

Arquitectura de Ordenadores 2006/07

# Introducción

Prof. Dr. Ralf E.D. Seepold  
[ralf.seepold@uc3m.es](mailto:ralf.seepold@uc3m.es)



# Índice

- Breve descripción
- Objetivos
- Material de teoría
- Material de laboratorio
- Evaluación

# Breve descripción

Breve descripción	
Curso:	4º
Créditos:	6 (4 teoría + 2 laboratorio/prácticas)
Cuatrimestre:	1º
Horas/Semana:	4
Horario:	Martes 9:00-11:00, 19:00-21:00 Viernes 9:00-11:00, 18:00-20:00
Grupo:	91/92- 4º I Telecomunicación
Aula de teoría:	2.3.C.04/4.1.E.01
Aula de prácticas:	1.2.G.03/04 - 4.S.D.03/04
Profesores:	Abelardo Pardo, Luis Sánchez
	Ralf Seepold
Profesores de Laboratorio:	Norberto Fernández García
	Jorge Blanco Rodríguez
	Eduardo Martín Rojo

# Objetivos de la asignatura

- Ofrecer una profunda visión del tema de Sistemas Operativos (SO) y varios conceptos de soporte
- Explicar conceptos básicos de SO
- Introducir el lenguaje de programación C que es fundamental para entender muchas implementaciones de SO
  - *Nota: Hay enlaces a tutoriales en la página web de la asignatura*
- Enfocar técnicas recientes como los SO distribuidos
- Preparar todos los conceptos aplicados en la asignatura de **Laboratorio de Arquitectura de Ordenadores** (2º cuatrimestre) y de este modo establecer una fuerte unión entre ambas.
- La participación con éxito contribuye a los siguientes objetivos:
  - Conocimiento de conceptos fundamentales y algoritmos de SO modernos
  - Visión clara de la arquitectura de SO y conocimiento de términos relevantes cuando se diseñan SO
  - Aplicación con éxito de la teoría de la asignatura en varias prácticas que acompañan de cerca cada tema
  - Aprendizaje de un lenguaje adicional de programación como C
  - Preparación de todos los conceptos teóricos requeridos para la implementación avanzada de prácticas en la asignatura de *Laboratorio de Arquitectura de Ordenadores*

# Material de teoría

- Serán ofrecidas principalmente como transparencias para descargar
- Se sugiere ampliar las transparencias mediante:
  - Notas individuales tomadas durante la asignatura
  - Estudio de literatura adicional
    - **A. Silberschatz, P. Galvin, Operating Systems Concepts - 6th Edition/Windows XP Update, ISBN: 0-471-25060-0, John Wiley & Sons, 2002. Library: L/S 004.451 SIL**
    - A. Tanenbaum, Modern Operating Systems, ISBN: 0130926418, Prentice Hall, 2001. Library: L/S 004.451 TAN
    - A. Tanenbaum, Maarten van Steen, Distributed Systems: Principles and Paradigms, ISBN: 0-13-088893-1, Prentice Hall, 2002. Library: L/D 004.272.43 TAN
    - Brian W. Kernighan, Dennis M. Ritchie, The C programming language, 2nd ed., ISBN: 0-13-110362-8, Prentice Hall, 1989. Library: L/S 004.438 C KER 1988
    - John L. Hennessy, David A. Patterson, David Goldberg, Computer Architecture, ISBN: 1558607242, Morgan Kaufmann, 2002. Se dará una explicación adicional en la pizarra

# Material de laboratorio

- Los laboratorios deben servir para reforzar los conceptos teóricos
- Se pueden introducir nuevos temas
- El material de laboratorios es relevante para el examen
- Las tareas pueden ser descargadas de la página web del laboratorio
- Las prácticas se realizan en equipos de dos personas
- Preparación de los laboratorios
  - Estudiar el material de la asignatura (transparencias y material adicional)
  - Estudiar las partes relevantes en la literatura fundamental
  - Estudiar la literatura adicional dada para el laboratorio
- Entrega
  - Subida a través de una página de entrega

# Evaluación

- La descripción detallada de la evaluación de la asignatura se encuentra en la página web

# Arquitectura de los Ordenadores



# Arquitectura de Ordenadores:

## Índice

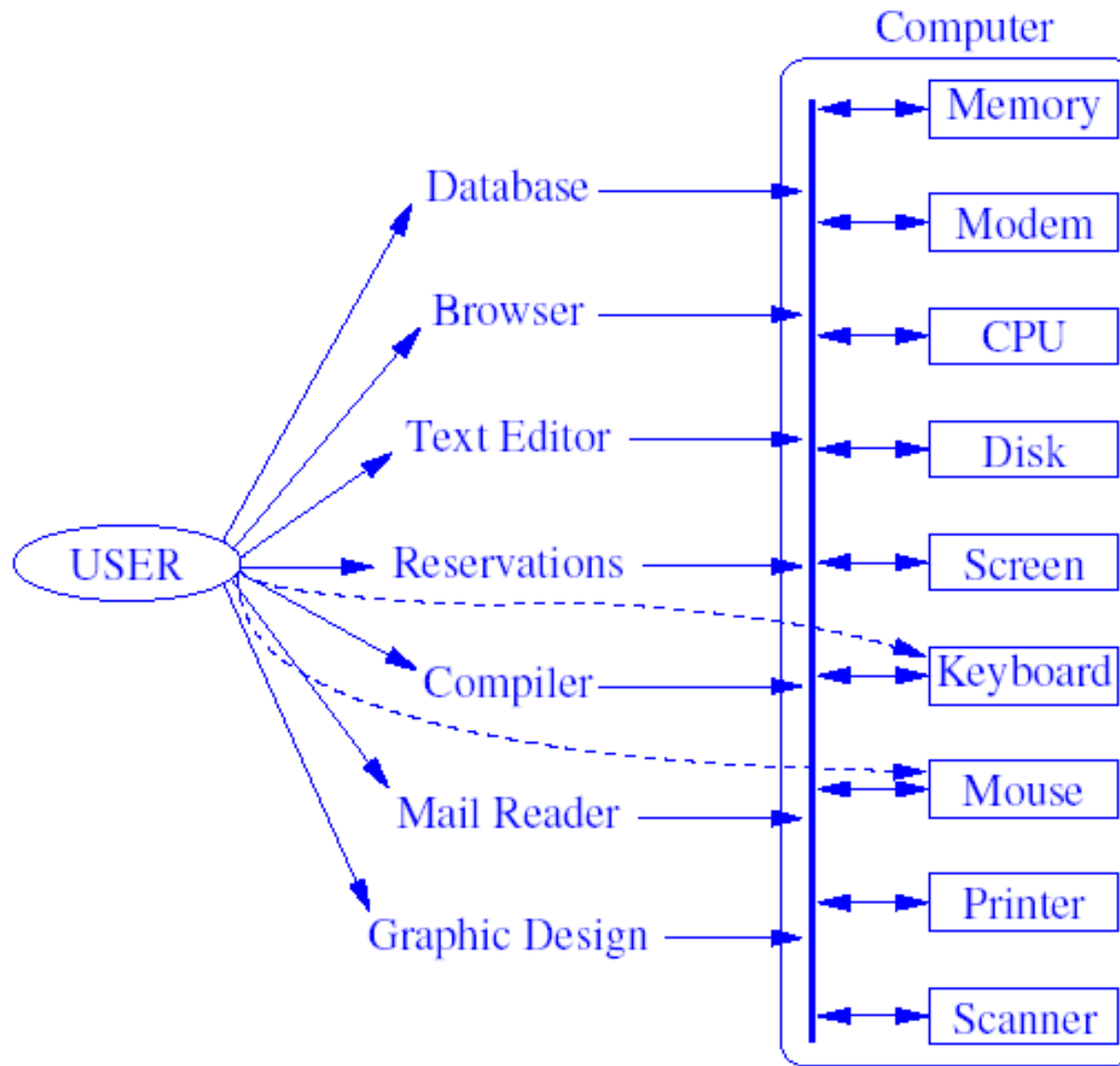
- Definición de SO
- Historia
- Vista de componentes
- Diferentes tipos de SO
  - Mainframe
  - Escritorio
  - Multiprocesador
  - Distribuido
  - En clúster
  - Tiempo real
  - Manejable
  - Tarjeta inteligente
- Estructura de un Sistema Informático
  - Operación de un Sist. Informático
  - Estructura E/S
  - Estructura de almacenamiento
  - Estructura jerárquica
  - Protección del Hardware
- Estructura de un Sistema Operativo
  - Componentes del sistema
    - Administración de procesos
    - Administración de Memoria Principal
    - Administración de ficheros
    - Administración del Sistema de E/S
    - Administración secundaria
    - Conexión de red
    - Sistema de Protección
    - Sistema de intérprete de comandos
  - Llamadas a sistema
  - Diseño del sistema
    - Monolítico
    - Por capas
    - Virtuales (máquinas)
- Tareas de un SO

# Una definición para un SO

- “Un sistema operativo es un programa que actúa como un intermediario entre el usuario de un ordenador y el hardware del ordenador. El propósito de un sistema operativo es proporcionar un entorno en el que cada usuario pueda ejecutar programas de una manera *conveniente y eficiente*.”

Silberschatz 2002

# Visión sencilla de material complejo



# Objetivos de diseño

- SO conveniente
- SO eficiente
- Combinación de ambos

# Historia (I)

- Primera generación (1945-55)
  - Implementación de código máquina
  - SO y lenguajes de programación no inventados aún
- Segunda generación (1955-65)
  - Fiabilidad mejorada a través de transistores
  - Distinción entre desarrollador, proveedor/fabricante, operadores y programadores
  - Programas codificados en tarjetas perforadas
  - Principalmente sistemas “mainframe” en modo “batch”

# Historia (II)

- Tercera generación (1965-80)
  - Invención de los circuitos integrados
  - Existencia de líneas paralelas de productos (incompatibles entre ellas)
    - Ordenador orientado a palabras (operación numérica para ciencias e industria)
    - Ordenador orientado a carácter (p.ej. Compañías de seguros)
    - Problemas de portabilidad
  - Problemas que surgen
    - Se requiere software eficiente y comunmente aplicable
    - SO sobrecargado con requerimientos contradictorios

# Historia (III)

- Tercera generación (1965-80) (cont.)
  - Aparición de la crisis del software
    - SO crece en tamaño
    - Las aplicaciones llegan a ser demasiado complejas y erróneas
    - Demanda de
      - Diseño estructural
      - Sostenibilidad
      - Fiabilidad
      - Seguridad
      - Programación modular
      - Tipos de datos abstractos y orientados a objetos
    - Consecuencia
      - Aplicación de lenguajes de alto nivel para diseño e implementación
      - El **proceso** se vuelve un mecanismo con permisos y direcciones privadas y protegidas.

# Historia (IV)

- Tercera generación (1965-80) (cont.)
  - Desarrollo e integración de tecnologías clave
    - Multiprogramación
    - Tiempo compartido (sistema compatible de tiempo compartido)
    - Spooling (operación de periféricos simultáneos en línea)
    - Paralelización
    - Proceso (procesadores virtuales)
    - Memoria virtual
    - Los procesos son la estructura de un SO



# Historia (V)

- Cuarta generación (1980-90)
  - CP/M (programa de control para microcomputadoras)
    - SO basado en disco para Intel 8080
  - MS-DOS (Microsoft disk operating system)
    - Basado en DOS
    - IBM une PC y SO (¡éxito!)
  - MacOS
    - Desarrollo del GUI
  - UNIX
    - Extensión para un GUI (concepto X del MIT)
    - Posterior desarrollo en Motif

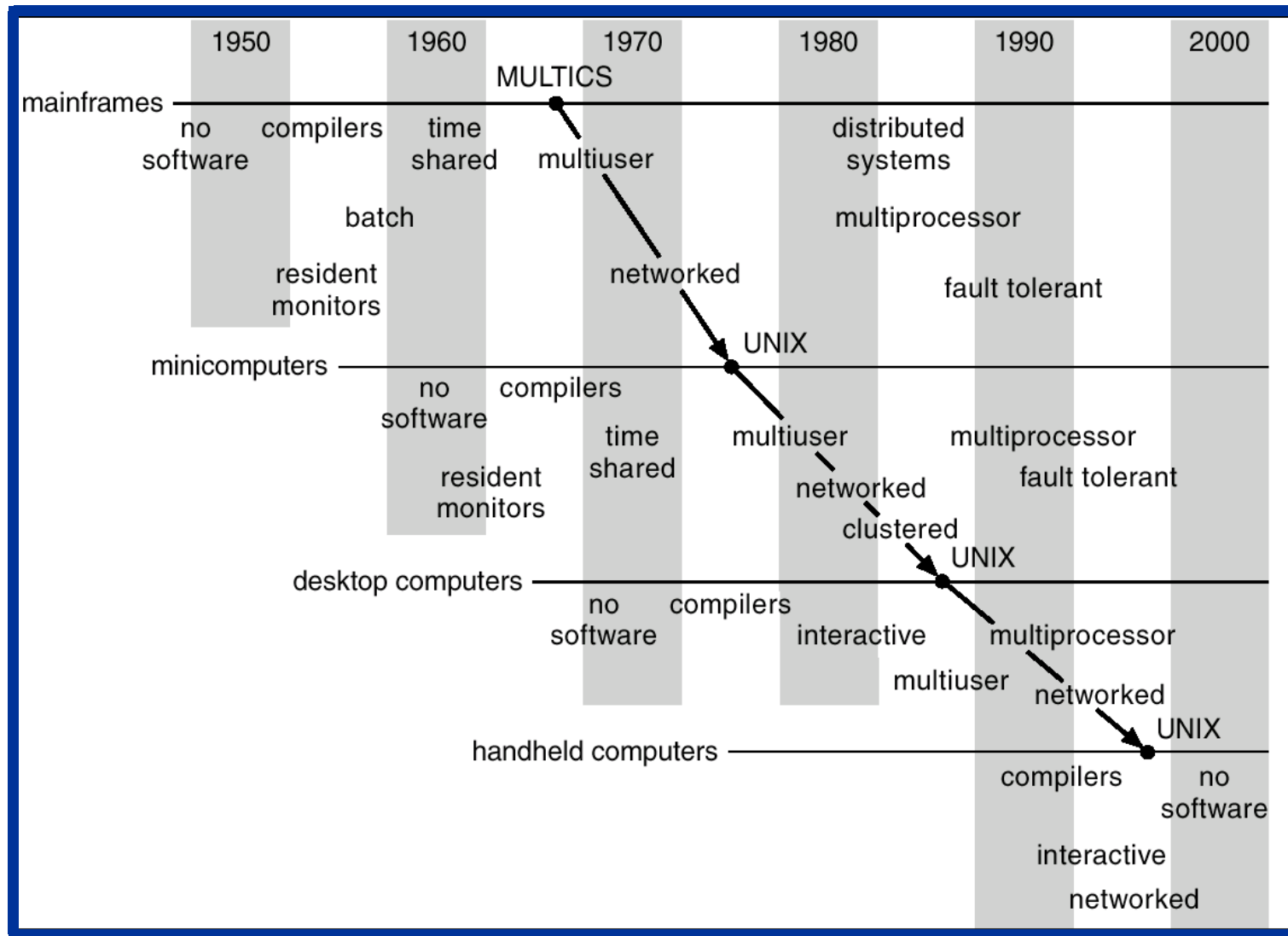
# Desarrollo de UNIX/Linux

- Empezó con el MULTICS (MULTI-plexed information and computing system)
  - Desarrollado por Bell Labs, MIT y General Electric
  - Continuado sólo por MIT
- Portabilidad bastante difícil
- Nuevo lenguaje de programación desarrollado: C
- C es EL estándar para programación de sistemas
- Tanenbaum desarrolló MINIX (UNIX con propósito educativo)
- Linus Torvalds lanza Linux 0.0.1 en 1991
  - La versión 1.0 en 1994 contiene 165k líneas
  - La versión 2.0 en 1996 contiene 470k líneas

# Linux: El comienzo

- > Summary: small poll for my new operating system
- > Message-ID: <1991Aug25.205708.9541@klaava.Helsinki.Fi>
- > Date: 25 Aug 91 20:57:08 GMT
- > Organization: University of Helsinki
- >
- > Hello everybody out there using minix –
- > I'm doing a (free) operating system (just a hobby, won't be big and
- > professional like gnu) for 386(486) AT clones. This have been brewing since
- > april, and is starting to get ready. I'd like any feedback on things people
- > like/dislike in minix, as my OS resembles it somewhat (same physical layout of
- > the file system (due to practical reasons) among other things).
- >
- > I've currently ported bash(1.08) and gcc(1.40), and things seem to work.
- > This implies that I'll get something practical within a few months, and I'd like
- > to know what features most people would want. Any suggestions are welcome,
- > but I won't promise I'll implement them :-)
- > Linus (torvalds@kruuna.helsinki.fi)
- > P.S. It is NOT portable (uses 386 task switching etc.), and it probably never will
- > support anything other than AT-harddisks, as that's all I have :-).

# Evolución de características



# Historia (cont.)

- Desarrollo de tecnologías clave
  - SO en red
  - Estandarización
    - OSI, TCP/IP, POSIX, OSF, NFS etc.
- Desarrollo de hilos (procesos ligeros)
- Sistemas masivamente paralelos
- Sistemas multimedia
- Sistemas empotrados
- Sistemas interoperables
  - Sistemas Heterogéneos
  - Emulación

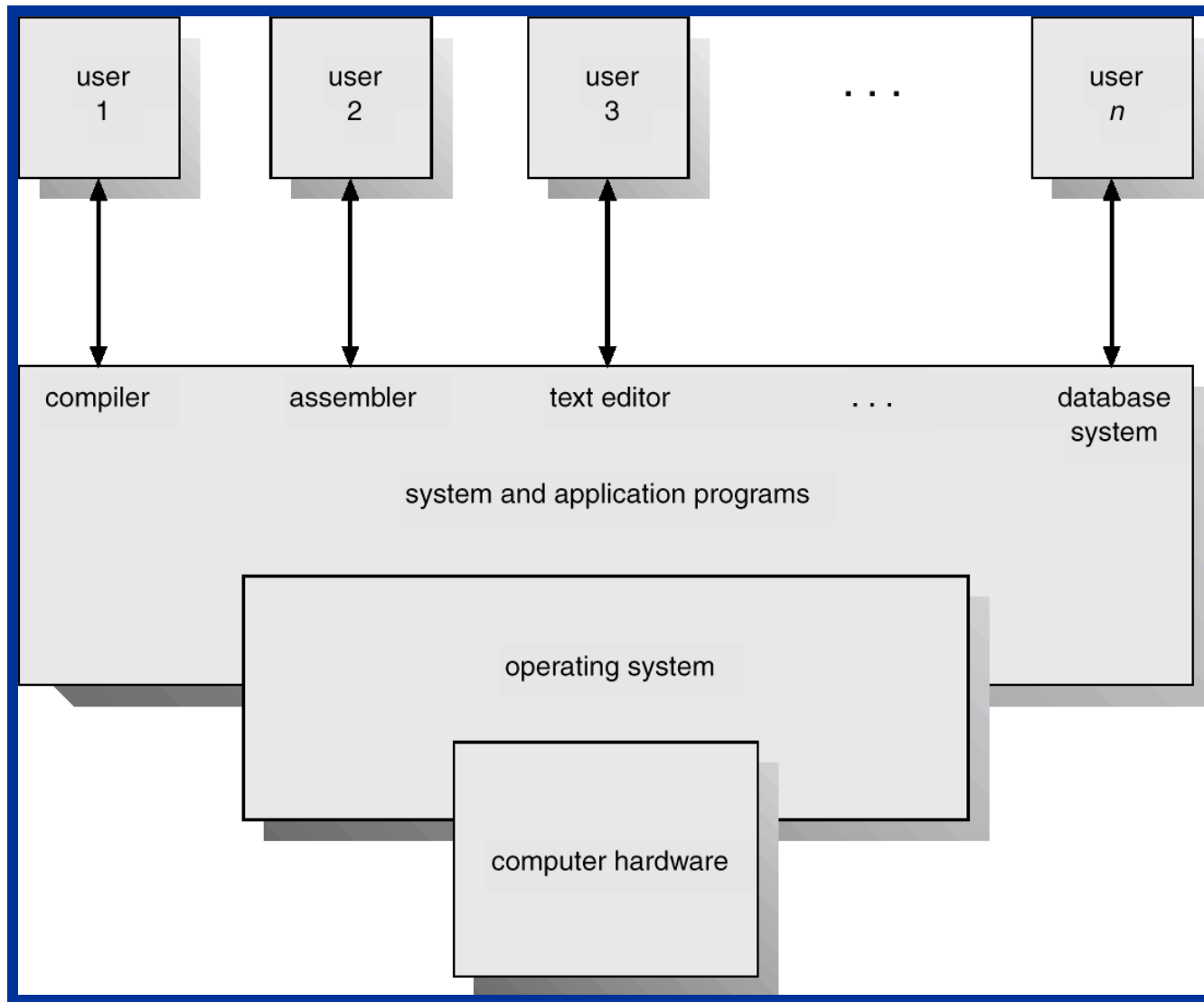
# Índice

- Definición de SO
- Historia
- Vista de componentes
- Diferentes tipos de SO
  - Mainframe
  - Escritorio
  - Multiprocesador
  - Distribuido
  - En clúster
  - Tiempo real
  - Manejable
  - Tarjeta inteligente
- Estructura de un Sist. Informático
  - Operación de un Sist. Informático
  - Estructura E/S
  - Estructura de almacenamiento
  - Estructura jerárquica
  - Protección del Hardware
- Estructura de un Sist. Operativo
  - Componentes del sistema
    - Administración de procesos
    - Administración de Memoria Principal
    - Administración de ficheros
    - Administración del Sistema de E/S
    - Administración secundaria
    - Conexión de red
    - Sistema de Protección
    - Sistema de intérprete de comandos
  - Llamadas a sistema
  - Diseño del sistema
    - Monolítico
    - Por capas
    - Virtuales (máquinas)
- Tareas de un SO

# Vista de componentes

1. Hardware – proporciona recursos de computación básicos (CPU, memoria, dispositivos E/S).
2. Sistema operativo – controla y coordina el uso del hardware entre los distintos programas de aplicación para los distintos usuarios.
3. Programas de aplicación – define las formas en que los recursos del sistema se usan para resolver los problemas de cómputo de los usuarios (compiladores, sistemas de bases de datos, videojuegos, programas de negocio).
4. Usuarios (gente, máquinas, otros ordenadores).

# Componentes (cont.)





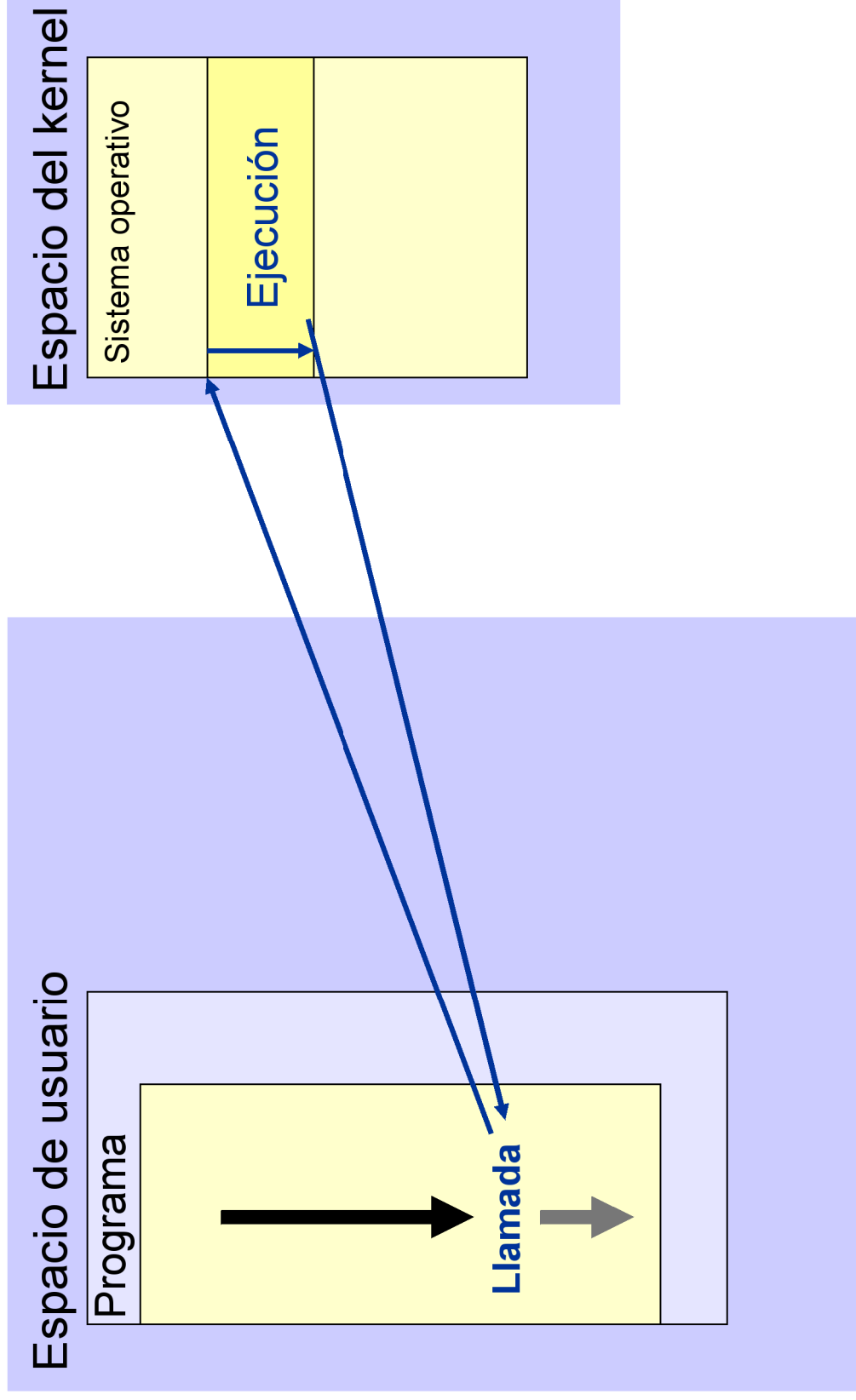
# Vista de usuario

- Un usuario a menudo se sienta enfrente de un ordenador
  - El objetivo principal: “fácil de usar”
- Usuarios que se sientan en frente de un terminal conectado a un mainframe o mini-ordenador
  - El objetivo principal: “utilización de recursos”
- Usuarios que se colocan en una estación de trabajo (conectada a una red o a otras estaciones de trabajo/servidores)
  - SO diseñado para proporcionar un compromiso entre usabilidad individual y utilización de recursos
- PDAs (Personal digital assistant; conexión inalámbrica)
  - SO principalmente diseñado buscando usabilidad pero también orientado al rendimiento (debido a los limitados recursos de las baterías)

# Vista del sistema

- Desde el punto de vista de un ordenador es el software que está más firmemente relacionado al hardware específico
- Un SO puede verse como el asignador de recursos
  - Tiempo de CPU
  - Espacio de memoria
  - Almacén de ficheros, etc.
- Un SO puede verse como un programa de control
  - Administrar la ejecución de programas
  - Previene errores
  - Previene uso impropio
  - Controla dispositivos de E/S

# Ejecución de programa



# Segunda definición

- No hay disponible una definición aceptada en general
- Definición más común
  - Un SO es el único programa que corre todo el tiempo en el ordenador, todos los otros son programas de aplicación
- Kernel: Porción del sistema operativo que está ejecutando continuamente.

# Índice

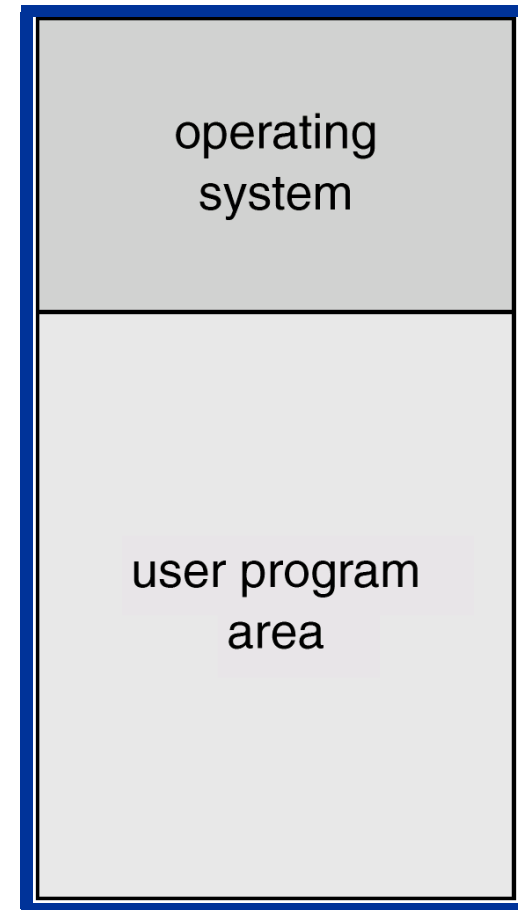
- Definición de SO
- Historia
- Vista de componentes
- **Diferentes tipos de SO**
  - **Mainframe**
  - **Escritorio**
  - **Multiprocesador**
  - **Distribuido**
  - **En clúster**
  - **Tiempo real**
  - **Manejable**
  - **Tarjeta inteligente**
- Estructura de un Sist. Informático
  - Operación de un Sist. Informático
  - Estructura E/S
  - Estructura de almacenamiento
  - Estructura jerárquica
  - Protección del Hardware
- Estructura de un Sist. Operativo
  - Componentes del sistema
    - Administración de procesos
    - Administración de Memoria Principal
    - Administración de ficheros
    - Administración del Sistema de E/S
    - Administración secundaria
    - Conexión de red
    - Sistema de Protección
    - Sistema de intérprete de comandos
  - Llamadas a sistema
  - Diseño del sistema
    - Monolítico
    - Por capas
    - Virtuales (máquinas)
- Tareas de un SO

# Diferentes tipos de SO: Entornos informáticos

- Informática tradicional
  - “Entorno típico de oficina”
- Informática basada en Web
  - PCs
  - PDAs
  - Teléfonos móviles
- Informática empotrada
  - Informática empotrada en tiempo real
  - Proporciona sólo características primitivas
  - Su uso continúa en expansión
- ... y más

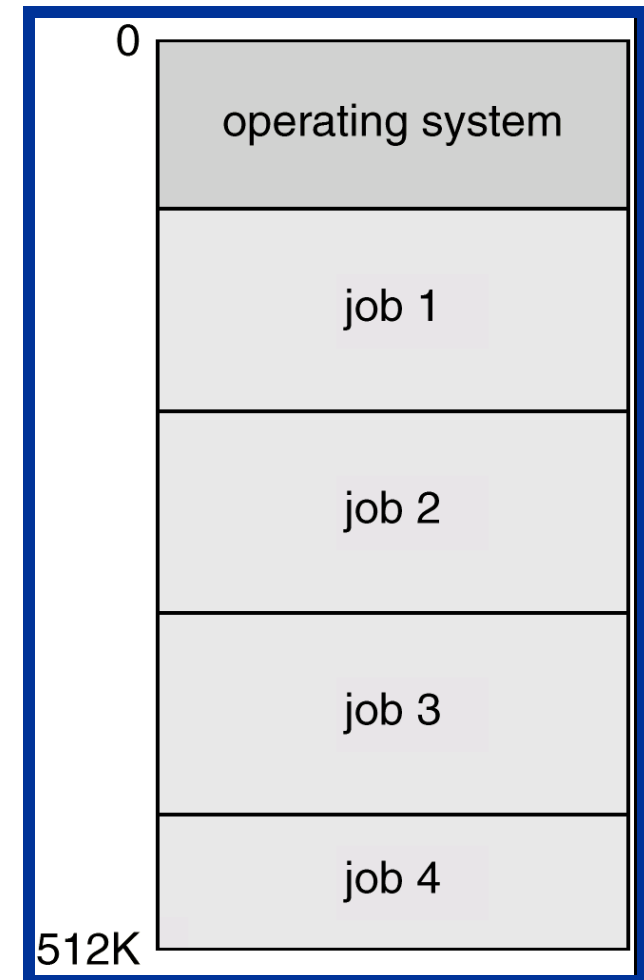
# Sistemas “mainframe”: Sistemas por lotes (“batch systems”)

- Dispositivos de entrada común: lector de tarjetas
- Dispositivos de salida común: impresora, cinta, tarjetas
- Pasa un tarjeta perforada por un operador
  - Programa, datos, información de control
- Recibe el resultado de vuelta (de horas o días)
- Los operadores colocaban **en lotes** trabajos similares juntos
- Problemas
  - Velocidad de entrada de las tarjetas 3 órdenes de magnitud más lenta que la CPU



# Sistemas “mainframe”: Sistema multiprogramado

- Planificar tareas es la habilidad de la multiprogramación
- Incrementa la utilización de CPU y organiza tareas para que la CPU tenga siempre una que ejecutar
- Muchas tareas se mantienen en memoria
- Corre las tareas hasta que alguna necesitan alguna operación E/S (tiempo desocupado para la CPU)
- Cambia a una tarea diferente mientras la otra está esperando





# Sistemas “mainframe”:

## Sistemas de tiempo compartido

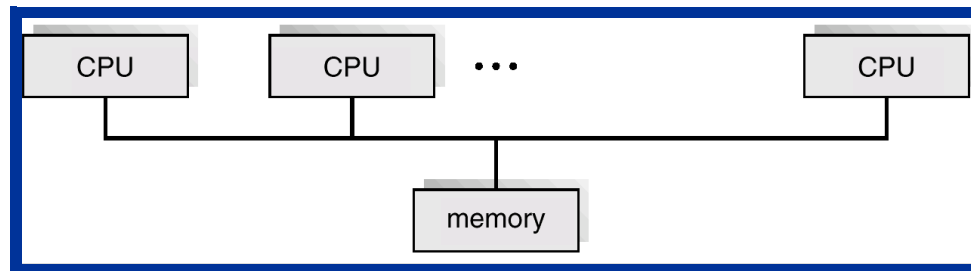
- Tiempo compartido o multitarea es la extensión lógica de la multiprogramación
- La CPU ejecuta múltiples tareas, conmuta entre ellas muy frecuentemente
- El usuario puede inter-actuar con los programas
- El tiempo de respuesta debe ser corto
- La impresión es que un usuario usa el sistema exclusivamente
- Un programa que está en la memoria y en ejecución se le llama **proceso**
- Los procesos se conmutan dentro y fuera de la memoria, gracias al concepto de **memoria virtual**

# Sistemas de escritorio

- Ordenadores personales (aparecidos en 1970)
- Se movieron desde el puro uso “personal” a entornos más compartidos
- El hardware cuesta muy poco. El uso individual es todavía común
- Los SO siguieron esa tendencia
- Puede adoptar tecnología desarrollada para sistemas operativos grandes aunque a menudo los individuos tienen uso exclusivo del ordenador y no necesitan una utilización avanzada de las características de protección.
- Desarrollos recientes
  - De MS-DOS a Microsoft Windows
  - MacOS X (núcleo basado en Mach y FreeBSD UNIX)
  - Linux (sistema tipo UNIX de código abierto)

# Sistemas multiprocesador

- También denominados en paralelo o sistemas fuertemente acoplados, están creciendo en importancia
- Varios procesadores comparten el reloj del bus y (a veces) memoria o dispositivos periféricos
- Ventajas
  - Mayor rendimiento
  - Economía de escala
  - Mayor fiabilidad
- Multiprocesamiento Simétrico (SMT)
  - Cada procesador corre una copia idéntica del SO
  - Coste adicional para la comunicación entre ellos



# Sistemas multiprocesador (cont.)

- Multiprocesamiento asimétrico
  - Cada procesador está asignado a una tarea específica
  - Un procesador maestro controla el sistema (relación maestro-esclavo)

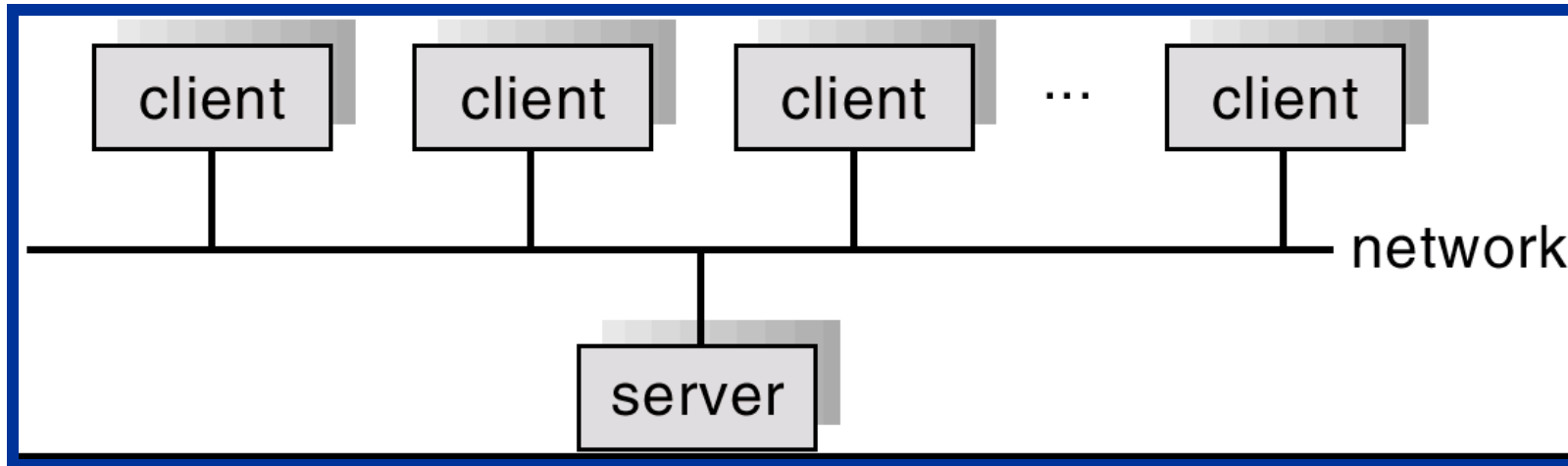
# Sistemas distribuidos

- El concepto básico es una red
  - Local-area network (LAN)
  - Wide-area network (WAN)
  - Metropolitan-area networks (MAN)
  - Small-area network (p.ej. con Bluetooth)
- Conectada a través de protocolos
  - TCP/IP
  - Otros propietarios etc.
- Sobre muy diferentes medios
  - Cable de cobre, estándares de fibra, transmisión inalámbrica, por satélite, radio etc.

# Sistemas distribuidos:

## Sistemas cliente-servidor

- Sistemas centralizados que actúan como servidor de sistemas para satisfacer las peticiones generadas por sistemas clientes
- Dos áreas principales se pueden distinguir
  - Sistemas de servicio de cálculo
  - Sistemas de servicio de ficheros



# Sistemas distribuidos: Sistemas “peer-to-peer”

- Después de la introducción de WWW (mediados los 90) las capacidades de red de un ordenador (SO) se vuelven muy importantes
- La navegación para acceder a documentos de hipertexto forma parte del software tanto como el email, acceso remoto, cliente y servidor de transferencia de ficheros
- Los sistemas se refieren como sistemas débilmente acoplados (ó sistemas distribuidos)
- Incorporación de sistemas operativos en red
  - Proporcionan características como compartir ficheros a través de la red
  - Sistemas operativos distribuidos (trabajando juntos) deberían dar la impresión de que sólo un SO controla el sistema (y la red)

# Sistemas en “cluster”

- Junta múltiples CPUs
- Compuesto de dos o más sistemas individual acoplados juntos
- Configuración típica:
  - Compartir almacenamiento
  - Unidos de cerca a través de una LAN
- Proporcionan una gran disponibilidad
- Clustering asimétrico
  - Una máquina está en modo “hot standby”
  - Otra máquina esta corriendo la aplicación
- Modo simétrico
  - Dos o más máquinas están corriendo juntas y se monitorizan mutuamente



# Sistemas en Tiempo Real

- SO de propósito especial
- Rígidamente restricciones de tiempo enfocadas a la operación de un procesador o al flujo de datos
- Ejemplos
  - Sistemas automóbiles
  - Sistemas médicos
  - Sistemas de armamento
  - Sistema popular: VxWorks
- El procesamiento se debe hacer dentro de un tiempo definido o el sistema falla
- Sistemas en Tiempo Real Duros
  - Garantizan que una tarea crítica se completa a tiempo
- Sistemas en Tiempo Real Blandos
  - La tarea crítica tiene prioridad sobre las otras tareas (y retiene esa prioridad hasta que se completa)

# Sistemas móviles

- Incluyen asistentes digitales personales (PDAs)
  - Examples
    - Microsoft Windows PocketPC 2003
    - PalmOS
- Teléfonos móviles
- Problemas:
  - Memoria limitada
  - Procesadores lentos
  - Pantallas con pequeño display

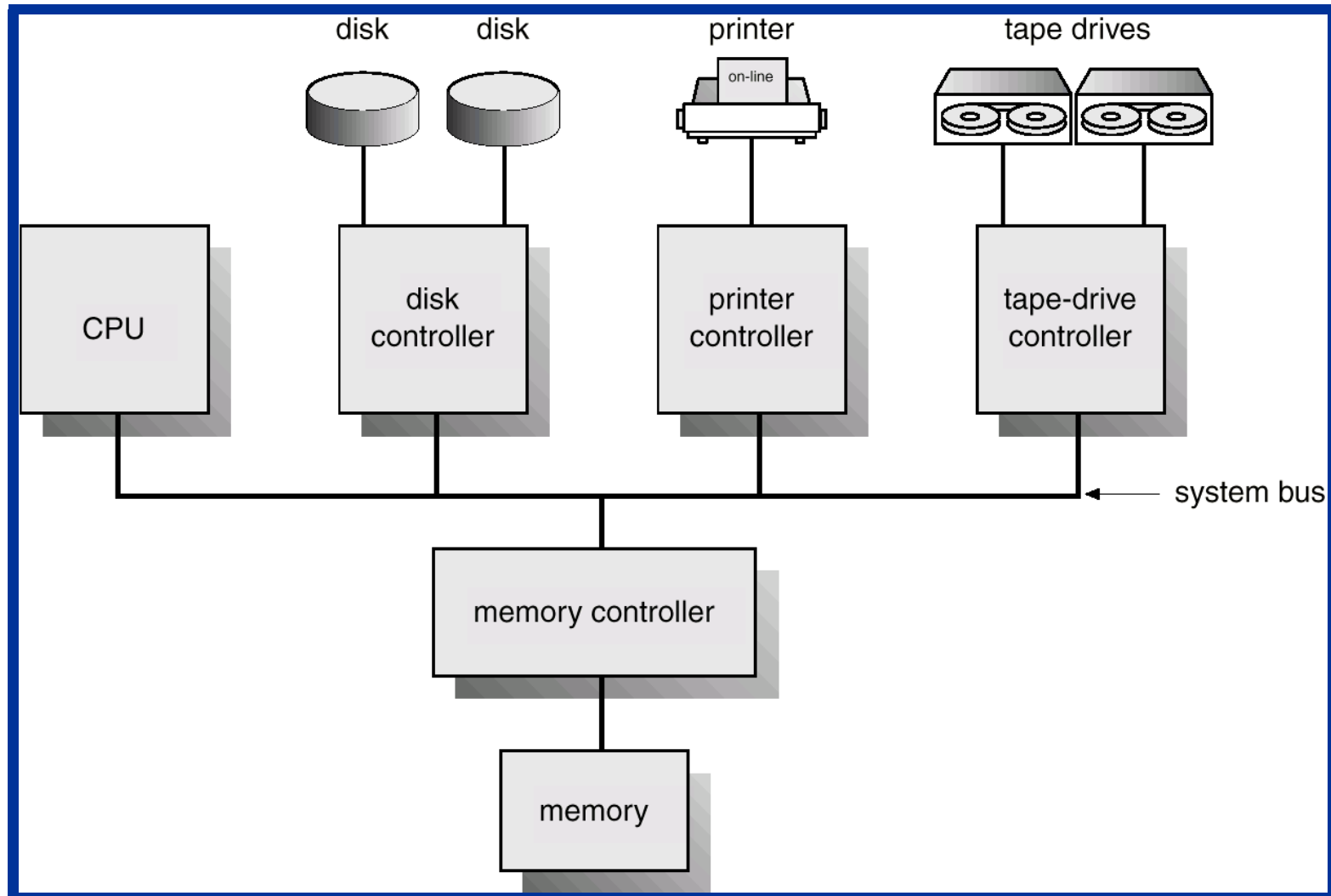
# Sistemas de tarjetas inteligentes

- Memoria limitada
- (A menudo) funcionalmente limitados
- Sistemas basado en Java con JVM
- Ganancia de importancia
  - Teléfonos móviles
  - Cuestiones de seguridad (p.e. autenticación)

# Índice

- Definición de SO
- Historia
- Vista de componentes
- Diferentes tipos de SO
  - Mainframe
  - Escritorio
  - Multiprocesador
  - Distribuido
  - En clúster
  - Tiempo real
  - Manejable
  - Tarjeta inteligente
- **Estructura de un Sist. Informático**
  - **Operación de un Sist. Informático**
  - **Estructura E/S**
  - **Estructura de almacenamiento**
  - **Estructura jerárquica**
  - **Protección del Hardware**
- Estructura de un Sist. Operativo
  - Componentes del sistema
    - Administración de procesos
    - Administración de Memoria Principal
    - Administración de ficheros
    - Administración del Sistema de E/S
    - Administración secundaria
    - Conexión de red
    - Sistema de Protección
    - Sistema de intérprete de comandos
  - Llamadas a sistema
  - Diseño del sistema
    - Monolítico
    - Por capas
    - Virtuales (máquinas)
- Tareas de un SO

# Estructura de un Sistema Informático



# Dispositivos en operación

- Los dispositivos de E/S y la CPU pueden ejecutarse concurrentemente
- Cada dispositivo controlador está a carga de un tipo de dispositivo particular
- Cada dispositivo controlador tiene un buffer local
- La CPU mueve datos desde/a la memoria principal a/desde los buffers locales
- La E/S va desde el dispositivo al buffer local del controlador
- El dispositivo controlador informa a la CPU que ha finalizado su operación causando una interrupción.
- Observación: el funcionamiento de un sistema operativo descansa sobre el mecanismo de interrupciones

# Estructura E/S

- Después de que empiece la E/S, el control vuelve al programa de usuario sólo hasta completar la E/S
  - Instrucción de espera que mantiene ociosa la CPU hasta la próxima interrupción
  - Bucle de espera (competición por acceso a memoria)
  - Como mucho hay una petición de E/S pendiente al mismo tiempo, no hay procesado simultáneo de E/S
- Después de que empiece la E/S, el control vuelve al programa de usuario sin esperar que se complete la E/S
  - *Llamada al sistema* – petición al sistema operativo para permitir al usuario esperar la terminación de E/S
  - *Tabla de estados de dispositivos* - contiene entradas para cada dispositivo de E/S indicando su tipo, dirección, y estado
  - El sistema operativo indexa en la tabla de dispositivos de E/S para determinar el estado del dispositivo y modificar la entrada en la tabla para incluir la interrupción

# Acceso Directo a Memoria

- Usado para dispositivos E/S de gran velocidad
- Bloques transferidos desde el buffer de almacenamiento a la memoria principal sin intervención de la CPU
- Se genera una interrupción por bloque (en vez de una interrupción por byte)

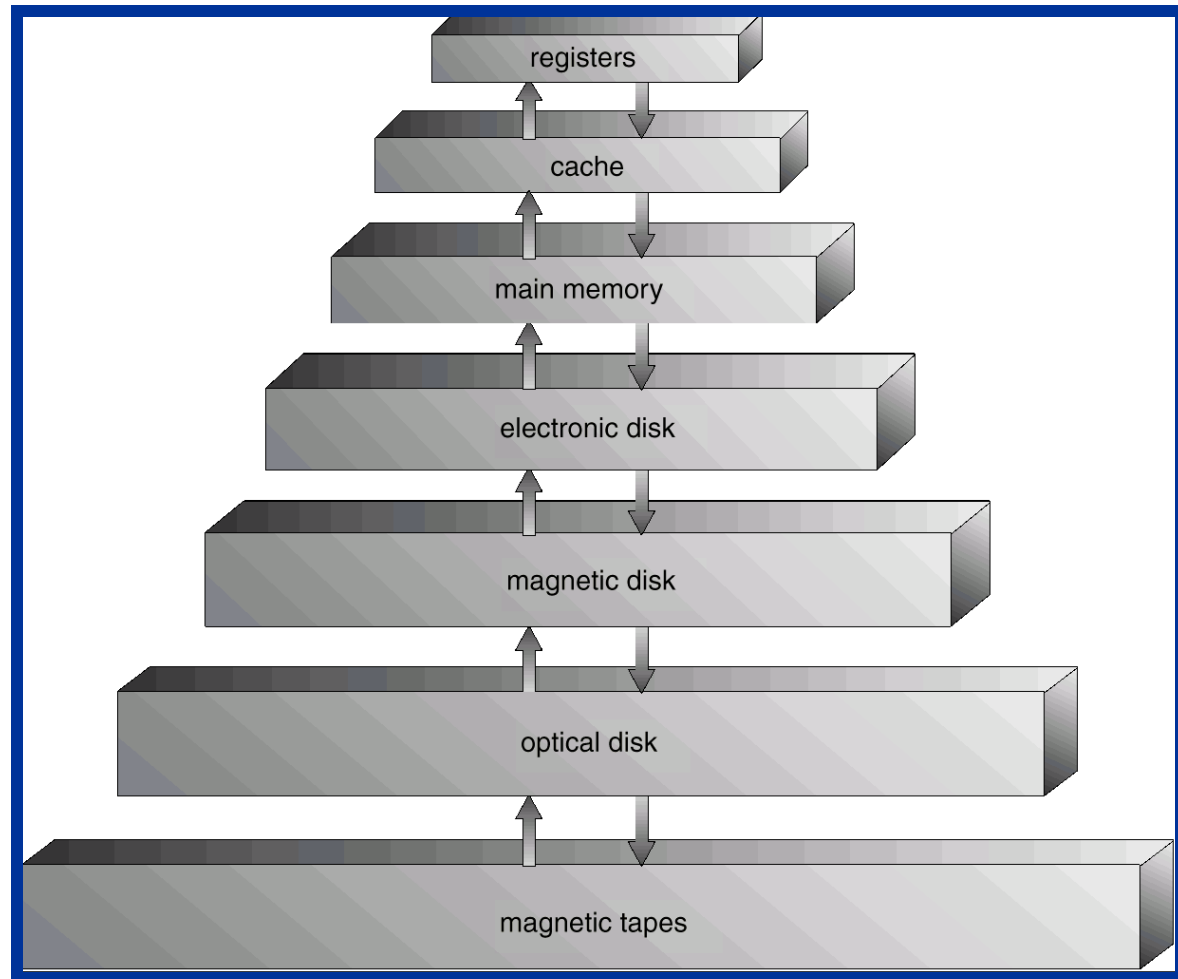


# Estructura de almacenamiento

- Memoria principal
  - El único medio grande de almacenamiento al que la CPU puede tener acceso directamente.
- Memoria secundaria
  - Extension de la memoria principal que proporciona una gran capacidad de almacenamiento no volátil
- Discos magnéticos
  - Metal rígido o platos de cristal cubiertos con material de grabación magnético
    - La superficie del disco está lógicamente dividida en *pistas*, que se subdividen en *sectores*.
    - El *controlador de disco* determina la interacción lógica entre el dispositivo y el ordenador.

# Jerarquía de la estructura de almacenamiento

- Los sistemas de almacenamiento se organizan en una jerarquía:
  - Velocidad
  - Coste
  - Volatilidad



# Protección hardware

- Operación en modo dual
  1. *Mode de usuario* – se ejecuta un programa de usuario
  2. *Modo monitor* (kernel o *sistema*) – se ejecuta el sistema operativo.
- Protección de E/S
  - Todas las instrucciones E/S son instrucciones privilegiadas.
  - Deben asegurar que un programa de usuario nunca podría ganar el control del ordenador en modo monitor
- Protección de memoria
  - **Registro base** – mantiene la dirección de memoria física legal más pequeña.
  - **Registro límite** – contiene el tamaño del rango
- *Temporizador* – interrumpe al ordenador después de un período especificado para asegurar que el sistema operativo mantiene el control.
- El temporizador se utiliza para implementar multitarea y para calcular el tiempo de ejecución

# Índice

- Definición de SO
- Historia
- Vista de componentes
- Diferentes tipos de SO
  - Mainframe
  - Escritorio
  - Multiprocesador
  - Distribuido
  - En clúster
  - Tiempo real
  - Manejable
  - Tarjeta inteligente
- Estructura de un Sist. Informático
  - Operación de un Sist. Informático
  - Estructura E/S
  - Estructura de almacenamiento
  - Estructura jerárquica
  - Protección del Hardware
- **Estructura de un Sist. Operativo**
  - **Componentes del sistema**
    - Administración de procesos
    - Administración de Memoria Principal
    - Administración de ficheros
    - Administración del Sistema de E/S
    - Administración secundaria
    - Conexión de red
    - Sistema de Protección
    - Sistema de intérprete de comandos
  - **Llamadas a sistema**
  - Diseño del sistema
    - Monolítico
    - Por capas
    - Virtuales (máquinas)
- Tareas de un SO

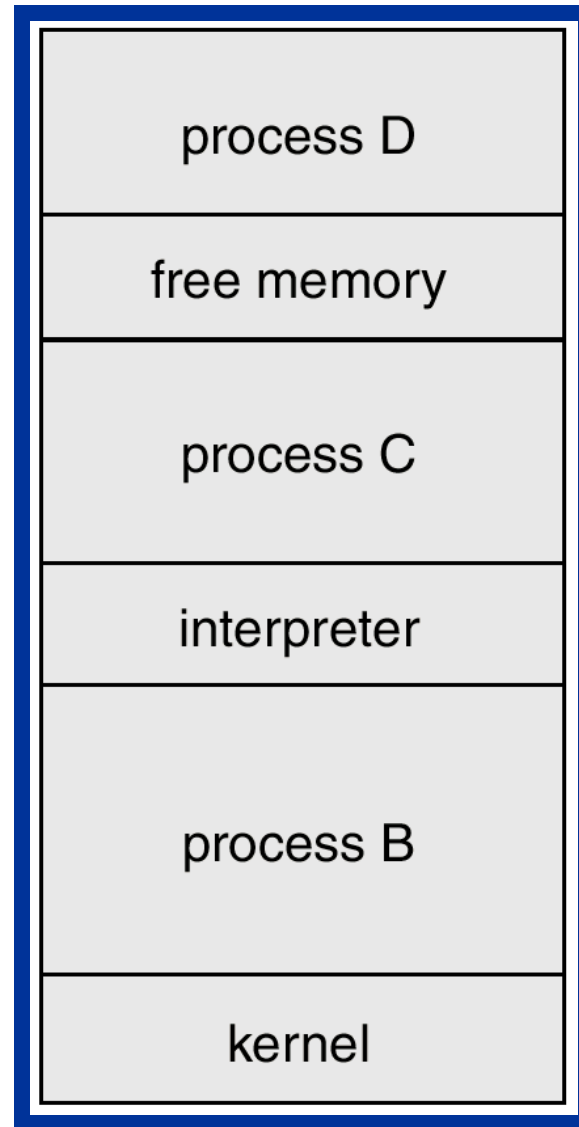
# Componentes del sistema: Administración de procesos

- Un *proceso* es un programa en ejecución. Un proceso necesita de ciertos recursos, incluyendo tiempo de CPU, memoria, ficheros, y dispositivos de E/S, para cumplir su tarea.
- El sistema operativo es responsable de las siguientes actividades relativas a la administración de procesos.
  - Creación y borrado de procesos.
  - Detención y recuperación de procesos.
  - Ofrecer mecanismos para:
    - Sincronización de procesos
    - Comunicación entre procesos

# Administración de memoria principal

- La memoria es una gran tabla de palabras o bytes, cada una con su propia dirección.
- El sistemas operativo es responsable de:
  - Saber qué partes de la memoria están siendo actualmente usadas y por quién
  - Decidir qué procesos deben estar disponibles en memoria
  - Reservar y liberar espacio de memoria según sea necesario

# Ejemplo: UNIX corriendo varios programas



# Administración de ficheros

- Un fichero es una colección de información relacionada definida por su creador (p.e. programas y datos)
- El sistema operativo es responsable de:
  - Creación y borrado de ficheros.
  - Creación y borrado de directorios.
  - Soporte de primitivas para manipular ficheros y directorios.
  - Mapear ficheros en memoria secundaria.
  - Copia de seguridad del fichero en un medio de almacenamiento estable (no volátil).



# Administración del sistema de E/S

- El sistema de E/S consiste en:
  - Un sistema de buffer y caché
  - Una interfaz general para los drivers de los dispositivos
  - Drivers para dispositivos hardware específicos

# Memoria de almacenamiento secundaria

- El ordenador debe proporcionar un *almacenamiento secundario* (p.e. disco) como soporte de apoyo a la memoria principal
- El sistema operativo es responsable de las siguientes actividades en conexión con el administrador de disco:
  - Administración del espacio libre
  - Elección del espacio a utilizar para almacenar datos
  - Planificación de operaciones en disco

# Redes (Sistemas distribuidos)

- Un sistema *distribuido* es una colección de procesadores que no comparten una memoria o un reloj. Cada procesador tiene su propia memoria local.
- Los procesadores en el sistema están conectados a través de una red de comunicaciones.
- La comunicación tiene lugar usando un *protocolo*.
- Un sistema distribuido proporciona al usuario acceso a varios recursos del sistema.
- El acceso a recursos compartidos permite:
  - Aumentar la velocidad de cálculo
  - Incrementar la disponibilidad de datos
  - Mejorar la fiabilidad

# Sistema de protección

- La *protección* se refiere al mecanismo para controlar el acceso de programas, procesos, o usuarios tanto a los recursos del sistema como del usuario.
- El mecanismo de protección debe:
  - Distinguir entre uso autorizado y desautorizado.
  - Especificar los controles que se deben imponer.
  - Proporcionar un medio de ejecución.

# Sistema de Intérprete de comandos

- Muchos comandos se dan al sistema operativo mediante sentencias de control que se ocupan de:
  - creación y administración de procesos
  - gestión de E/S
  - administración de memoria secundaria
  - administración de memoria principal
  - acceso al sistema de ficheros
  - protección
  - acceso a la red

# Llamadas al sistema

- Las llamadas al sistema proporcionan el interfaz entre el programa que está corriendo y el sistema operativo.
  - Generalmente disponibles como instrucciones en lenguaje ensamblador.
  - Se definen lenguajes para reemplazar al lenguaje ensamblador para programación de sistemas que permiten que las llamadas se hagan directamente (p.e. C, C++)
- Tres métodos para pasar parámetros entre el programa y el sistema operativo.
  - Los parámetros se pasan en *registros*.
  - Se almacenan los parámetros en una tabla en memoria, y la dirección de la tabla se pasa como parámetro en un registro.
  - Hace un *push* (guarda) de los parámetros en la *pila* del programa, y hace un *pop* (saca) de la pila del sistema operativo.

# Tipos de llamadas al sistema

- Control de procesos
- Administración de ficheros
- Administración de dispositivos
- Mantenimiento de information
- Comunicaciones

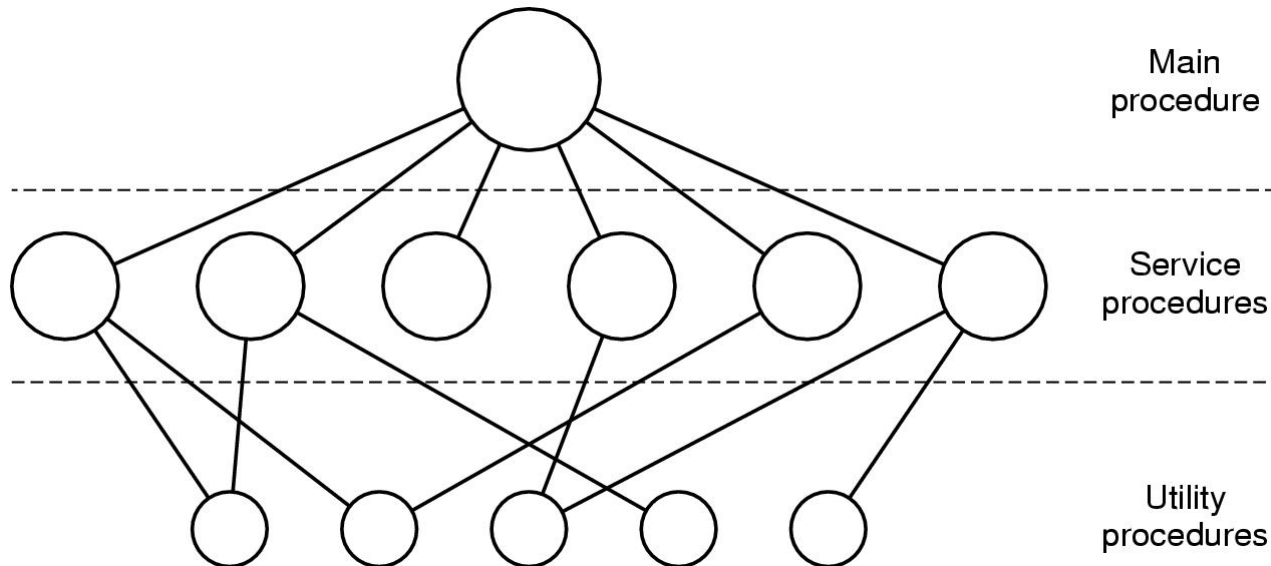
# Índice

- Definición de SO
- Historia
- Vista de componentes
- Diferentes tipos de SO
  - Mainframe
  - Escritorio
  - Multiprocesador
  - Distribuido
  - En clúster
  - Tiempo real
  - Manejable
  - Tarjeta inteligente
- Estructura de un Sist. Informático
  - Operación de un Sist. Informático
  - Estructura E/S
  - Estructura de almacenamiento
  - Estructura jerárquica
  - Protección del Hardware
- Estructura de un Sist. Operativo
  - Componentes del sistema
    - Administración de procesos
    - Administración de Memoria Principal
    - Administración de ficheros
    - Administración del Sistema de E/S
    - Administración secundaria
    - Conexión de red
    - Sistema de Protección
    - Sistema de intérprete de comandos
  - Llamadas a sistema
  - **Diseño del sistema**
    - **Monolítico**
    - **Por capas**
    - **Virtuales (máquinas)**
- **Tareas de un SO**



# Estructura de un SO: sistemas monolíticos

- Fichero objeto compilado para un bloque
- Cada procedimiento puede ver (llamar) a todos los otros
- Se proporcionan diferentes modos
  - Modo de usuario
  - Modo de kernel



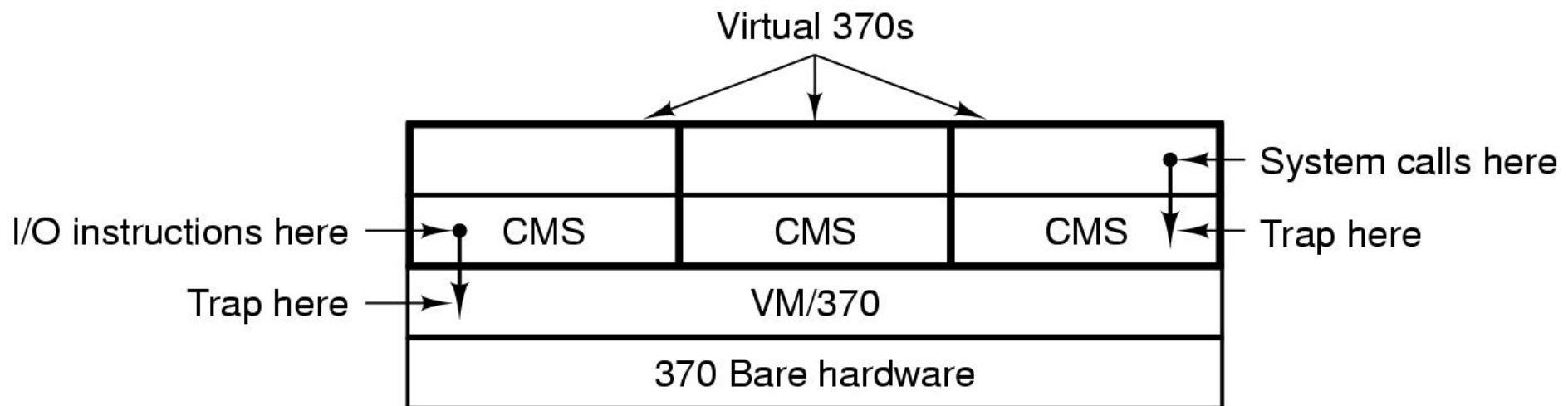
# Estructura de un SO: sistemas en capas

- Jerarquía de capas relacionadas
- Ejemplo: El sistema

Layer	Function
5	The operator
4	User programs
3	Input/output management
2	Operator-process communication
1	Memory and drum management
0	Processor allocation and multiprogramming

# Estructura de un SO: máquinas virtuales

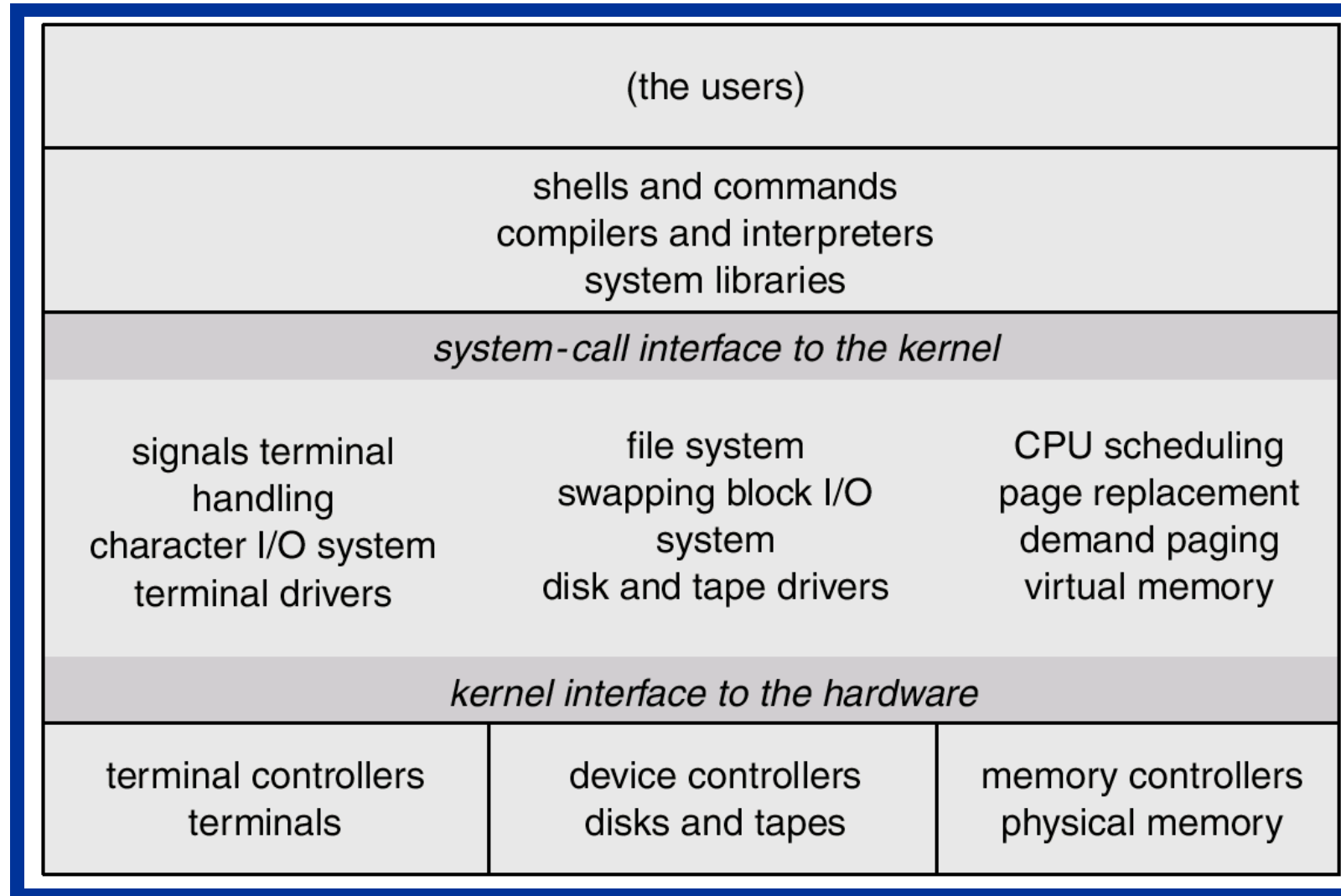
- Reconstruyen el hardware subyacente
- Ejemplo: IBM VM/370
  - Monitor de máquina virtual corriendo en el hardware subyacente y administrando la multiprogramación
  - Proporciona varias máquinas virtuales en a higher level as exact copies of the hardware



# Ejemplo: estructura de un sistema UNIX

- UNIX – limitado por la funcionalidad del hardware, el sistema operativo UNIX original tenía una estructura limitada. El SO UNIX consta de dos partes separables.
  - Programas de sistema
  - El kernel
    - Consta de todo lo que está por debajo de la interfaz de llamadas al sistema y por encima del hardware físico
    - Proporciona el sistema de ficheros, el planificador de CPU, administración de memoria, y otras funciones del sistema operativo; un gran número de funciones para un solo nivel.

# Estructura del sistema UNIX



# Principales tareas del SO

- Manejo de interrupciones
  - Identificación de la señal y manejo
- Dispatching
  - Conmutación de procesos
- Administración de recursos
  - Ubicación, disposición y administración de recursos
- Ubicación de programa
  - Enlazado, carga y eliminación a/desde la memoria principal
- Administración de ficheros
  - Organización, ubicación, modificación, reutilización
- Control de tareas
  - Planificación
- Fiabilidad
  - Robustez, tolerancia y corrección