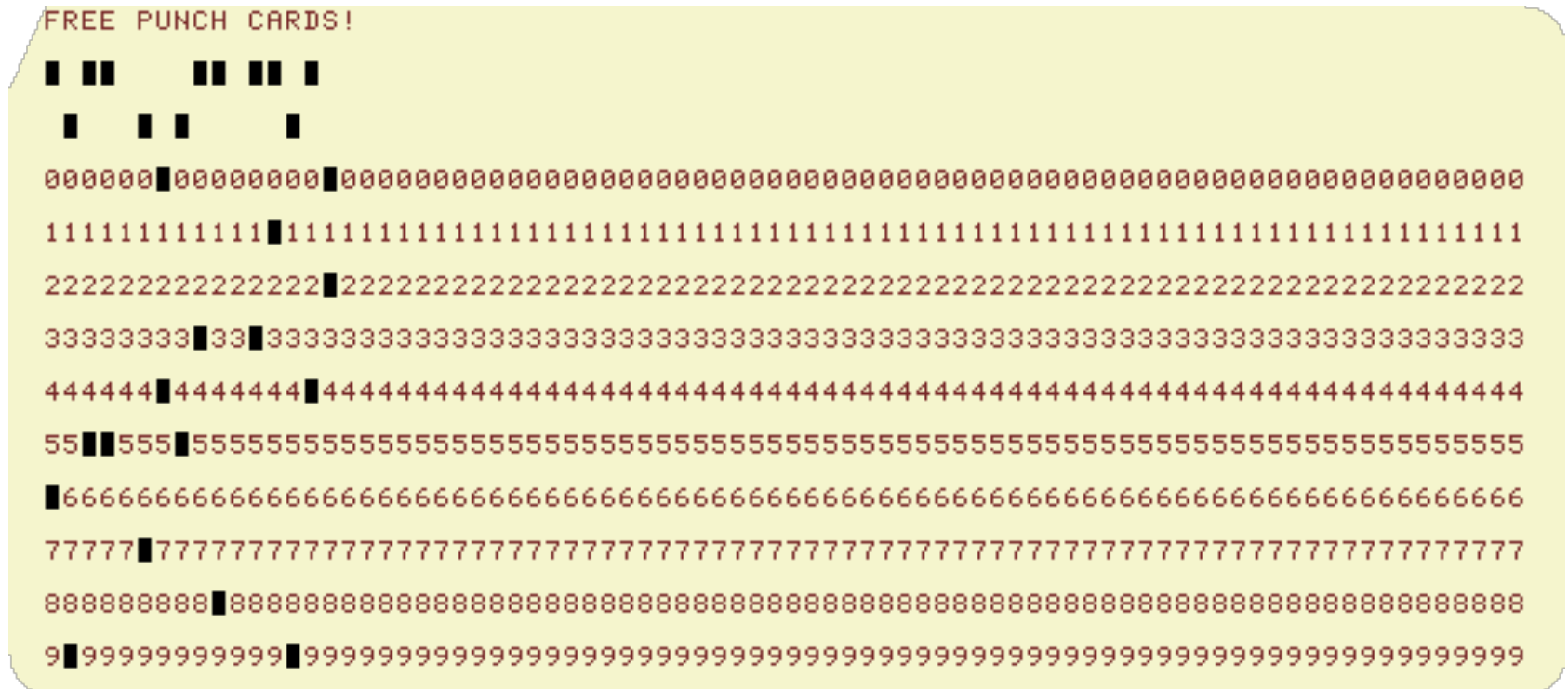
Programs were loaded into memory manually using switches, punched cards, or paper tapes.



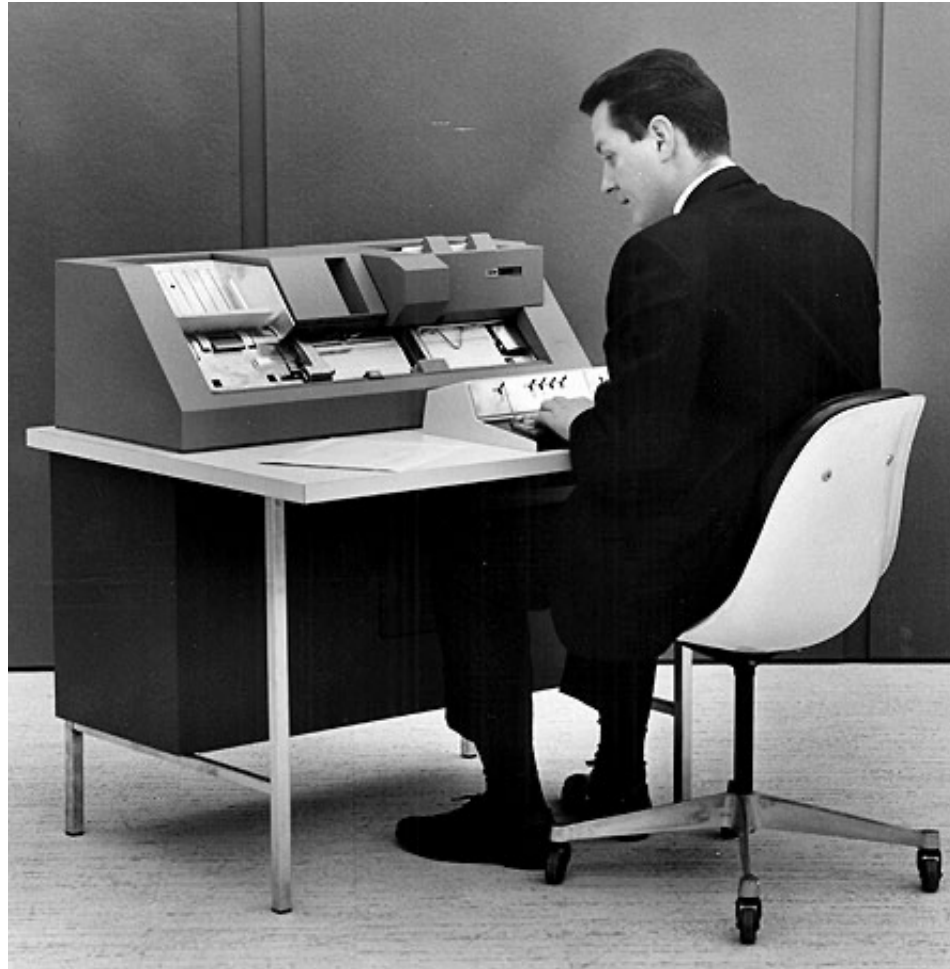
ENIAC : coding by cable connections

## 1.2 History of Operating Systems

# punch card



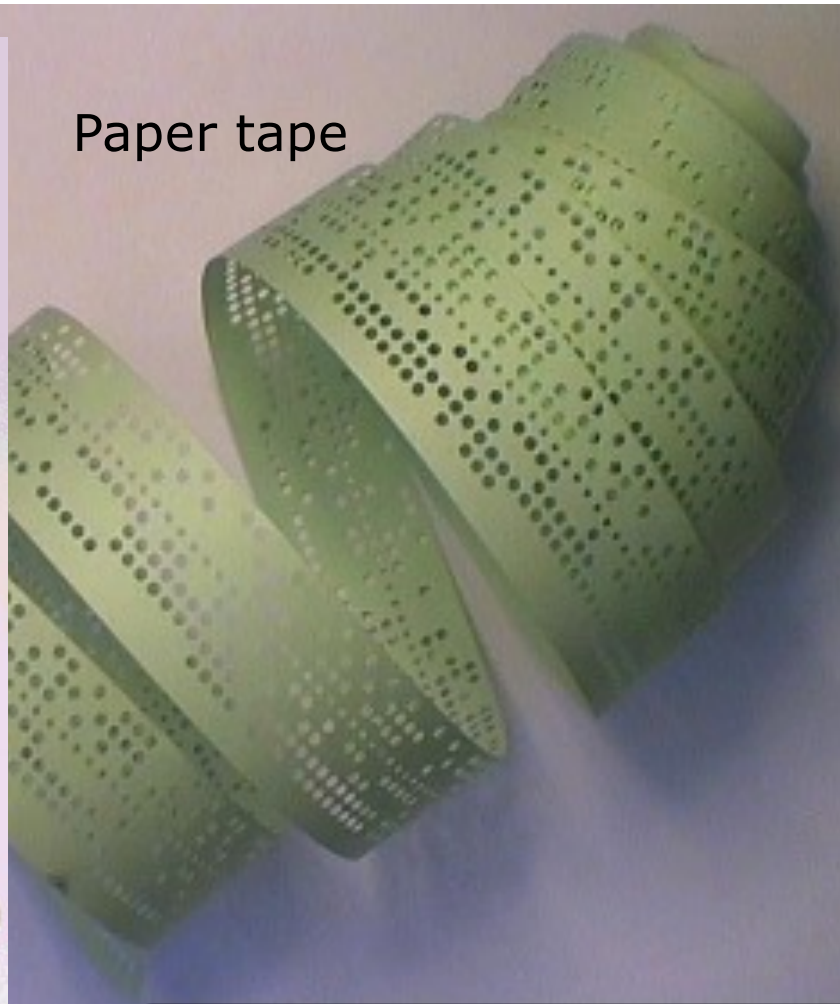
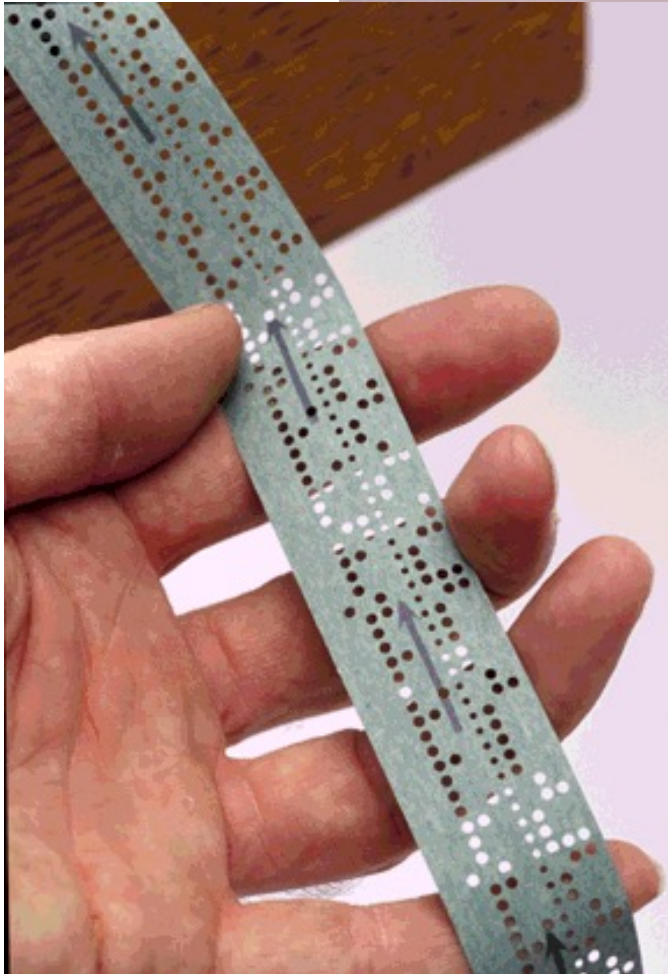
## 1.2 History of Operating Systems



**FREE PUNCH CARDS!**

[illegible]

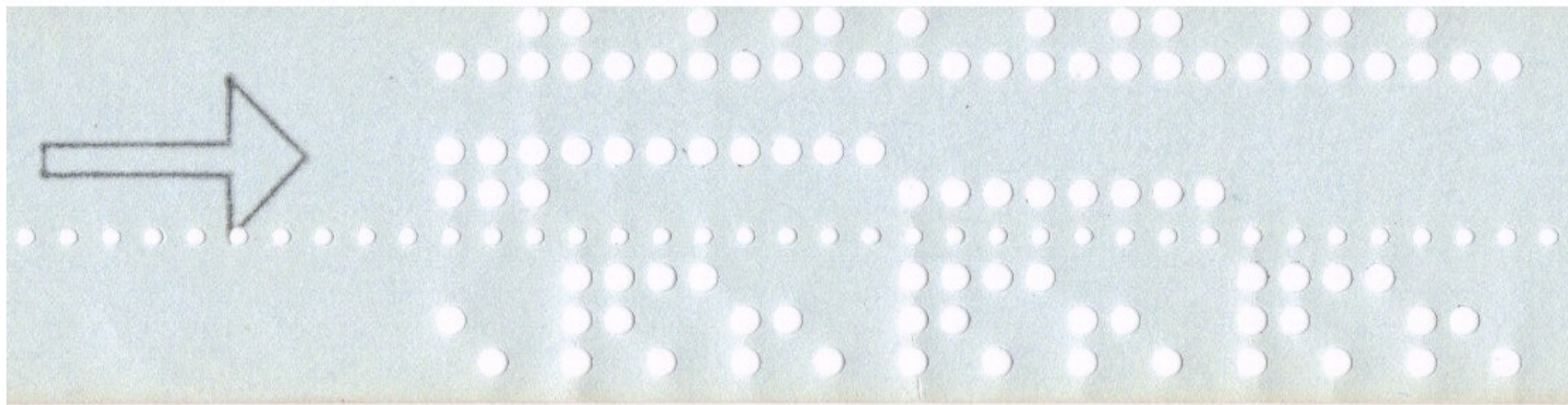
## 1.2 History of Operating Systems





# 1.2 History of Operating Systems

Punched Paper Tape 25.4 mm wide. Ascii 7-bit character code. Even Parity.



Parity (Even)  
Bit 7  
Bit 6  
Bit 5  
Bit 4  
Sprocket Holes  
Bit 3  
Bit 2  
Bit 1

Z Y X W V U T S R Q P O N M L K J I H G F E D C B A

## 1.2 History of Operating Systems

- As time went on, card readers, printers, and magnetic tape units were developed as additional hardware elements.
- Assemblers, loaders and simple utility libraries were developed as software tools.
- ESTOS FUERON LOS EMBRIONES DE LOS S.O.'S

## 1.2 History of Operating Systems

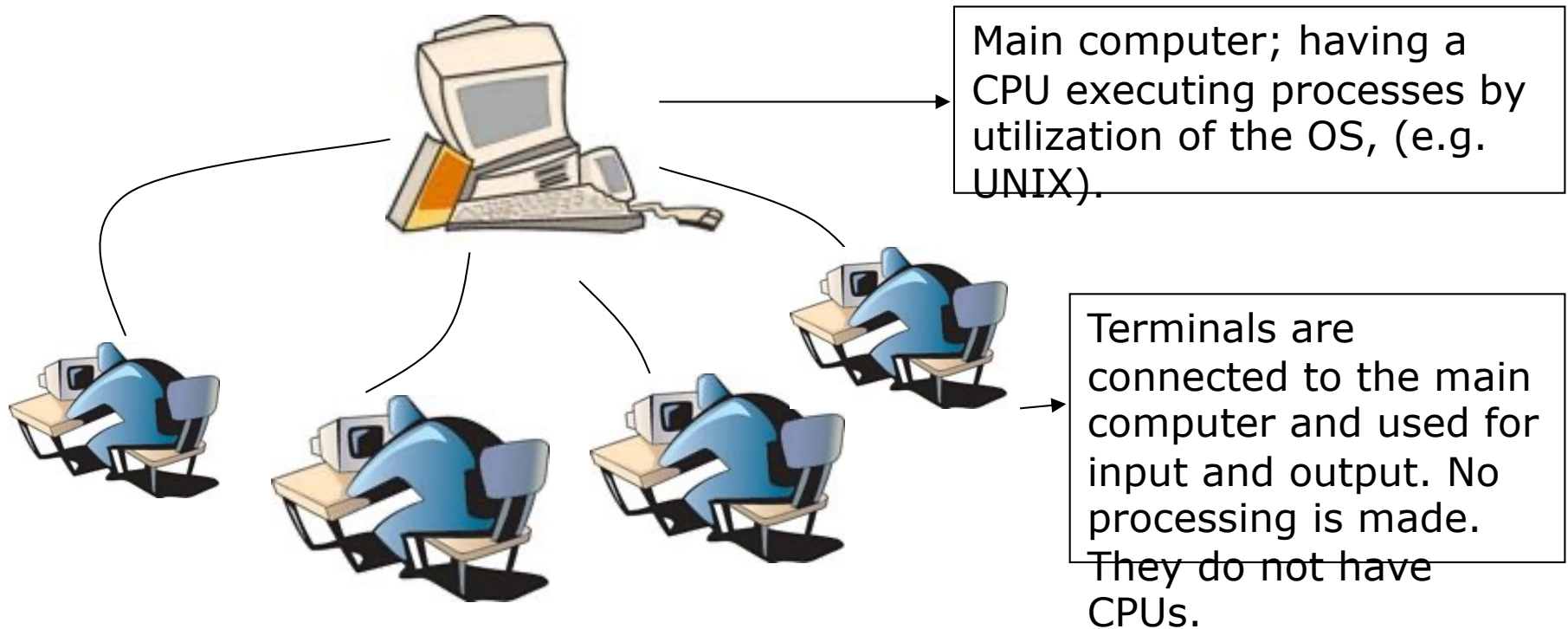
- Finally, the idea of **multiprogramming** came.
- Multiprogramming means sharing of resources between more than one processes.
- By multiprogramming the CPU time is not wasted, because, while one process moves on some I/O work, the OS picks another process to execute until the current one passes to I/O operation.

## 1.2 History of Operating Systems

- With the development of interactive computation in 1970s, **time-sharing systems** emerged.
- In these systems, multiple users have *terminals* (not computers) connected to a *main computer* and execute her task in the main computer.



## 1.2 History of Operating Systems



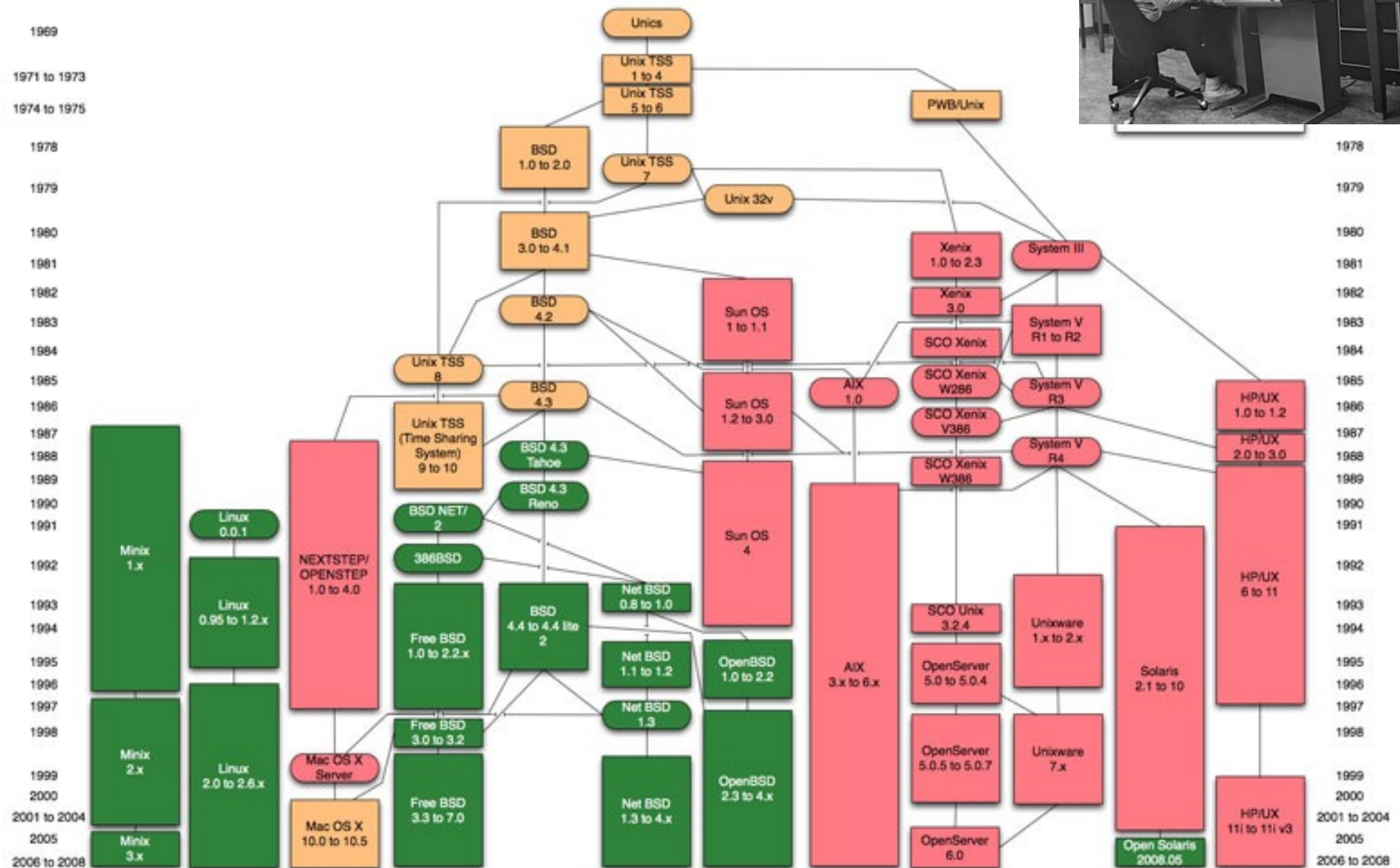
## 1.2 History of Operating Systems

- Another computer system is the **multiprocessor system** having multiple processors sharing memory and peripheral devices.
- With this configuration, they have greater computing power and higher reliability.

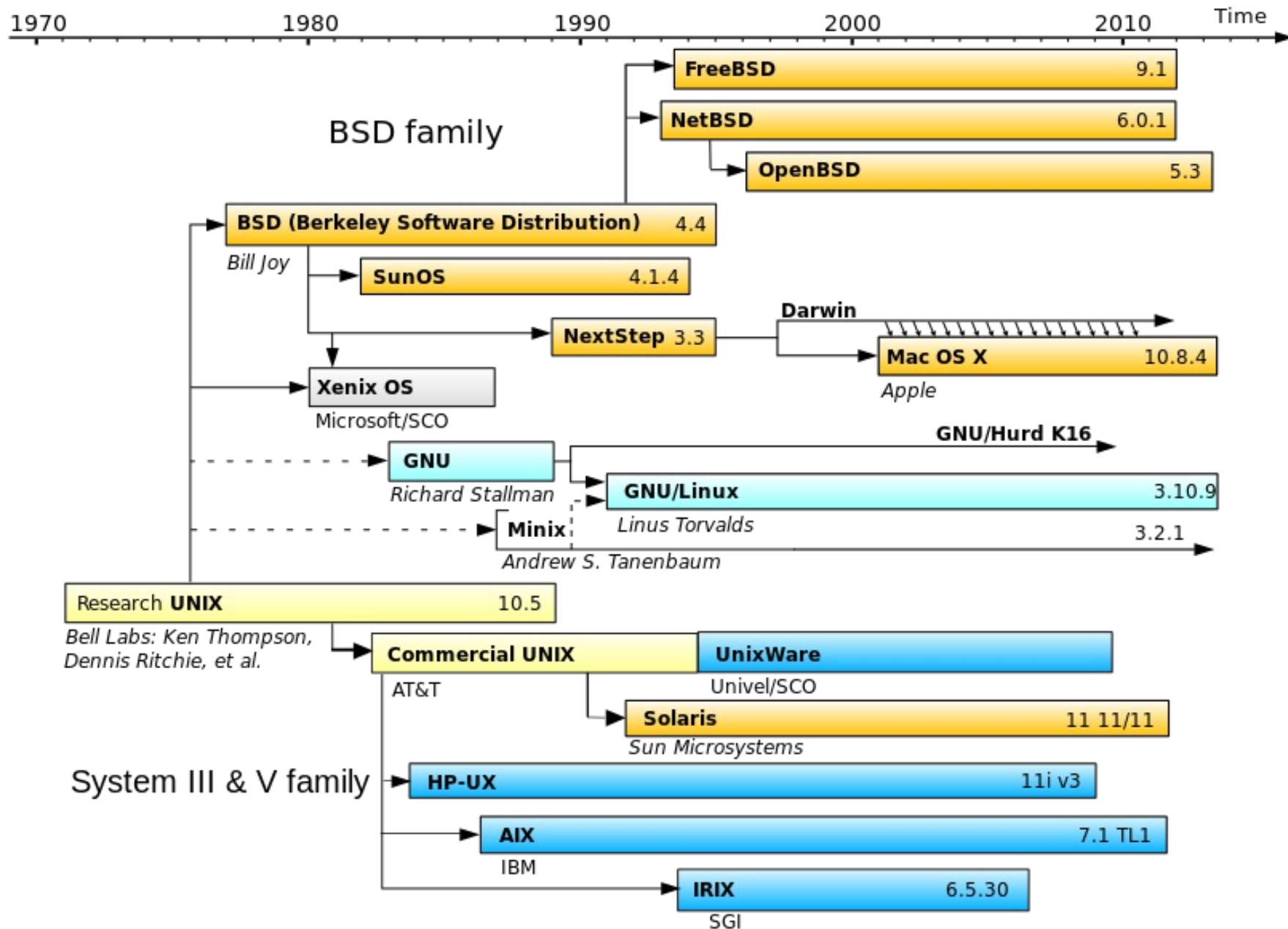
## 1.2 History of Operating Systems

- Multiprocessor systems are classified into two as tightly-coupled and loosely-coupled (distributed).
- In the tightly-coupled one, each processor is assigned a specific duty but processors work in close association, possibly sharing the same memory.
- In the loosely coupled one, each processor has its own memory and copy of the OS.

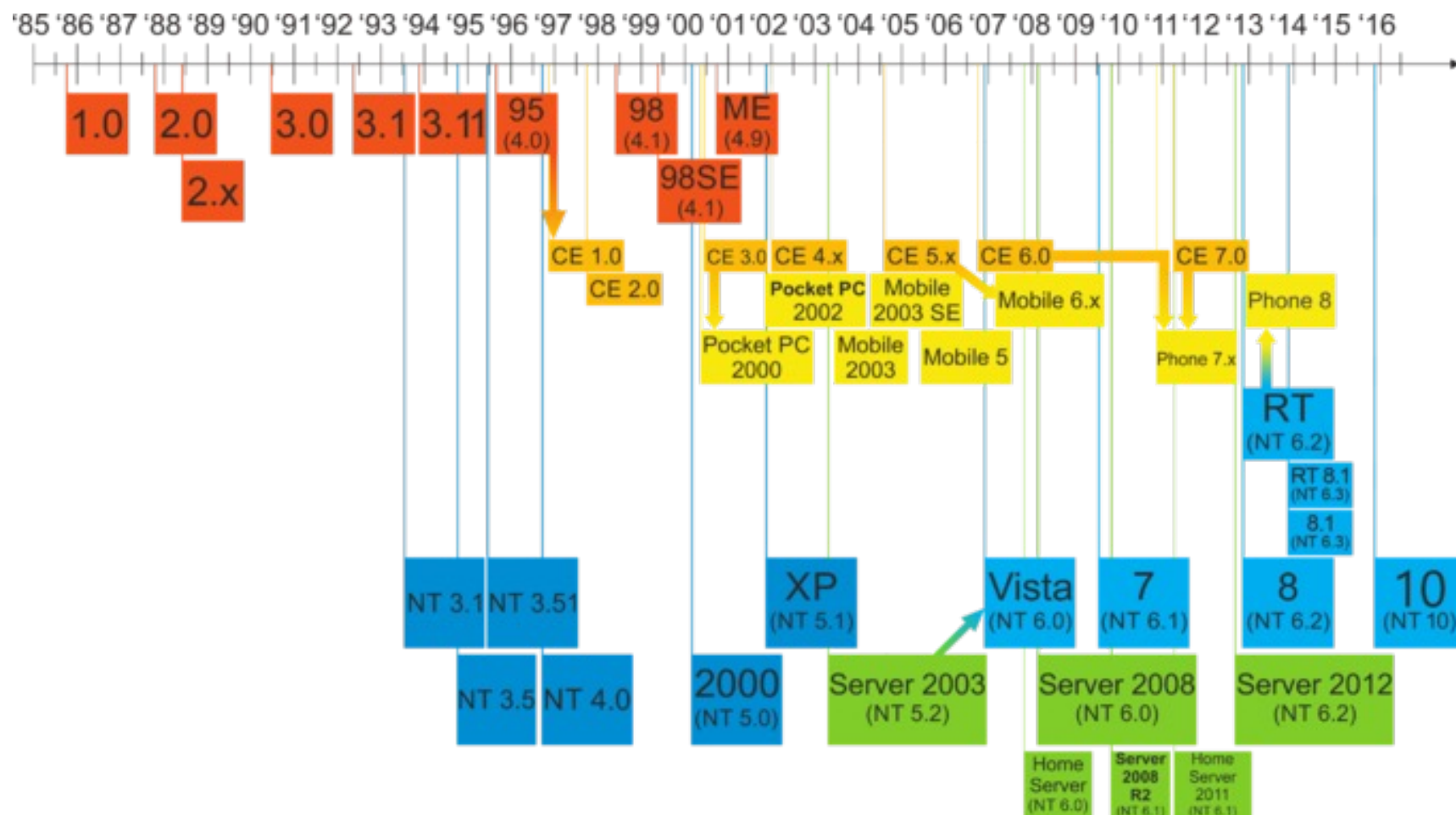
# Árbol genealógico de Unix



# Árbol genealógico de Unix



# Árbol genealógico de Windows



# Línea de tiempo de S.O.'s en wikipedia

---

[https://en.wikipedia.org/wiki/Timeline\\_of\\_operating\\_systems](https://en.wikipedia.org/wiki/Timeline_of_operating_systems)

# Hoy en día un S.O. ...

---

- Carga programas y los ejecuta
- Les da memoria.
- Les asigna tiempo de procesador
- Les permite comunicarse
- Los elimina
- Controla los dispositivos
- Provee E/S de alto nivel (archivos, sockets)
- Ofrece recursos gráficos (ventanas)
- Administra energía
- .....
- Tareas administrativas, cuyo propósito final es permitir que haya aplicaciones.



# Objetivos

---

- Mostrar el panorama de los componentes más importantes de los sistemas operativos
- Dar principios elementales de organización de computadoras

# ¿Qué es un sistema operativo?

---

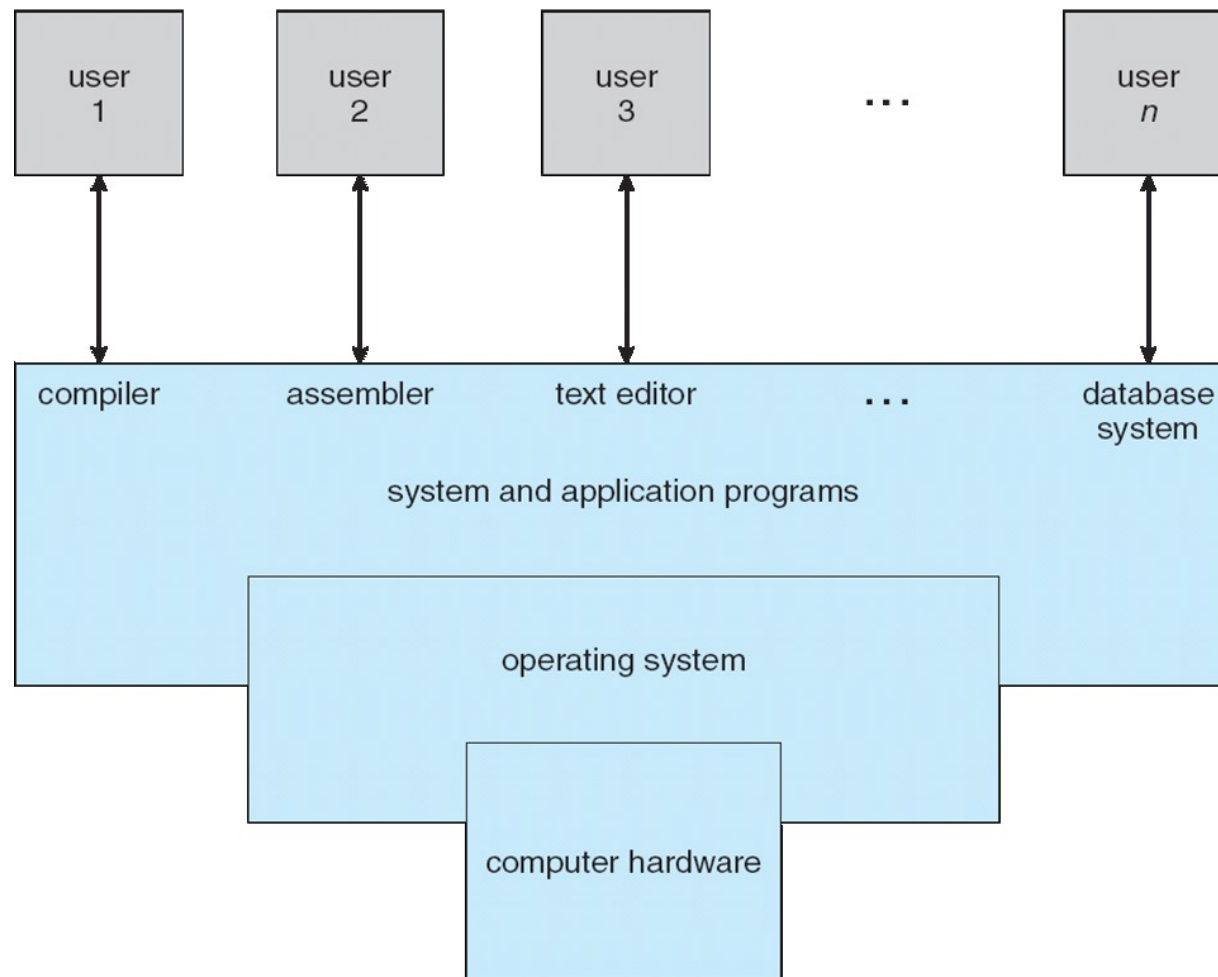
- Un *programa* que actúa como intermediario entre un usuario/proceso y el hardware
- Metas de un sistema operativo:
  - Ejecutar programas de usuario
  - Hacer el uso del sistema más cómodo
  - Usar el hardware de manera eficiente
  - Maximizar el uso del CPU (?)
  - Maximizar responsabilidad
  - Maximizar tiempo de batería

# Estructura de computadoras

---

- Un sistema de cómputo se puede dividir en cuatro componentes:
  - Hardware – ofrece recursos de cómputo elementales
    - ▶ CPU, memoria, Dispositivos de E/S
  - Sistema Operativo
    - ▶ Controla y coordina el uso de hardware entre varias aplicaciones y usuarios
  - Programas de aplicación – Definen la manera en que los recursos de cómputo son usados para resolver los problemas de los usuarios
    - ▶ Procesadores de texto, compiladores, navegadores web, videojuegos
  - Usuarios
    - ▶ Gente, máquinas, otras computadoras

# Cuatro componentes de un sistema de cómputo



# Definición de Sistema Operativo

---

- Un SO es un **asignador de recursos**
  - Administra todos los recursos
  - Decide entre solicitudes en conflicto para un uso justo y eficiente de los recursos
  
- Un SO es un **programa de control**
  - Controla la ejecución de programas para prevenir errores y un uso impropio de la computadora

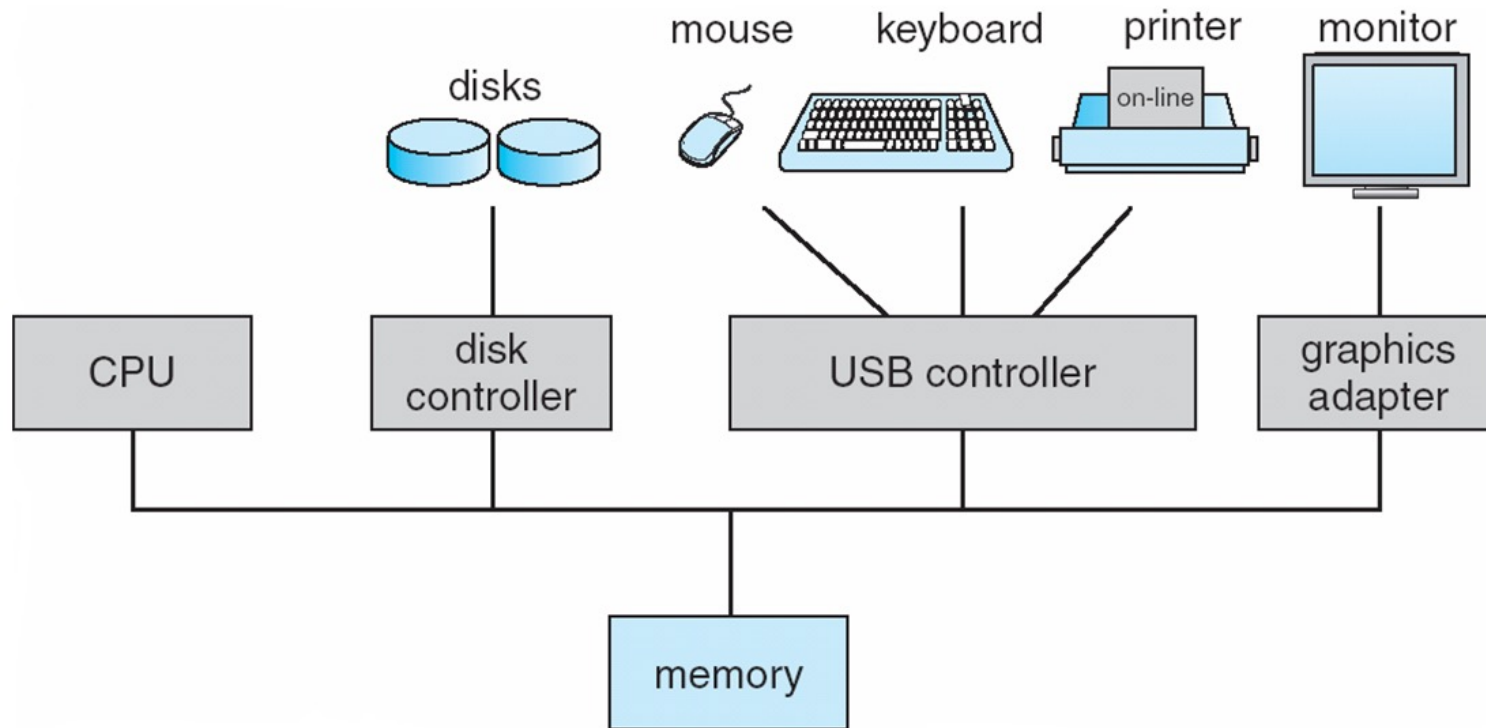
# Definición de Sistema Operativo (Cont.)

---

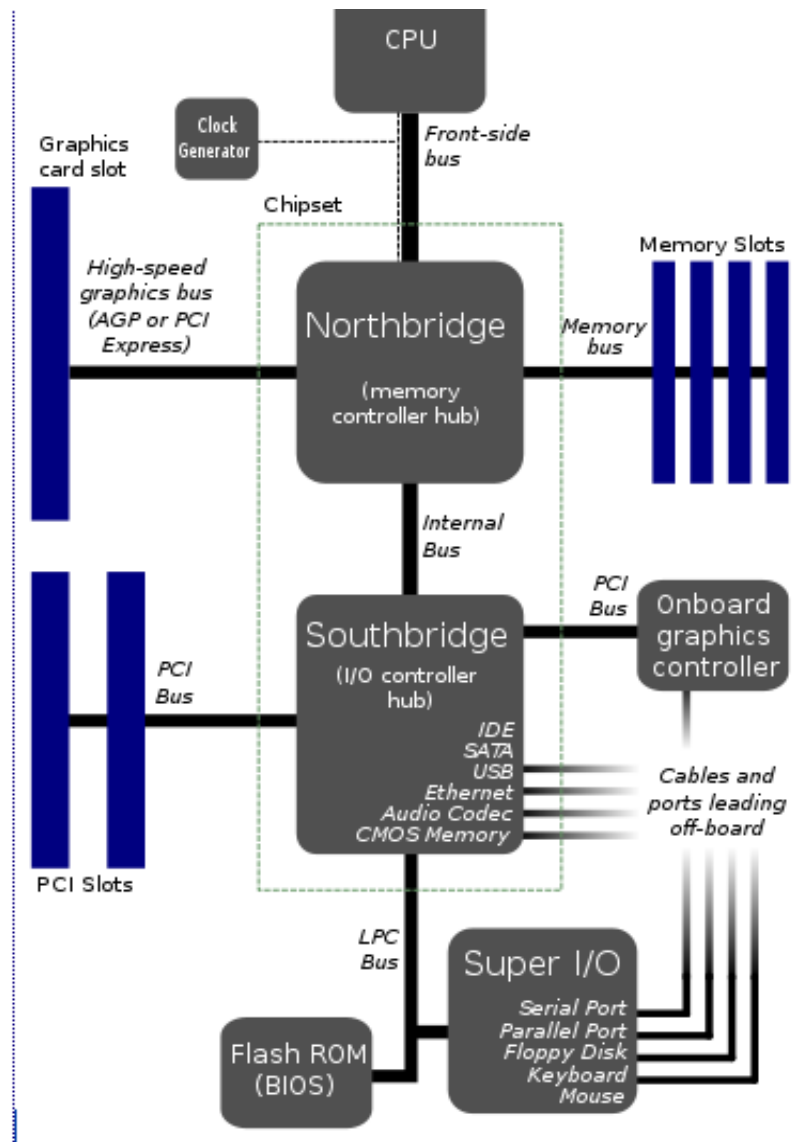
- No hay definición aceptada universalmente
- “Todo lo que un vendedor incluye cuando ordenas un sistema operativo” es una primera aproximación
  - Pero cada vez se incluyen más extras
- “El programa que corre todo el tiempo” es el **kernel**. Todo lo demás (ejecutable) es un programa del sistema (viene con el sistema operativo) o un programa de aplicación.
  - “Un sistema operativo es como un gobierno”

# Organización de computadoras

- Operación de un sistema de cómputo
  - Uno o más CPUs y controladores de dispositivos se conectan a un bus común brindando acceso a una memoria compartida
  - Ejecución concurrente de CPUs y dispositivos compitiendo por ciclos de la memoria



# Organización de componentes en una PC





# Operación de Sistemas de Cómputo

---

- Los dispositivos de E/S y los CPUs se pueden ejecutar simultáneamente
- Cada controlador de dispositivo está a cargo de un tipo de dispositivo particular
- Cada controlador de dispositivo tiene un buffer local
- El CPU mueve datos desde/hacia memoria principal hacia/desde buffers locales
- La E/S es del dispositivo al buffer local del controlador
- El controlador de dispositivo informa al CPU que ha terminado su operación causando una *interrupción*

# Funciones comunes de las interrupciones

---

- Generalmente transfieren el control a la **rutina de servicio de interrupción**.
- El sistema de interrupciones debe guardar la dirección de la instrucción interrumpida.
- Las demás interrupciones son deshabilitadas mientras otra interrupción está siendo procesada para prevenir una pérdida de interrupción.
- Una **trampa** es una interrupción generada por software causada por un error o una solicitud.
- Una **llamada al sistema** es una *trampa* que es activada por un proceso para, para solicitar un servicio al S.O.
- **Un sistema de cómputo es dirigido por interrupciones.**

# Arranque de la computadora

---

- **El programa de arranque** es cargado en cada inicio o reinicio
  - Típicamente almacenado en ROM o EPROM, generalmente llamado firmware
  - Inicializa todos los aspectos del sistema
  - Carga el kernel e inicia su ejecución

# Llamadas al sistema

---

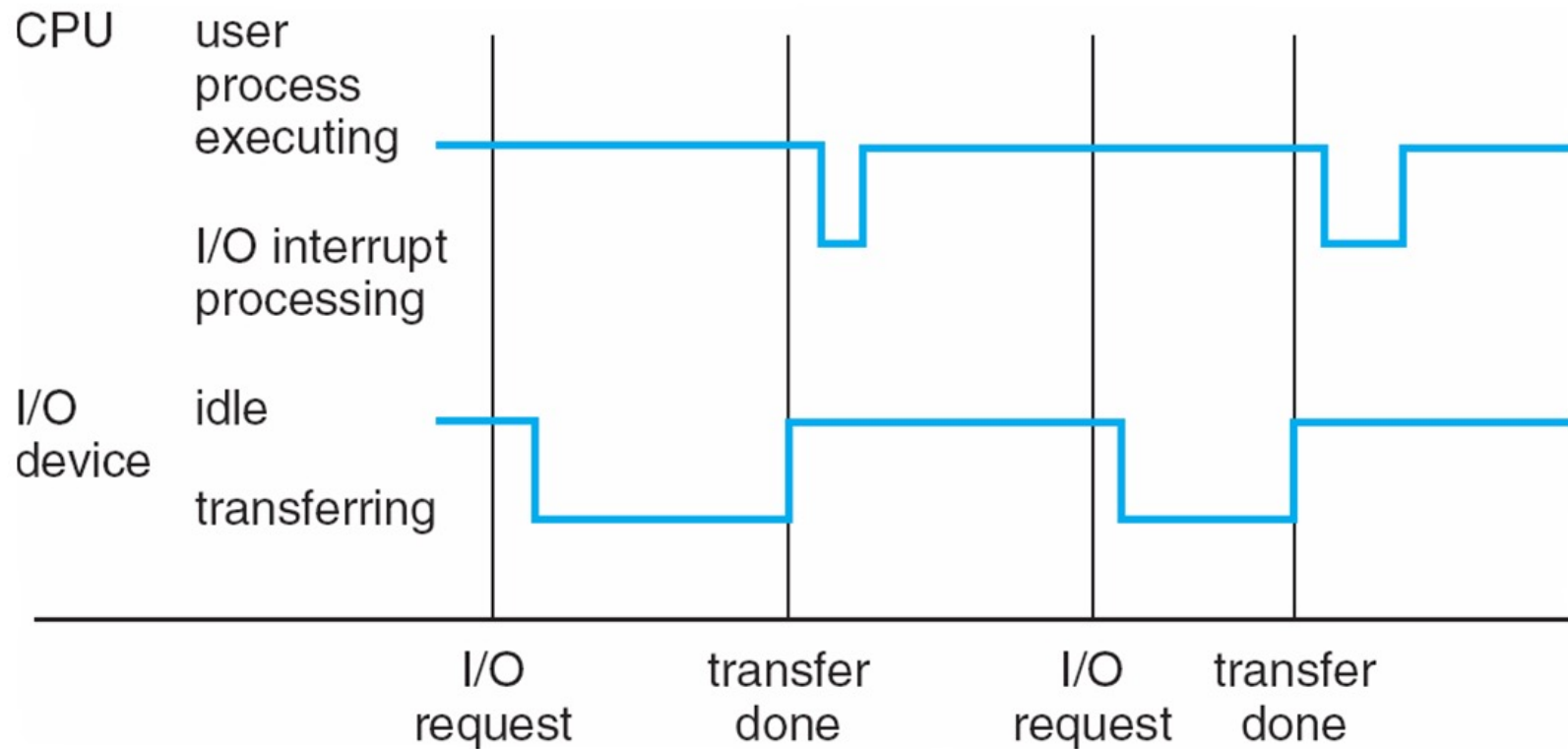
- Son el principal mecanismo para que un proceso solicite un servicio al sistema operativo.
  - Se implementan mediante interrupciones de software, que provocan que el CPU comience a ejecutar la rutina del kernel adecuada.

# Manejo de interrupciones

---

- El sistema operativo preserva el estado del CPU respaldando los registros y el apuntador de instrucción.
- Determina que tipo de interrupción ocurrió:
  - **Encuestando** a los controladores
  - Con un **sistema de vectores de interrupcion**
- Diferentes *rutinas de servicio de interrupción* determinan que acción debe tomarse ante diferentes tipos de interrupción

# Línea de tiempo de interrupciones



# Solicitudes de E/S

---

- Sin Kernel: después de que comienza la E/S, el control regresa al proceso hasta que la E/S termina
  - La instrucción wait detiene al CPU hasta la próxima interrupción
  - A lo más una solicitud de E/S es atendida en un instante, no hay procesamiento simultáneo de E/S.
  
- Con Kernel: después de que la E/S inicia, el control regresa al proceso sin esperar a que la E/S termine
  - **Llamada al sistema** – solicitud al sistema operativo para permitir al usuario esperar a que la E/S termine
    - Llamada asíncrona: cuando el dispositivo termina informa al CPU con una interrupción, y el kernel informa al usuario
    - Llamada síncrona: el proceso se suspende hasta que el dispositivo informa al kernel que la operación terminó. Mientras, otros procesos se calendarizan.

# Acceso Directo a Memoria (DMA)

---

- Usado por dispositivos de E/S de alta velocidad para que puedan transmitir información a velocidades cercanas a las de la memoria
- El controlador del dispositivo transmite bloques de datos del buffer a memoria principal sin intervención del CPU
- Una sola interrupción se genera por bloque, en lugar de una por byte\*



# Unidades de almacenamiento

---

- Memoria principal – el único almacenamiento grande al que el CPU tiene acceso directo
- Almacenamiento secundario – extensión de la memoria principal que provee gran capacidad de almacenamiento persistente
  - Discos Magnéticos – platos rígidos de metal o vidrio cubiertos con material de escritura magnética
    - ▶ La superficie del disco se divide en **pistas**, los cuales se subdividen en **sectores**
    - ▶ El **controlador de disco** determina la interacción lógica entre el dispositivo y el cpu
  - Discos de estado sólido – Varios chips de memoria flash almacenan la información.
    - ▶ El controlador de disco presenta la misma interfaz que el controlador de un disco magnético. De esa manera, no hace falta hacer ningún cambio en el kernel.

# Jerarquía de almacenamiento

---

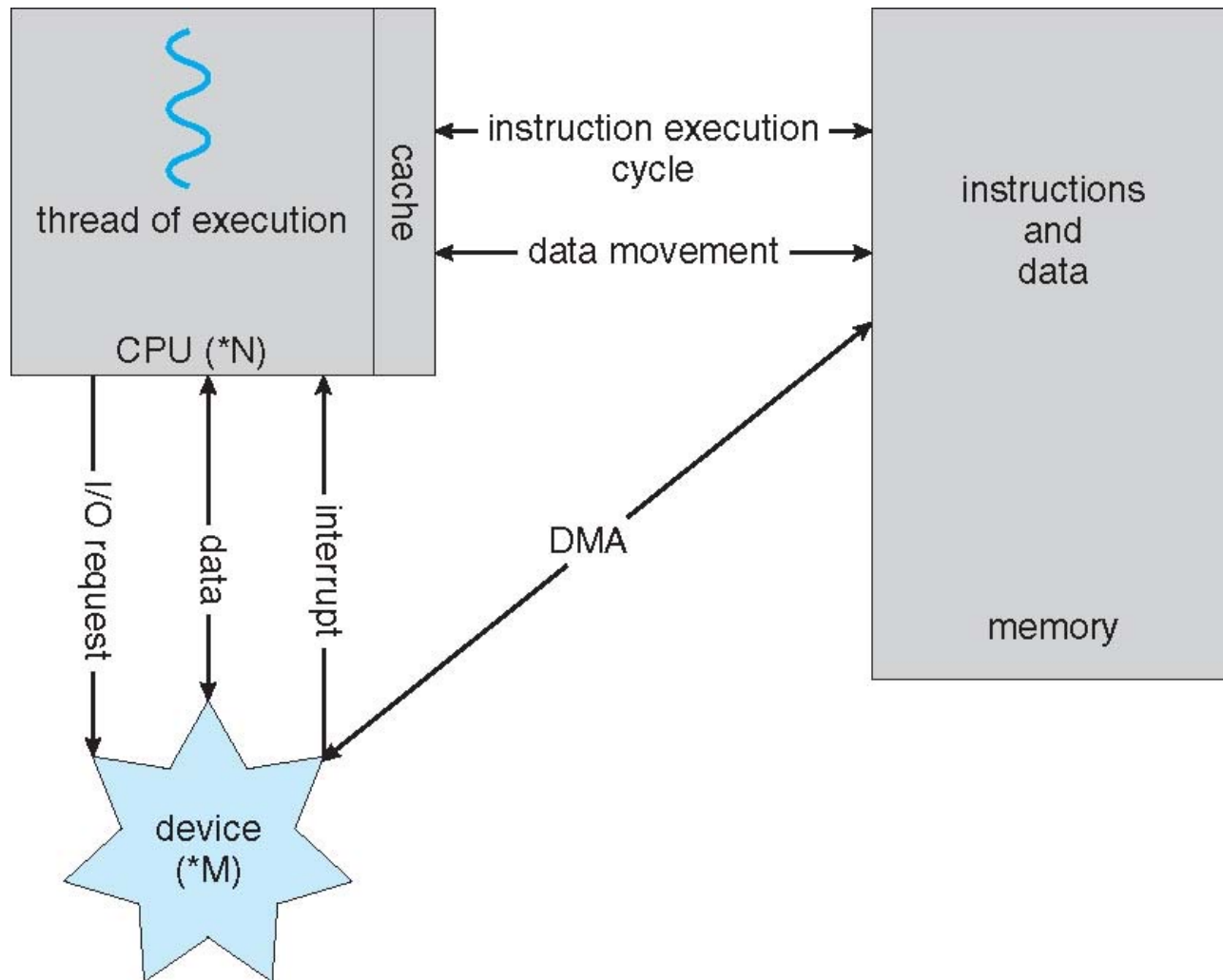
- Los sistemas de almacenamiento se organizan en jerarquías
  - Velocidad
  - Costo
  - Volatilidad
- **Cacheo** – copiar información en un sistema de almacenamiento más veloz; la memoria principal se puede ver como un último cache para el almacenamiento secundario

# Arquitectura de un sistema de cómputo

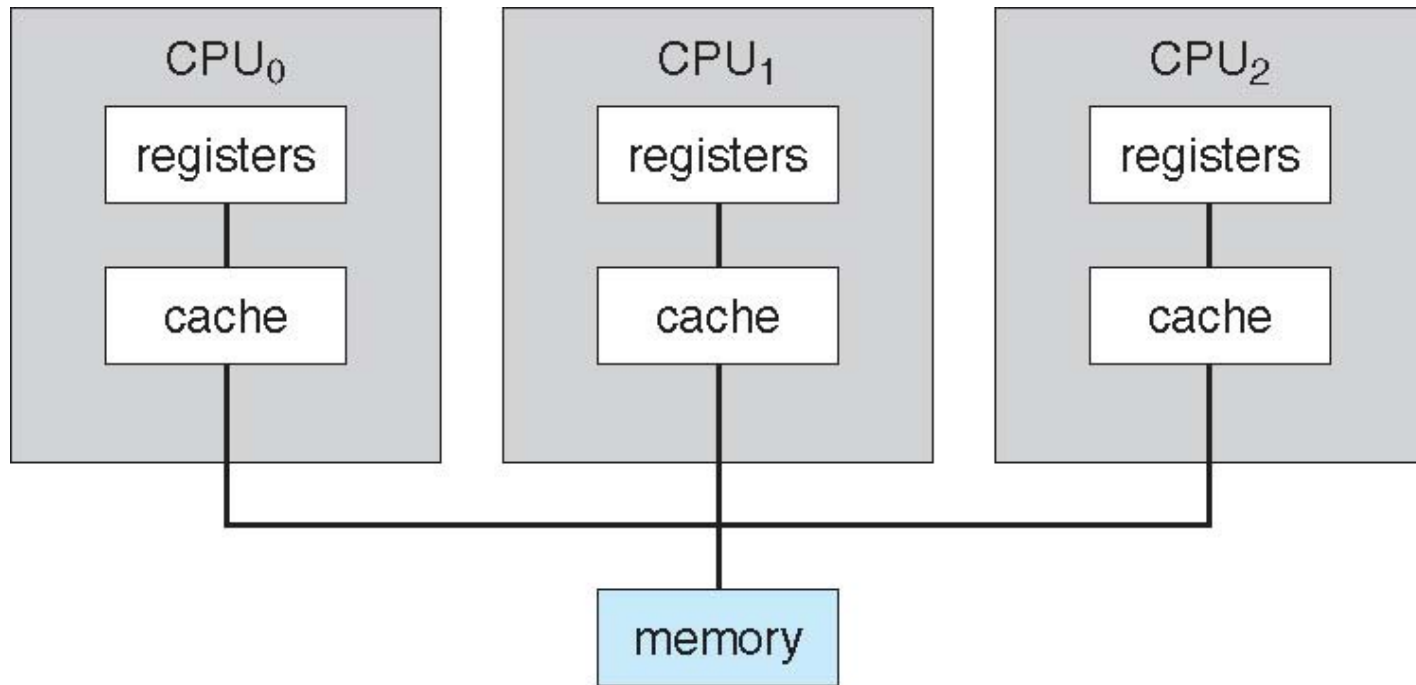
---

- La mayoría de los sistemas contienen un único procesador de propósito general (desde PDA's hasta Mainframes).
  - La mayoría también tiene procesadores de propósito específico.
- Los sistemas **Multiprocesador** crecen en importancia y uso
  - También llamados **sistemas paralelos**, **sistemas fuertemente acoplados**
  - Las ventajas incluyen:
    1. **Mayor rendimiento**
    2. **Mayor confiabilidad – degradación agraciada o tolerancia a fallos**
  - Dos tipos:
    - 4 **Multiproceso Asimétrico**
    - 4 **Multiproceso simétrico**

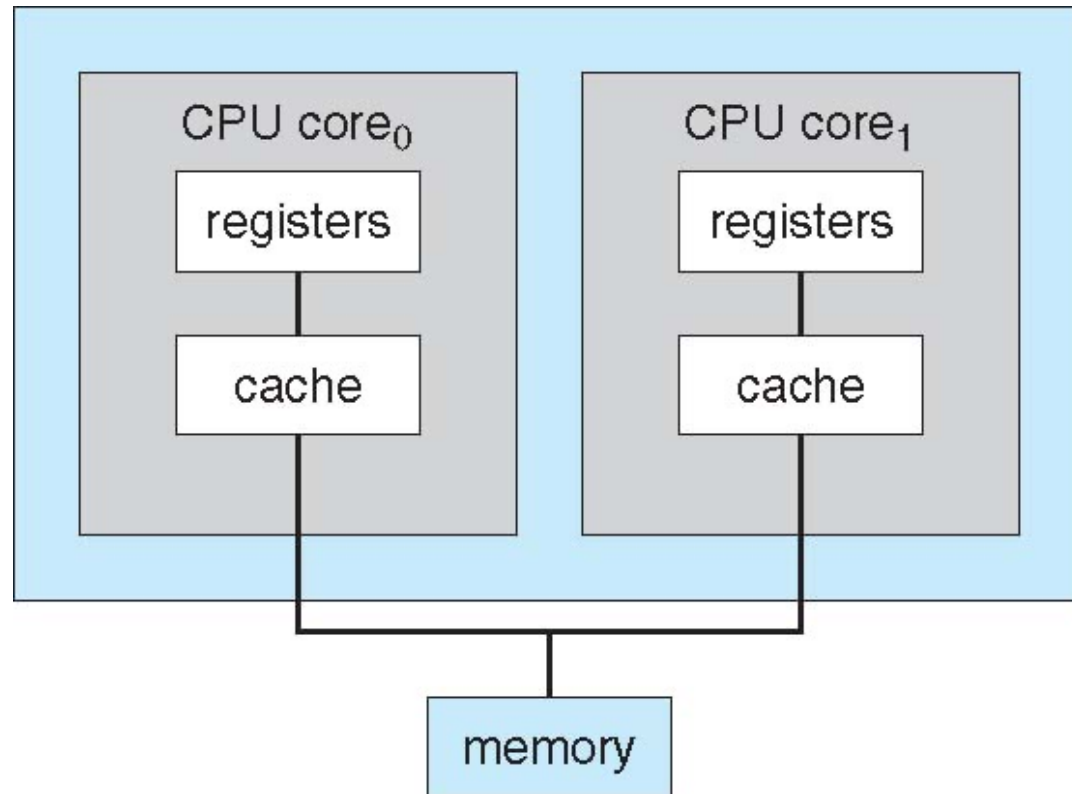
# Cómo trabaja una computadora moderna



# Arquitectura de Multi-Procesamiento Simétrico



# Un diseño de doble núcleo



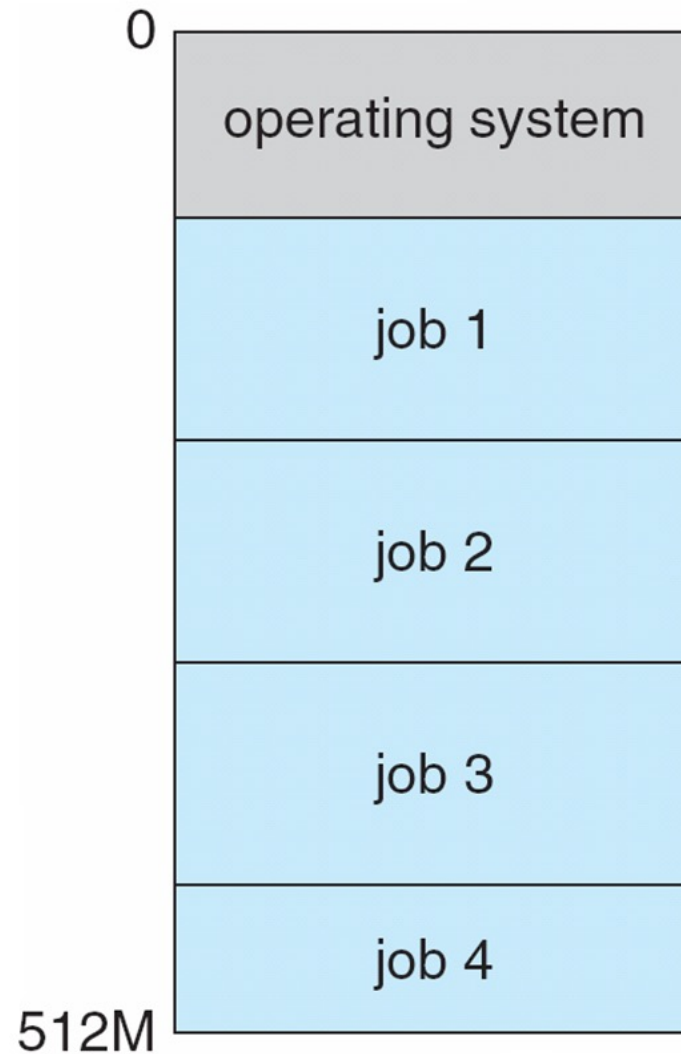
# SO's multitarea

---

- **Tiempo compartido (multitarea)** es una extensión lógica en la cuál el CPU intercambia tareas tan frecuentemente que los usuarios pueden interactuar con cada trabajo cuando está corriendo, creando cómputo **interactivo**
  - El **tiempo de respuesta** debe ser  $< 1$  segundo
  - Cada usuario tiene al menos un programa ejecutándose en memoria  $\Rightarrow$  **proceso**
  - Si hay varios procesos listos para correr en un momento  $\Rightarrow$  **Calendarización del CPU**
  - Si los procesos no caben en memoria, **swapping (intercambio)** los mueve hacia dentro y fuera
  - La **memoria virtual** permite la ejecución de procesos que no están completamente en memoria

# Organización de memoria para un sistema con multiprogramación primitivo

---





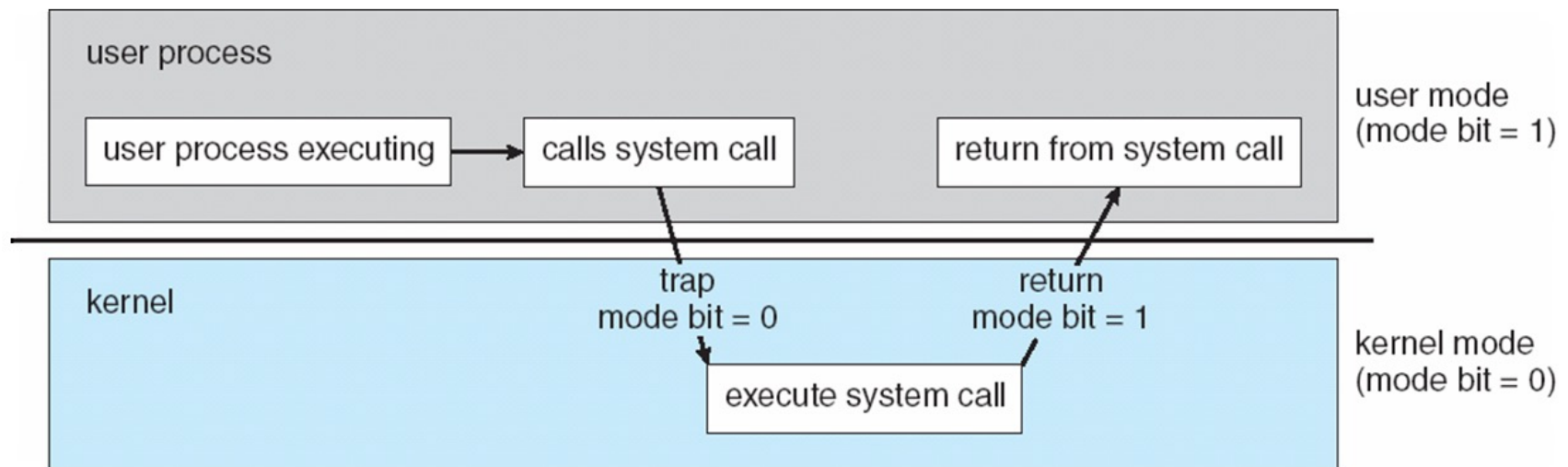
# Operación de un Sistema Operativo

---

- Interrupciones dirigidas por hardware
- Error de software o solicitud crea una **excepción** o **trampa**
  - División entre cero, solicitud de un servicio del S.O,...
- Otros problemas de procesos incluyen ciclo infinito, procesos modificándose uno a otro o al sistema operativo
- **Modo dual** de operación le permite al SO protegerse a si mismo y a otros componentes del sistema
  - **Modo usuario y modo kernel**
  - **Bit de modo** provisto por el hardware
    - ▶ Da la habilidad de distinguir si un proceso está corriendo código de usuario o código de kernel
    - ▶ Algunas instrucciones **privilegiadas**, solo ejecutables en modo kernel
  - Llamadas al sistema cambian el modo a modo kernel, el regreso de la llamada lo cambia a modo usuario

# Transición de modo usuario a modo kernel

- Temporizador para prevenir bucles infinitos/ monopolización de recursos
  - Programar interrupción para después de un periodo específico
  - Configurarla antes de calendarizar un proceso para retomar el control o terminar un programa que se pasa del tiempo asignado.



# Hoy en día un S.O. ...

---

- Carga programas y los ejecuta
- Les da memoria.
- Les asigna tiempo de procesador
- Les permite comunicarse
- Los elimina
- Controla los dispositivos
- Provee E/S de alto nivel (archivos, sockets)
- Ofrece recursos gráficos (ventanas)
- Administra energía
- .....
- Tareas administrativas, cuyo propósito final es permitir que haya aplicaciones.

# Administración de procesos

---

- Un proceso es un programa en ejecución. Es una unidad de trabajo en el sistema. Un *programa* es una entidad pasiva, un *proceso* es una unidad activa.
- Los procesos necesitan recursos para llevar a cabo su trabajo
  - CPU, memoria, E/S, archivos, red
  - Datos de inicialización
- La terminación de un proceso requiere la recuperación de los recursos que se le asignaron
- Un proceso de hilo único tiene un solo **contador de programa** especificando la localidad de la siguiente instrucción a ejecutar
  - Los hilos ejecutan instrucciones secuencialmente, una a la vez, hasta que terminan
- Los procesos multi-hilos tienen un contador de programa por hilo
- Típicamente un sistema tiene varios procesos, algunos de usuario, algunos de sistema, ejecutandose concurrentemente en uno o varios CPU's
  - Concurrencia multiplexando los CPU's entre los diferentes procesos/hilos

# Actividades de administración de procesos

---

El sistema operativo es responsable de las siguientes actividades relacionadas con la administración de procesos

- Crear y destruir procesos de usuario y sistema
- Suspende y reanuda un proceso
- Proveer mecanismos para sincronización de procesos
- Proveer mecanismos para comunicación de procesos
- Proveer mecanismos para manejo de interbloqueos

# Administración de memoria

---

- Todos los datos deben estar en memoria antes y después de procesar
- Todas las instrucciones en memoria para poder ejecutarse
- La administración de memoria determina que hay en memoria mientras
  - Optimiza la utilización del CPU y la respuesta del sistema a los usuarios
- Actividades de administración de memoria
  - Llevar registro de qué partes de la memoria se están usando y por quién
  - Decidir que procesos (o partes de) y datos se mueven dentro y fuera de la memoria
  - Asignar y recolectar memoria según sea necesario

# Subsistema de E/S

---

- Uno de los propósitos de los SO's es esconder a los usuarios las peculiaridades de los dispositivos de hardware
- El subsistema de E/S es responsable de
  - La administración de la memoria de E/S incluyendo *buffering* (almacenamiento temporal de datos mientras se transfieren), *cacheo* (almacenamiento de datos en almacenamiento más rápido para mayor rendimiento), *spooling* (evitar el traslape de la salida de un trabajo con la salida de otro trabajo)
  - Interface general de manejadores de dispositivos
  - Manejadores (*drivers*) para dispositivos de hardware específicos

# Administración de almacenamiento

---

- El SO brinda una imagen uniforme, lógica, del almacenamiento de información
  - Abstrae las propiedades físicas de los dispositivos mediante la unidad lógica de almacenamiento - **archivo**
  - Cada medio es asociado a un dispositivo (ej. Unidad de disco, unidad de almacenamiento)
    - ▶ Propiedades variables incluyen la velocidad de acceso, capacidad, tasa de transferencia, método de acceso (secuencial o aleatorio)



# Administración de los sistemas de archivos

---

- Los archivos generalmente se organizan en directorios
- El control de acceso en la mayoría de los sistemas, determina quién puede acceder a que información
- Actividades del SO incluyen
  - ▶ Crear y eliminar archivos y directorios
  - ▶ Primitivas para manipular archivos y directorios
  - ▶ Mapear archivos a almacenamiento secundario
  - ▶ Mapear archivos a regiones de memoria
  - ▶ Respalidar archivos en medios estables (no-volátiles)

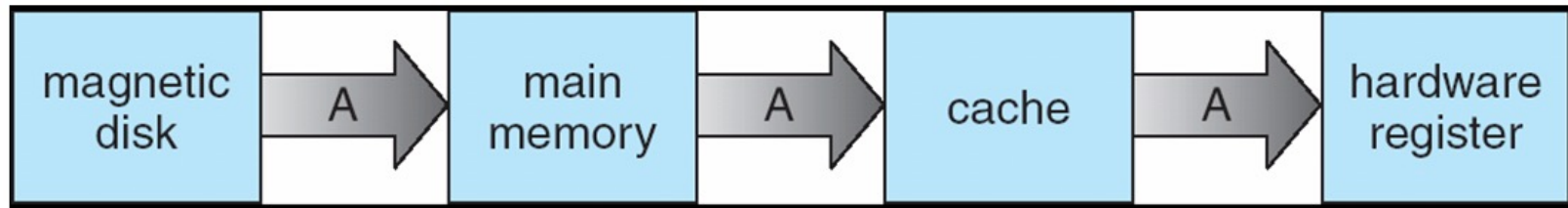
# Administración de almacenamiento masivo

---

- Actividades del SO
  - Administración del espacio libre
  - Asignación de almacenamiento
  - Calendarización del disco

## Migración del entero A de disco a registro

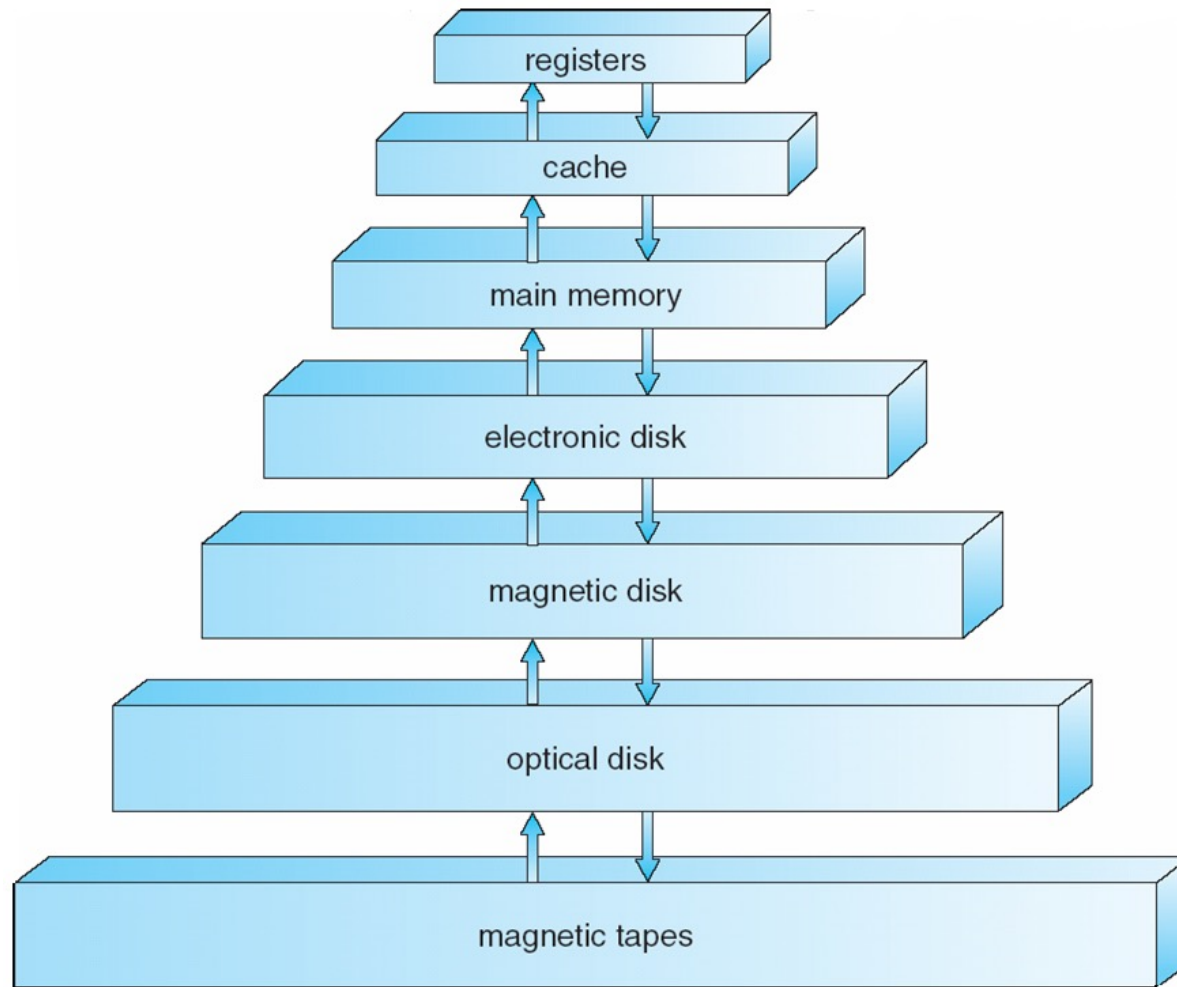
- Los ambientes multitarea deben tener cuidado de usar el valor más reciente, sin importar en que parte de la jerarquía de almacenamiento está almacenado



- Los ambientes multiprocesadores deben brindar coherencia de cachés en hardware para que todas las CPU's tengan el valor más reciente en su caché.
- La situación en ambientes distribuidos es mucho más compleja.
  - Pueden existir varias copias de un mismo dato, sin coherencia automática por hw.

# Jerarquía de los dispositivos de almacenamiento

---



# Protección y seguridad

---

- **Protección**– cualquier mecanismo para controlar el acceso de procesos o usuarios a recursos definidos por el SO
- **Seguridad**– Defensa del sistema contra ataques internos y externos
  - Rango enorme, incluyendo negación de servicio, gusanos, virus, robo de identidad, robo de servicio
- Generalmente los SO's distinguen a sus usuarios, para determinar quién puede hacer que cosa
  - Identidades de usuario (**user ID's**) incluyen nombre y número asociado, uno por usuario
  - El Id de usuario es asociado a todos los archivos y procesos de usuario para determinar el control de acceso
  - El identificador de grupo (**group ID**) permite que se definan conjuntos de usuarios a los que se les administran permisos, los cuales se asocian también a archivos y procesos
  - **Escalamiento de privilegios** Permite a los usuarios incrementar su ID de usuario efectiva para obtener más derechos

# Sistemas Operativos de fuente abierta

---

- Los sistemas operativos que se ofrecen en forma de código fuente en lugar de simplemente código binario de **fuentes cerrados**
- Para contrarrestar el movimiento de **protección de copia** y la **administración de derechos digital (DRM)**
- Comenzado por la **Free Software Foundation (FSF)**, la cual tiene “izquierdos de copia” **GNU Public License (GPL)**
- Ejemplos incluyen **GNU/Linux**, **BSD UNIX** (incluyendo núcleo de **Mac OS X**), y **Sun Solaris**

**Pintos**



---

**Fin del capítulo 1**