2: Estructuras de un S.O.

Sistemas Operativos 1
Ing. Alejandro León Liu

- Servicios que provee un S.O. a usuarios y programas.
- Estructura de un S.O.
- Configuración e instalación de un S.O.

SERVICIOS

- S.O. Provee servicios a:
  - Usuarios
  - Programas
    - Ambiente ejecución de programas.
    - Facilitar programación

Servicios proveídos a usuarios

- Interfaz de usuario
  - Línea de comando
  - Batch (.bat)
  - Interfaz gráfica
- Ejecución de programas
- I/O
  - No accedido directamente por usuarios
    - Eficiencia
    - Protección

Servicios proveídos a usuarios

- File System
  - Archivos / Carpetas
  - Permisos
- Comunicación
- Detección de errores

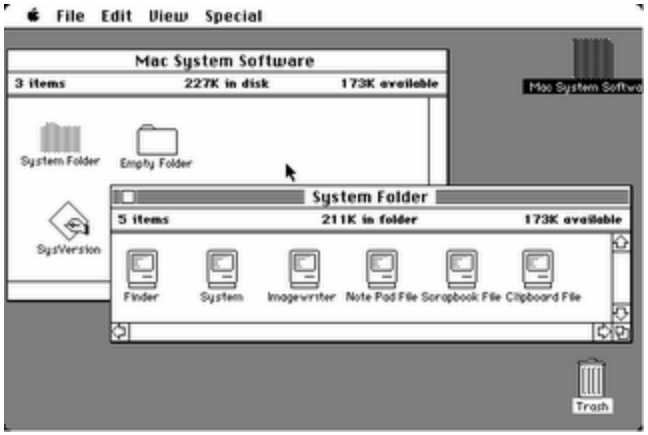
Servicios que proveen eficiencia

- Asignación de recursos
  - CPU
  - Memoria
  - Archivos
  - I/O
- Contabilizar recursos
  - Registro de recursos asignados
  - Estadísticas
- Protección y seguridad

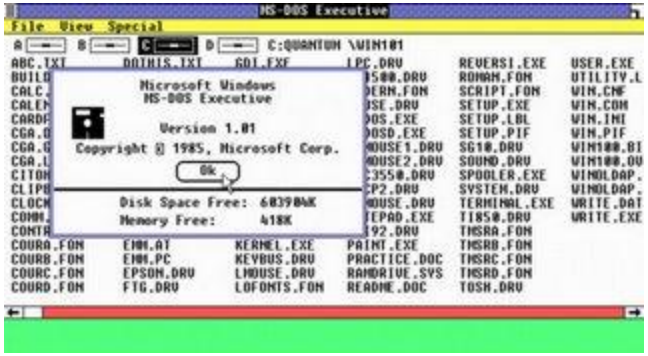
INTERFAZ DE USUARIO

- Línea de comando
  - Shell
  - Obtener y ejecutar siguiente comando
  - Dos implementaciones:
    - Ejecuta comandos internos (MS-DOS)
    - Invoca programas del sistema (Linux Shell)
- Interfaz gráfica
  - Linux: CDE (Common desktop environment), X, KDE, GNOME desktop.
  - Mac OS: no proveía GUI hasta Mac OS X.
  - Windows: MS-DOS shell

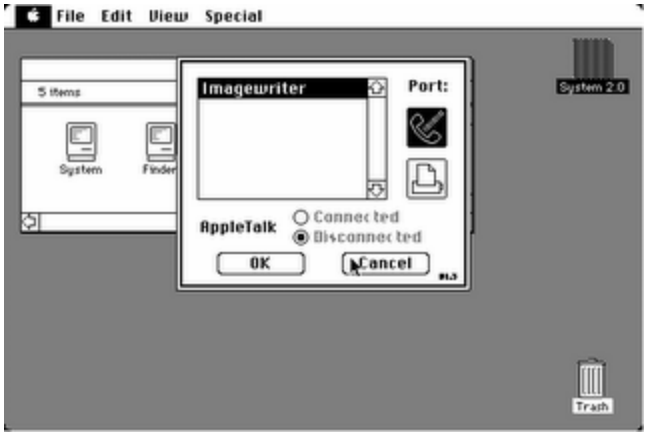
- Macintosh System 1



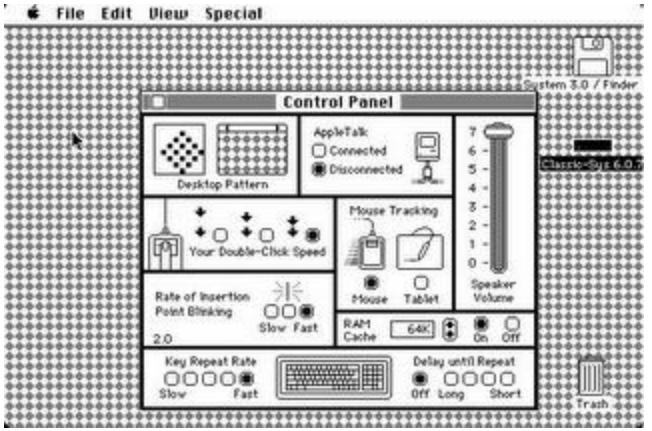
- Microsoft Windows 1.0



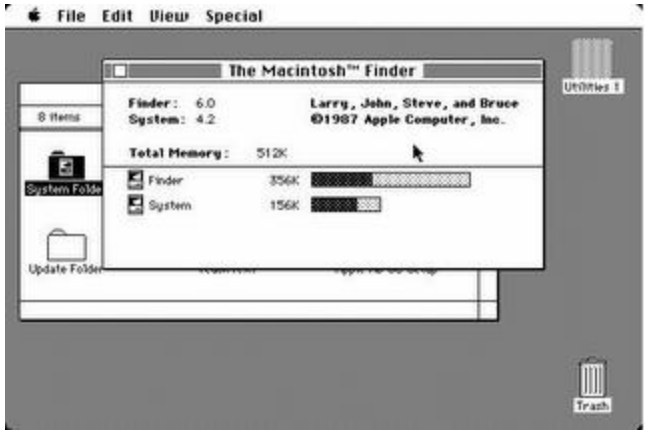
- Macintosh System 2



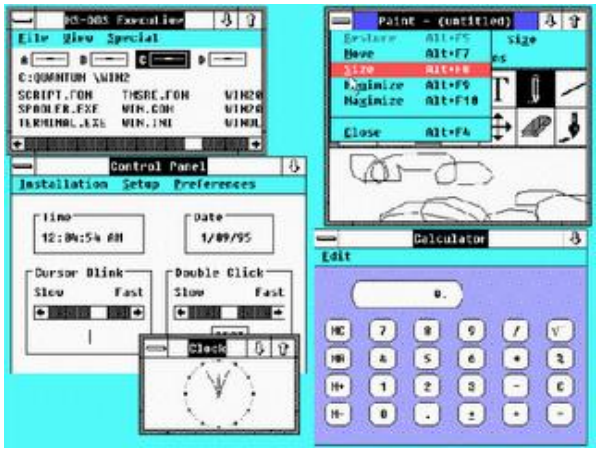
- Macintosh System 3



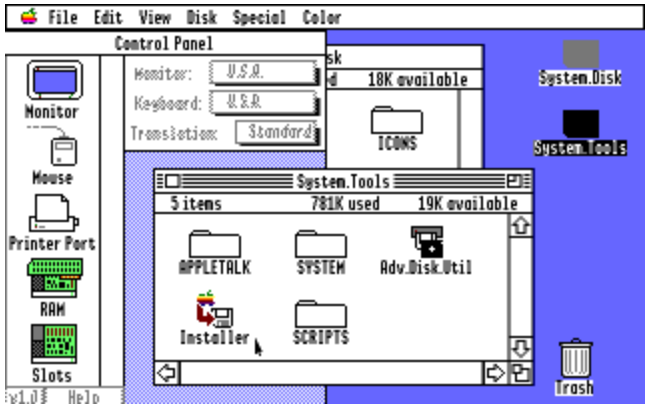
- Macintosh System 4



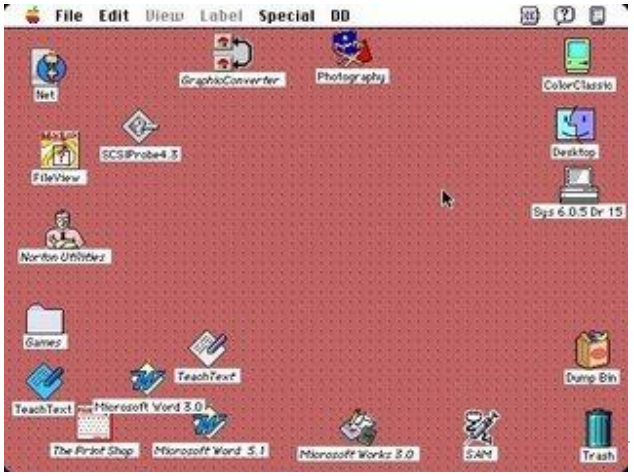
- Microsoft Windows 2.0



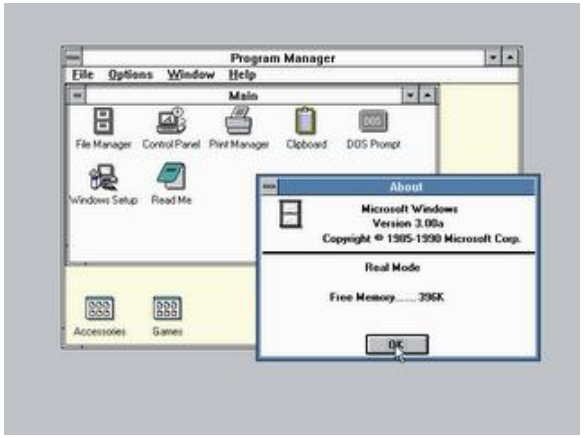
- Macintosh System 5



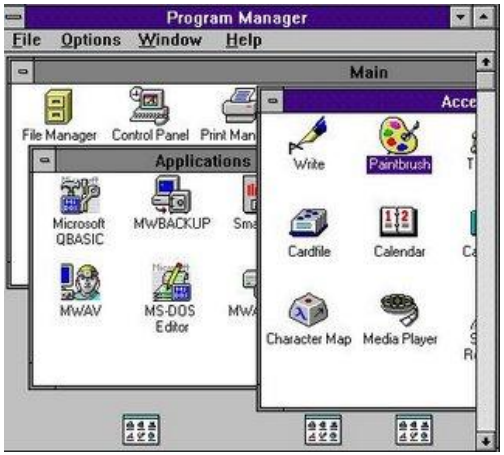
- Macintosh System 6



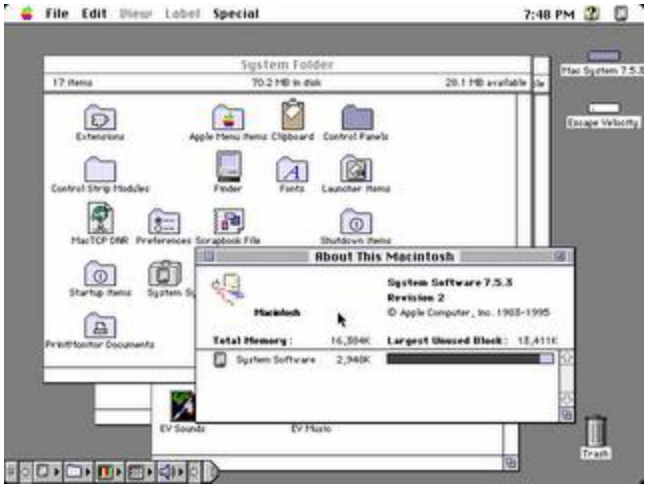
- Microsoft Windows 3.0



- Microsoft Windows 3.1

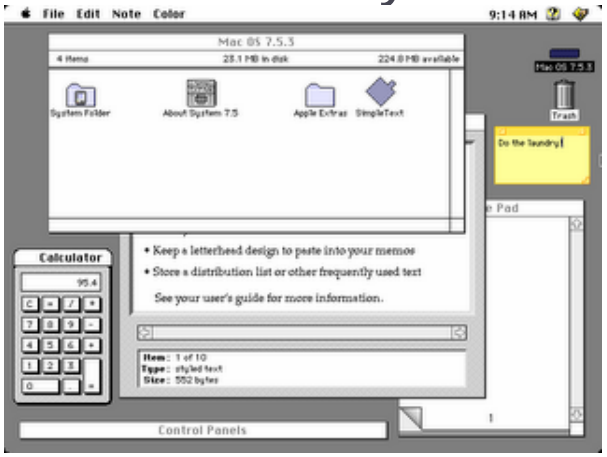


- Macintosh System 7

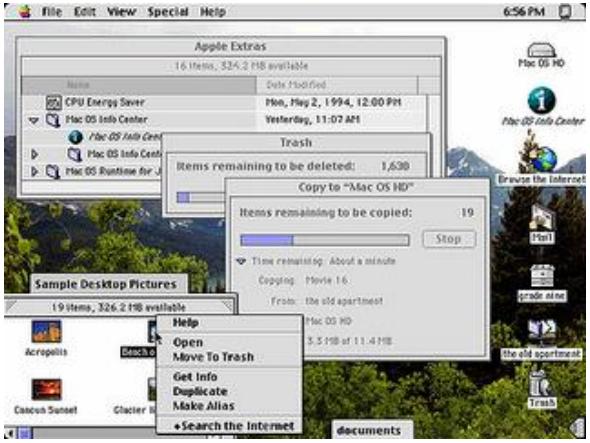


- Macintosh System 7.1

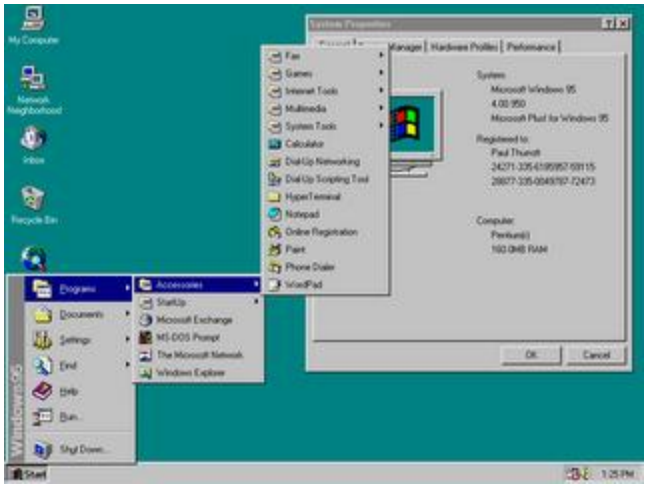
- Macintosh System 7.5



- Mac OS 8



- Microsoft Windows 95



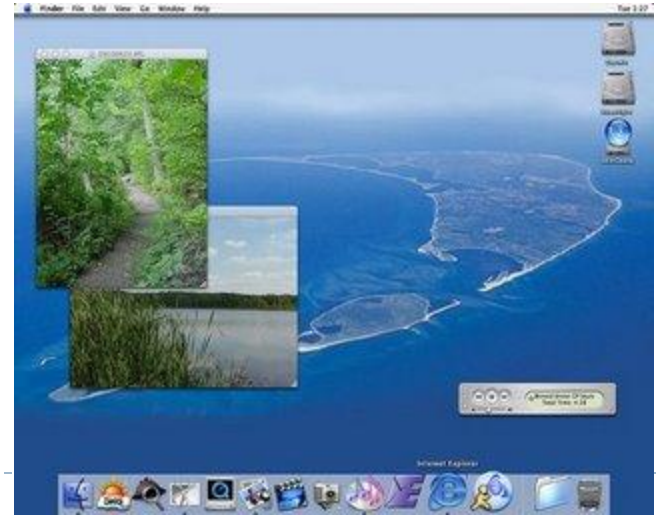
- Microsoft Windows 98

- KDE 1.0

- KDE 2.0

- Macintosh 9

- Mac OS X 10.1



- Microsoft Windows XP

- Mac OS X 10.5 (Leopard)

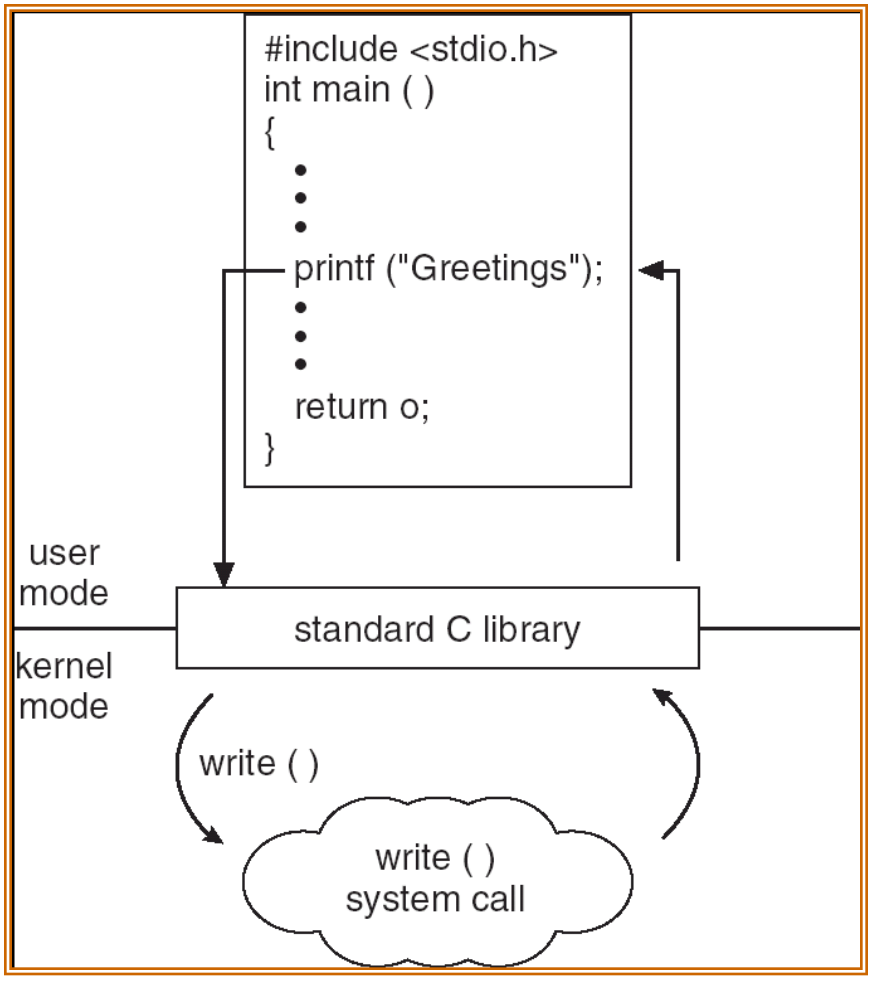
- KDE 3.5

- Microsoft Windows Vista

SYSTEM CALLS

- Interfaz a los servicios del S.O.
- C++, C, Assembler
- API (Application programming interface)
- Tabla de interrupciones
  - Tabla de bifurcación
- Paso de parámetros:
  - Registros
  - Bloque de memoria
  - Pila

SYSTEM CALLS



TIPOS DE SYSTEM CALLS

- Control de procesos
  - Terminar (con o sin error)
  - Crear / terminar procesos
  - Asignar y liberar memoria
  - Halt (esperar tiempo)
  - Esperar o crear evento
  - Get / Set attributes
  - Debug
    - trace
    - Memory dump
    - CPU: Single step mode

TIPOS DE SYSTEM CALLS

- Manejo de archivos
  - Crear / Borrar / Modificar
  - Abrir / Cerrar
  - Leer / Escribir / Reposicionar
- I/O
  - Solicitar / Liberar dispositivos
  - Leer / Escribir / Reposicionar (Similar a archivos)
- Informativas
  - Get / Set process attribute
  - Date & Time

TIPOS DE SYSTEM CALLS

- Comunicación entre procesos
  - Mensajes
    - Útil para poca información
    - Fácil implementación
  - Memoria compartida
    - Mayor velocidad
    - Problemas de protección
    - Problemas de sincronización
  - …. ¿Similar a Polling vrs. interrupts?

SYSTEM PROGRAMS

- Manejo de archivos
  - Ej: Finder
- Status / Information
  - Ej: Registry, task manager.
- Modificación de archivos
  - Ej: Editores de texto, grep.
- Soporte a desarrolladores
  - Ej: Compiladores, ensambladores, debuggers, intérpretes.
- Ejecución de programas
  - Ej: linkers, loaders.
- Comunicación
  - Ej: Remote desktop, browsers.

DISEÑO E IMPLEMENTACIÓN DE UN S.O.

- Definir metas y especificaciones
  - Tipo de sistema:
    - Tiempo compartido, multiusuario, distribuido etc…
  - Punto de vista usuario
    - Fácil de usar, conveniente, seguro, rápido, etc…
  - Punto de vista desarrolladores
    - Flexible, confiable, sin errores, eficiente, etc…
  - Las anteriores son especificaciones vagas

- Políticas (qué) y Mecanismo (cómo)
  - Políticas pueden cambiar
    - Peor de los casos: cambio en mecanismo
  - Separación -> Flexibilidad
  - UNIX
    - En un inicio, calendarizador de tiempo compartido
    - Cargar tablas para calendarización
    - Crear system calls
  - Windows & Mac OS
    - Políticas y mecanismo hardcoded

- Implementación
  - Lenguaje ensamblador
    - Rápido
    - Menor espacio
  - Lenguaje de alto nivel (c,c++)
    - Fácil programación
    - Código compacto
    - Fácil entender y depurar
    - Portable
      - Ej: Linux (C) vrs. MS-DOS (Assembler intel)
    - Optimización de compiladores

ESTRUCTURA DE UN S.O.

- Divide & Conquer
- Estructura simple
  - MS-DOS
    - Programas pueden acceder a I/O básico
    - Vulnerable o propenso a error
    - No había
      - Hardware protection (memory, cpu, i/o)
      - Dual mode
  - UNIX
    - Kernel bastante grande

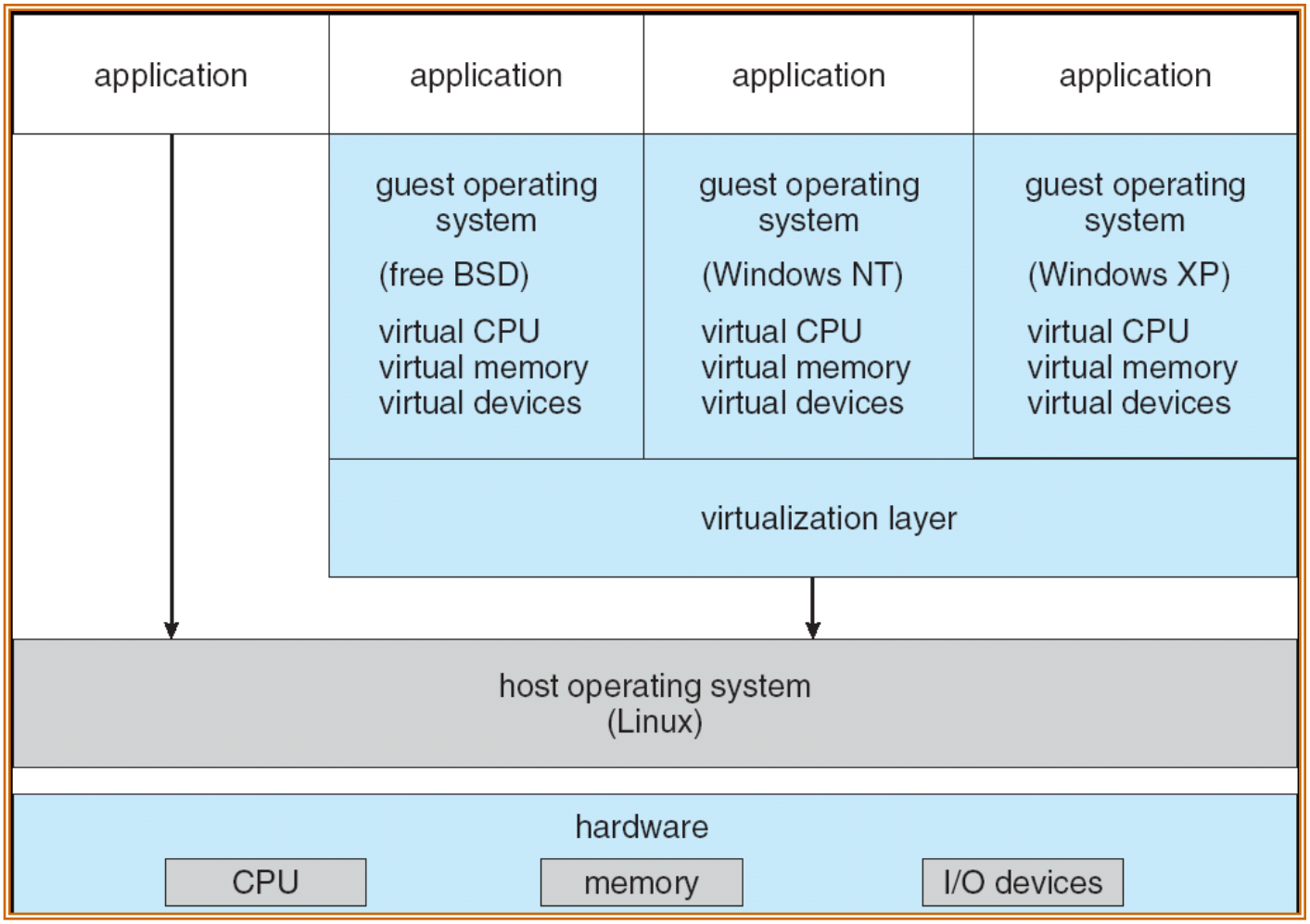
- Estructura de capas
  - Modularidad
  - Capa 0: Hardware
  - Capa n: GUI
  - Ventajas
    - Fácil implementación (abstracción)
    - Facilita pruebas y depuración
  - Capas utilizan capas inferiores
    - Difícil definir capas
    - Varias capas: ineficiente, modificación de parámetros

- Microkernels
  - Quitar funcionalidad a kernel
  - Agregar como system o user programs
  - Poca funcionalidad
    - Manejo de procesos
    - Manejo de memoria
    - Comunicación (Mensajes)
  - Facilidad de extender
    - Portable
  - Seguridad y confiabilidad
    - Servicios en modo usuario
  - Bajo rendimiento

- Módulos
  - Programación orientada a objetos
  - Enlazar dinámicamente otros módulos
  - Combina lo mejor de:
    - Estructura de capas
      - Secciones definidas que se interfazan
    - Microkernel
      - Módulo principal contiene funciones básicas
      - No utiliza mensajes para comunicación
  - Solaris, Linux Mac OS X

MÁQUINAS VIRTUALES

- Interfaz idéntica al hardware
- Proveer diferentes ambientes de ejecución
- Beneficios
  - Compartir hardware
  - Diferentes S.O. concurrentemente
  - No hay problemas de protección
  - Research and Development de S.O.
  - Facilidad de instalación
    - Snapshots


application
application
application
application
guest operating system
(free BSD)
virtual CPU
virtual memory
virtual devices
guest operating system
(Windows NT)
virtual CPU
virtual memory
virtual devices
guest operating system
(Windows XP)
virtual CPU
virtual memory
virtual devices
virtualization layer
host operating system
(Linux)
hardware
CPU
memory
I/O devices

- Java Virtual Machine
  - Bytecode (Independiente de máquina)
  - Java class loader
  - Java interpreter
  - JVM implementado en
    - Software sobre S.O. (Windows, Linux, Mac OS, etc…)
    - Java chip
- .NET Framework
  - Biblioteca de clases & Ambiente de ejecución
  - Múltiples lenguajes de programación
  - Just in time compiler (JIT)

GENERACIÓN DE S.O.

- Configuración de un S.O. en una computadora específica.
- Información que varía:
  - CPU
  - Memoria
  - Dispositivos I/O
  - Opciones del S.O.
    - Tipo de calendarizador de CPU
    - Número máximo de procesos
    - Tamaño de buffers
- Incluir o enlazar drivers y módulos
- Ciertos S.O. requieren volver a compilar el sistema