

Grupo ARCOS  
Departamento de Informática  
Universidad Carlos III de Madrid

# Lección 1 (a)

## Introducción

Diseño de Sistemas Operativos  
Grado en Ingeniería Informática y Doble Grado I.I. y A.D.E.



# Objetivos generales

1. Conocer **qué es** un sistema operativo.
  1. Software complejo y crítico.  
Precisa gran capacidad de adaptación.
2. Conocer **cómo es por dentro**.
  1. Objetivos y estructura

# A recordar...

---

Antes de clase

Clase

Después de clase

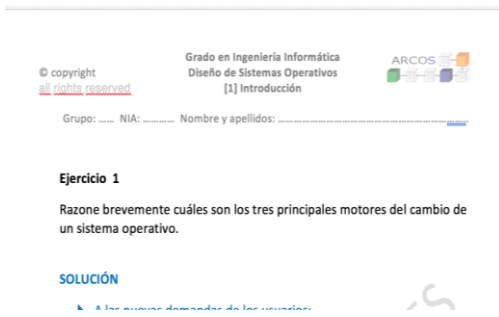

Preparar los pre-requisitos.

Estudiar el material asociado a la **bibliografía**:  
las transparencias solo no son suficiente.  
Preguntar dudas (especialmente tras estudio).

Ejercitar las competencias:

- ▶ Realizar todos los **ejercicios**.
- ▶ Realizar los **cuadernos de laboratorios** y las **prácticas** de forma progresiva.

# Ejercicios, cuadernos de prácticas y prácticas

Ejercicios ✓	Cuadernos de prácticas ✓	Prácticas ✗
 <p>© copyright <a href="#">all rights reserved</a></p> <p>Grado en Ingeniería Informática Diseño de Sistemas Operativos [1] Introducción</p> <p>ARCOS</p> <p>Grupo: ____ NIA: _____ Nombre y apellidos: _____</p> <p><b>Ejercicio 1</b></p> <p>Razone brevemente cuáles son los tres principales motores del cambio de un sistema operativo.</p> <p><b>SOLUCIÓN</b></p> <p>▶ A las nuevas demandas de los usuarios</p>	 <p>DISEÑO DE SISTEMAS OPERATIVOS</p> <p>GRADO EN INGENIERÍA INFORMÁTICA DOBLE GRADO EN INGENIERÍA INFORMÁTICA Y ADMINISTRACIÓN DE EMPRESAS</p> <p>uc3m   Universidad Carlos III de Madrid</p> <p><b>Compilación y depuración de software orientado a eventos: descubriendo el <u>nanokernel</u></b></p>	

# Lecturas recomendadas

---

## Base



1. Carretero 2007:
  1. Cap. 2

## Recomendada



1. Tanenbaum 2006:
  1. Cap. I
2. Stallings 2005:
  1. Parte uno. Transfondo.
3. Silberschatz 2006:
  1. Cap. I

# Contenidos

---

1. **Qué es un sistema operativo.**
  1. Definiciones, principales funciones y características
2. **Cómo es por dentro.**
  1. Principales objetivos y estructura

# Contenidos

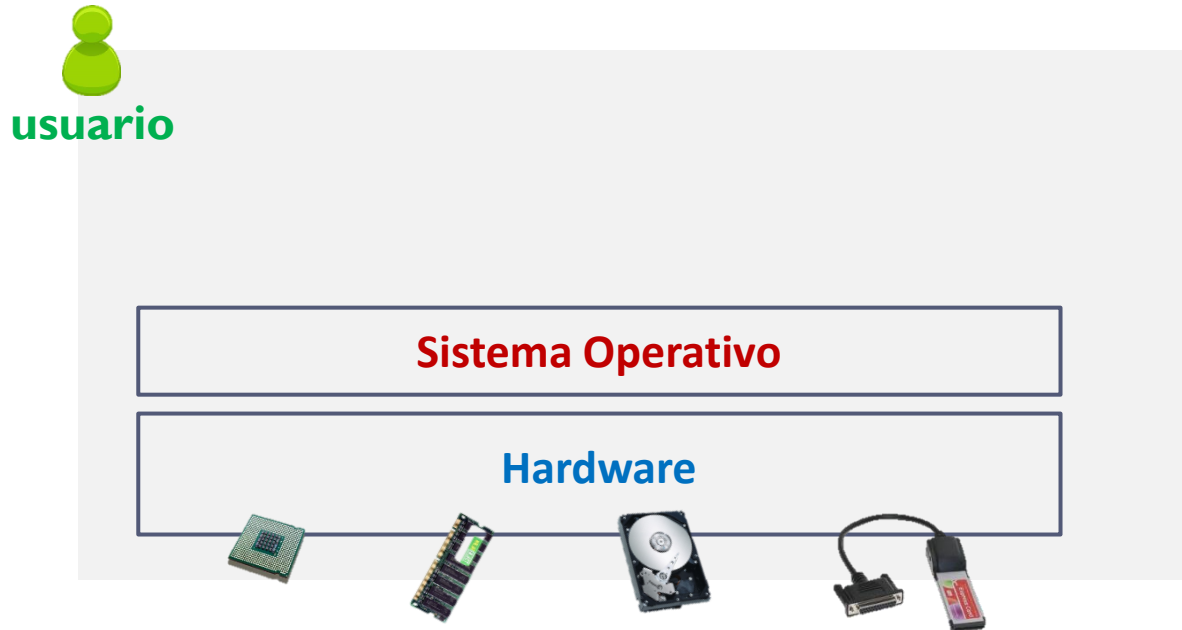
---

1. Qué es un sistema operativo.
  1. **Definiciones**, principales funciones y características
2. Cómo es por dentro.
  1. Principales objetivos y estructura

# Definición de sistema operativo

---

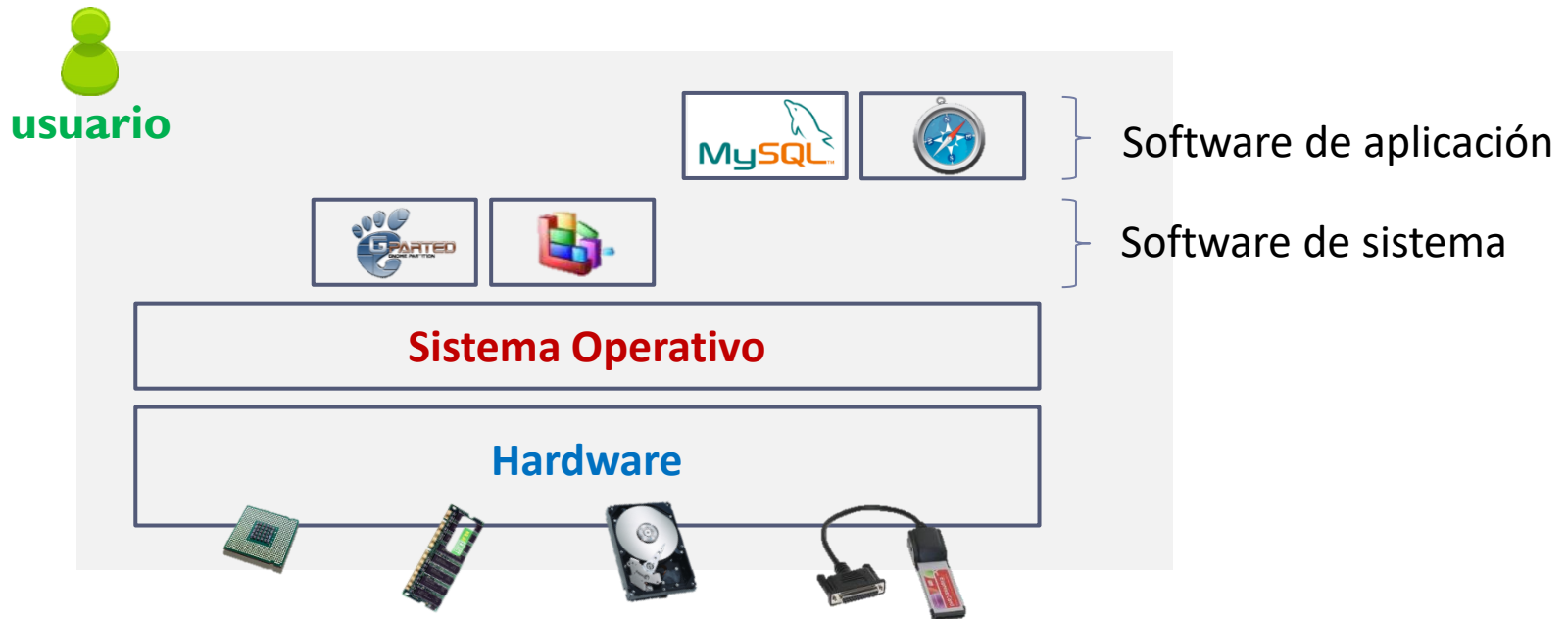
- ▶ **Sistema operativo:** software destinado a permitir la comunicación del **usuario** con un **ordenador** y gestionar sus recursos de manera cómoda y eficiente.





# Definición de sistema operativo

- ▶ **Sistema operativo:** software destinado a permitir la comunicación del **usuario** con un **ordenador** y gestionar sus recursos de manera cómoda y eficiente.



# Contenidos

---

1. Qué es un sistema operativo.
  1. Definiciones, **principales funciones** y características
2. Cómo es por dentro.
  1. Principales objetivos y estructura

# Funciones del sistema operativo

---



Usuario

Sistema operativo

- ▶ **Interfaz del usuario.**
- ▶ **Gestor de recursos:**
  - ▶ CPU, memoria, etc.
- ▶ **Máquina extendida:**
  - ▶ Servicios, interfaz del programador, etc.

Hardware

# Funciones del sistema operativo

---



Usuario

Sistema operativo

- ▶ Interfaz del usuario.
- ▶ Gestor de recursos:
  - ▶ CPU, memoria, etc.
- ▶ **Máquina extendida:**
  - ▶ Servicios, interfaz del programador, etc.

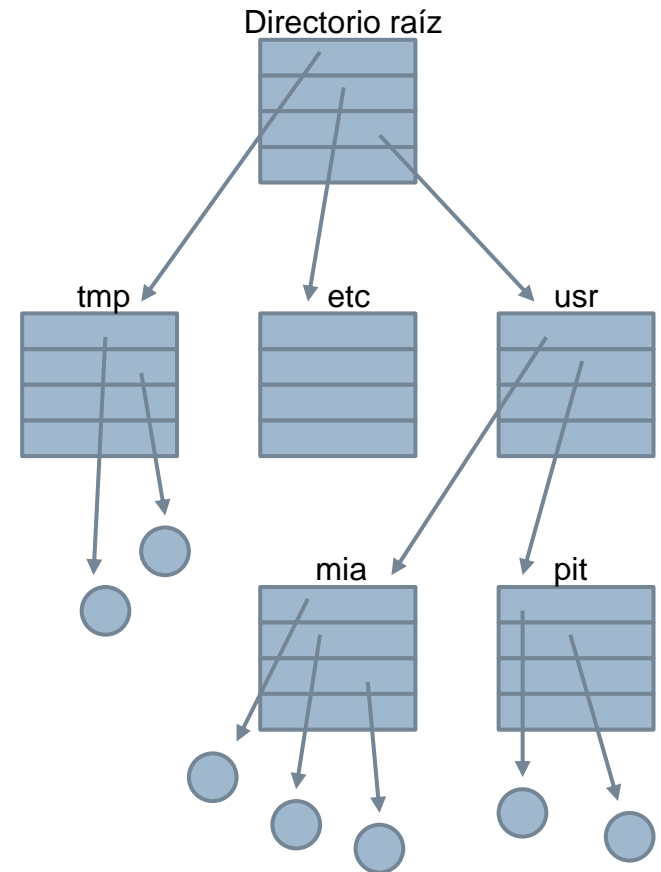
Hardware



# Abstracciones fundamentales

## Archivos

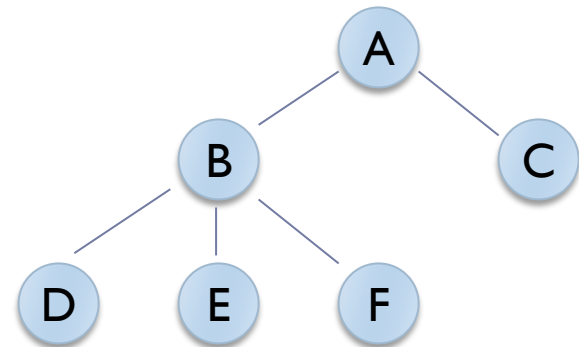
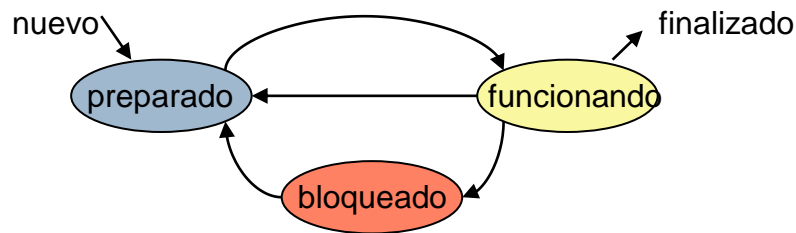
- ▶ Archivos y directorios
- ▶ Ruta, directorio de trabajo y raíz
- ▶ Protección
- ▶ Descriptor de archivo
- ▶ Archivos especiales:
  - ▶ Dispositivos E/S
    - ▶ E/S de bloque y de caracteres
  - ▶ Pipes
- ▶ Estándares entrada/salida/error



# Abstracciones fundamentales

## Procesos

- ▶ Procesos, tabla de procesos, árbol de procesos
- ▶ Imagen básica, planificación, señales
- ▶ Identificación de usuario y grupo
- ▶ Intérprete de mandatos (*shell*)



Árbol de procesos

# Funciones del sistema operativo

---



Usuario

Sistema operativo

- ▶ Interfaz del usuario.
- ▶ Gestor de recursos:
  - ▶ CPU, memoria, etc.
- ▶ Máquina extendida:
  - ▶ Servicios, interfaz del programador, etc.

Hardware



# Áreas de gestión

---

- ▶ Gestión de **Procesamiento**
  - ▶ Planificación
  - ▶ Prioridades, multiusuario
- ▶ Gestión de **Memoria**
  - ▶ Reparto de memoria entre procesos, con protección y compartición
- ▶ Gestión de **Almacenamiento**
  - ▶ Persistencia y etiquetación (ej. Sistema de Archivos) con una visión lógica unificada para usuarios y programas que sea independiente del medio físico
- ▶ Gestión de **Dispositivos**
  - ▶ Encubriendo las dependencias de hardware
  - ▶ Gestión de accesos concurrentes



# Funciones del sistema operativo

---



Usuario

Sistema operativo

- ▶ **Interfaz del usuario.**
- ▶ **Gestor de recursos:**
  - ▶ CPU, memoria, etc.
- ▶ **Máquina extendida:**
  - ▶ Servicios, interfaz del programador, etc.

Hardware

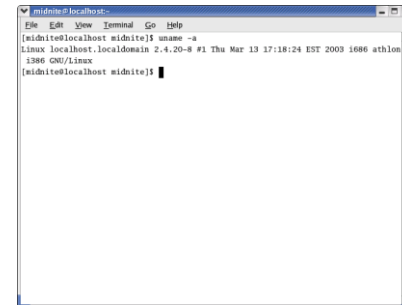


# Interfaz del usuario

- ▶ Interfaz del programador:
  - ▶ Mediante llamadas al sistema.

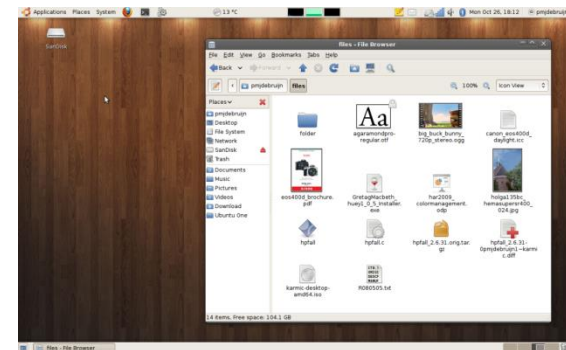
`ret = close (filedesc) ;`

- ▶ Interfaz de usuario:
  - ▶ Mediante línea de mandatos o CLI



```
midnite@localhost:~$  
File Edit View Terminal Go Help  
[midnite@localhost midnite]$ uname -a  
Linux localhost.localdomain 2.4.20-8 #1 Thu Mar 13 17:18:24 EST 2003 1686 athlon  
i386 GNU/Linux  
[midnite@localhost midnite]$
```

- ▶ Interfaz gráfica o GUI



# Funciones del sistema operativo

## resumen

---



Usuario

Sistema operativo

- ▶ Interfaz del usuario.
- ▶ Gestor de recursos:
  - ▶ CPU, memoria, etc.
- ▶ Máquina extendida:
  - ▶ Servicios, interfaz del programador, etc.

Hardware



# Un detalle más sobre máquina extendida...

---



Usuario

Sistema operativo

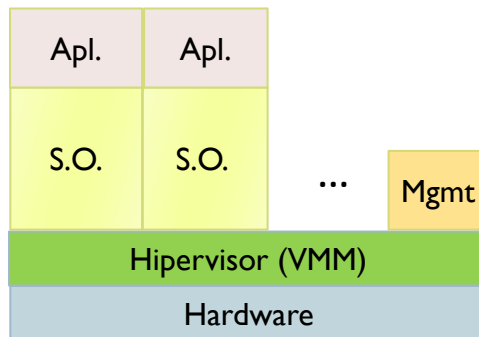
- ▶ Interfaz del usuario.
- ▶ Gestor de recursos:
  - ▶ CPU, memoria, etc.
- ▶ **Máquina extendida:**
  - ▶ Servicios, interfaz del programador, etc.

Hardware



# Máquinas virtuales

---



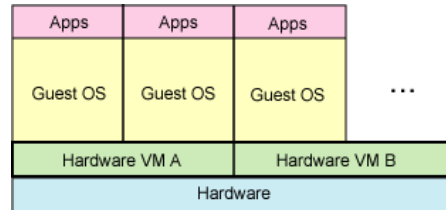
- ▶ El sistema operativo virtualiza ciertos elementos del hardware; ¿Por qué no **virtualizar todo**?
- ▶ **IBM** ha usado esta idea en sus mainframes desde principio de la década de los **70**.
- ▶ Un hipervisor virtualiza todo el ordenador, de manera que permite que múltiples sistemas operativos (o copias del mismo) ejecuten a la vez.
- ▶ La virtualización:
  - ▶ **[I]** supone **cierta sobrecarga**
  - ▶ **[V]** ofrece un **aislamiento excelente** entre sistemas y la flexibilidad en la reserva de recursos lo que **mejora el coste**, especialmente en «granjas»

# Máquinas virtuales

---

Distinto Hardware

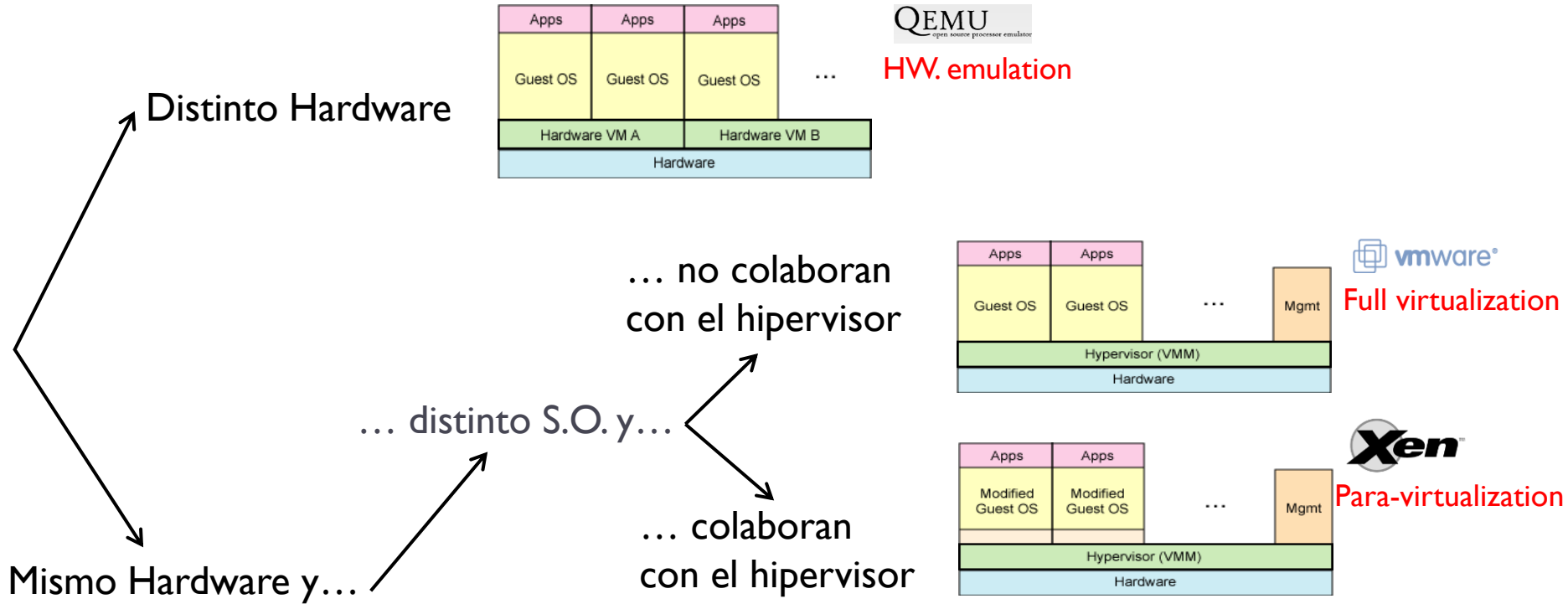
Mismo Hardware y...



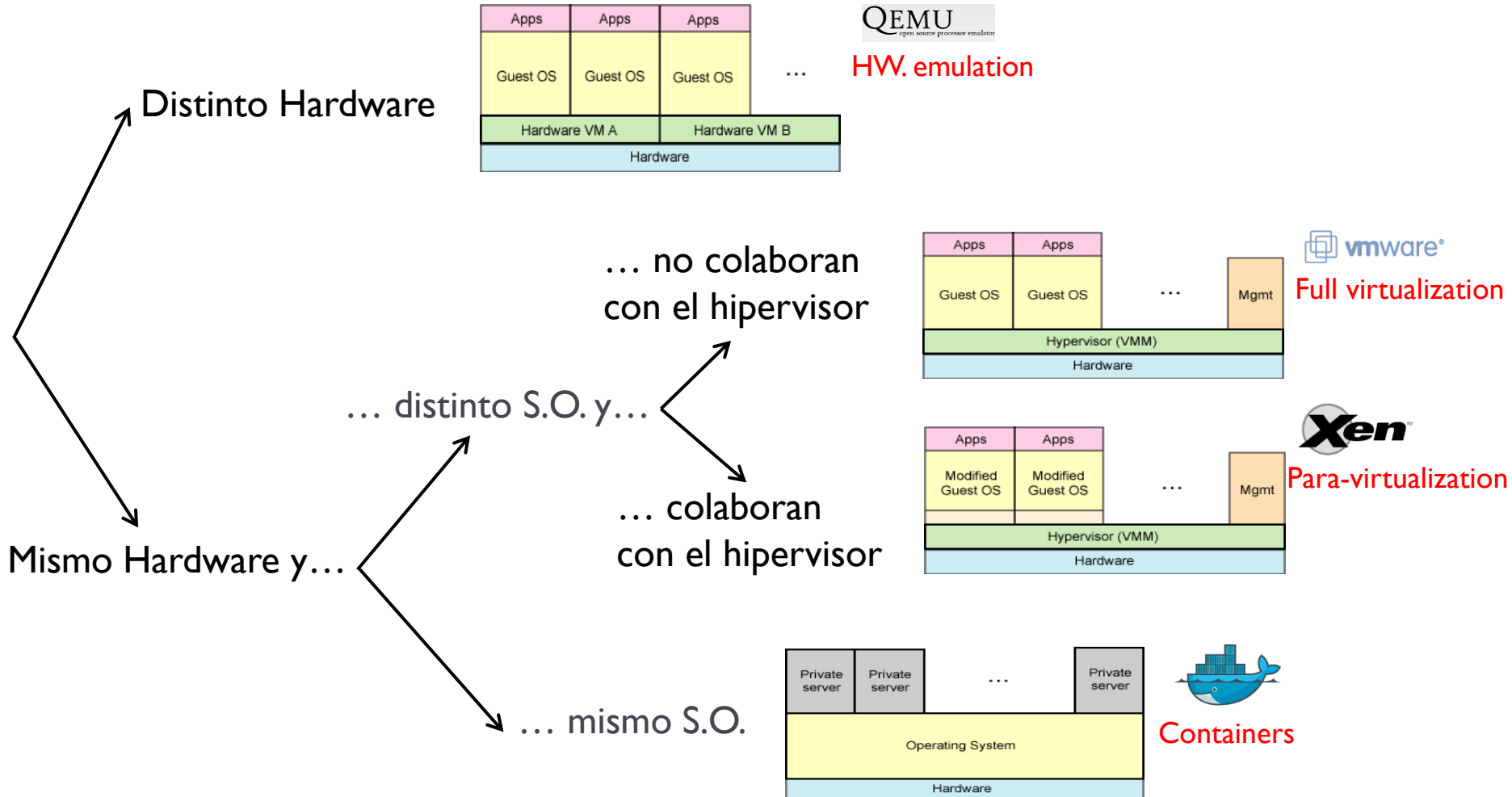
QEMU  
open source processor emulator

HW. emulation

# Máquinas virtuales



# Máquinas virtuales





# Contenidos

---

1. Qué es un sistema operativo.
  1. Definiciones, principales funciones y **características**
2. Cómo es por dentro.
  1. Principales objetivos y estructura

# Principales características

---

- ▶ **Versátil**
  - ▶ Portabilidad
- ▶ **Adaptativo**
- ▶ **Multidisciplinar**
- ▶ **Complejo**
- ▶ **Delicado**

# Versatilidad

---



Mainframe  
OS/360, z/OS, ...



Supercomputador  
Unix, Linux, ...



Miniordenadores y PC  
Unix, MacOS, Windows, ...



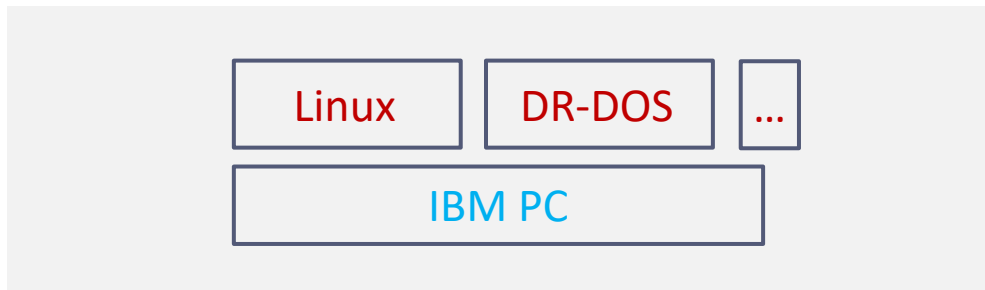
Empotrados  
VxWorks, QNX, LynxOS,  
Android, iOS,  
Windows Embedded, ...

Principales características:

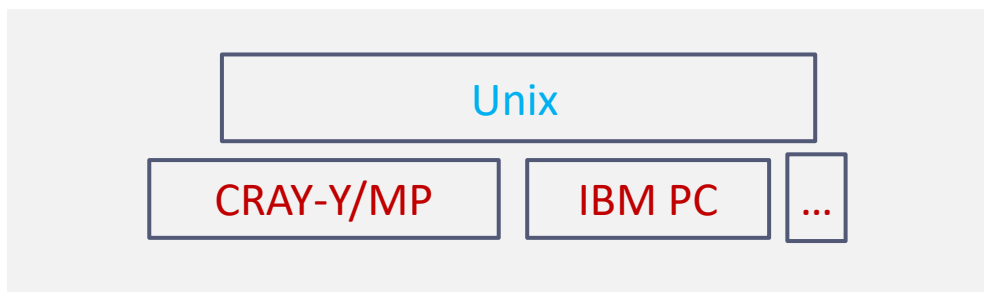
# 1) Versátil

---

- ▶ Mismo equipo, diferentes SSOO: **IBM PC**



- ▶ Mismo SO, diferentes equipos: **Unix**



Portabilidad



Principales características:

## 2) Continuos cambios para adaptarse

---

- ▶ A las nuevas demandas de los usuarios:
  - ▶ Reconocimiento de voz, entrada multitáctil, etc.
- ▶ A la evolución o nuevo tipo de hardware:
  - ▶ Controladores para todo tipo de nuevos dispositivos
  - ▶ Sistemas multicore, virtualización, etc.
- ▶ A integrar soluciones de distintos entornos:
  - ▶ Procesamiento por lotes, multiprogramación, tiempo compartido, etc.
  - ▶ Multiusuario, trabajo colaborativo, etc.
  - ▶ Sistemas distribuidos, servicios en red, etc.

# Resumen de la evolución

---

- ▶ Primera generación (años 50)
  - ▶ Monitor hardware.
- ▶ Segunda generación (años 60)
  - ▶ Procesamiento por lotes.
- ▶ Tercera generación (años 70)
  - ▶ Multiprogramación, tiempo compartido y multiusuario.
- ▶ Cuarta generación (años 80 – actualidad)
  - ▶ Sistemas distribuidos.
  - ▶ Interfaz gráfica, reconocimiento de voz, etc.
  - ▶ Virtualización.

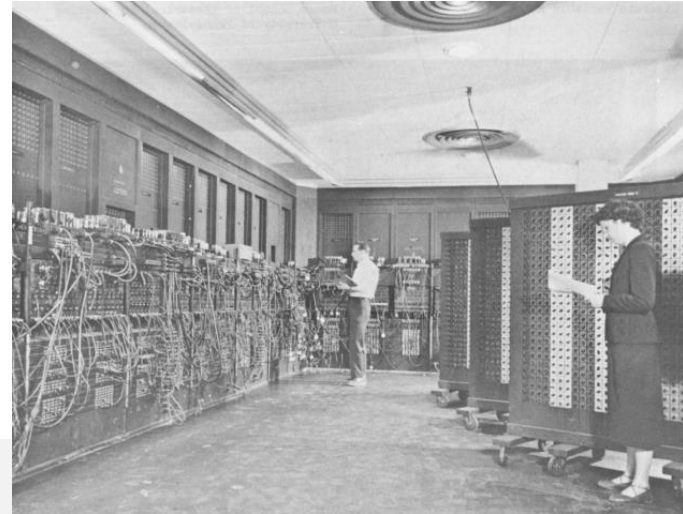
# Primera generación (1945-1955)

## monitor hardware

Hw

S.O.

- ▶ **Tubos de vacío.**
  - ▶ ENIAC, UNIVAC, ...
  - ▶ Grandes, lentos y caros.
  - ▶ Complicación para programar y depurar.
- ▶ **Monitor hardware.**
  - ▶ Asistente para el diagnóstico.



# Segunda generación (1955-1965)

## procesamiento por lotes

Hw

S.O.

### ▶ Transistores.

- ▶ IBM 7090, IBM 1620, ...
- ▶ Más pequeños, rápidos y fiables.
- ▶ Ligeras mejoras en su programación.



### ▶ Procesamiento por lotes.

- ▶ Ejecución de trabajos uno detrás de otro.
- ▶ Enlace de rutinas de librerías a programas.
- ▶ Gestión de archivos, dispositivos de E/S y almacenamiento secundario.
- ▶ Aparición de OS/360



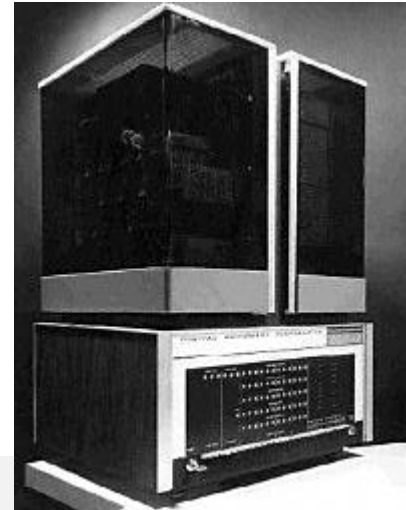
# Tercera generación (1965-1980)

## multiprogramación, multitarea y multiusuario

Hw

S.O.

- ▶ Circuitos integrados.
  - ▶ PDP-8, ...
  - ▶ Más pequeños, rápidos y fiables.
  - ▶ Fortran, C, Pascal, Basic, etc.



- ▶ Multiprogramación.
  - ▶ Gestión de recursos compartidos solapando esperas en CPU.
- ▶ Tiempo compartido y multiusuario.
  - ▶ Compartición de recursos y multiplexación del uso de la CPU.
  - ▶ Aparición de Multics, Unix, CP/M, etc.

# Cuarta generación (1980-) distribuidos, con interfaz gráfica, etc.

Hw

S.O.

- ▶ El microprocesador.
  - ▶ IBM-PC, ...
  - ▶ Ordenadores personales.
  - ▶ C++, etc.



- ▶ **Distribuidos (de tiempo real)**
  - ▶ Conexión a Internet, aplicaciones de trabajo distribuido
- ▶ **Interfaz gráfica.**
  - ▶ System 5, Windows 1.0, etc.



# Cuarta generación (2005-) virtualización, multiprocesamiento, etc.

Hw

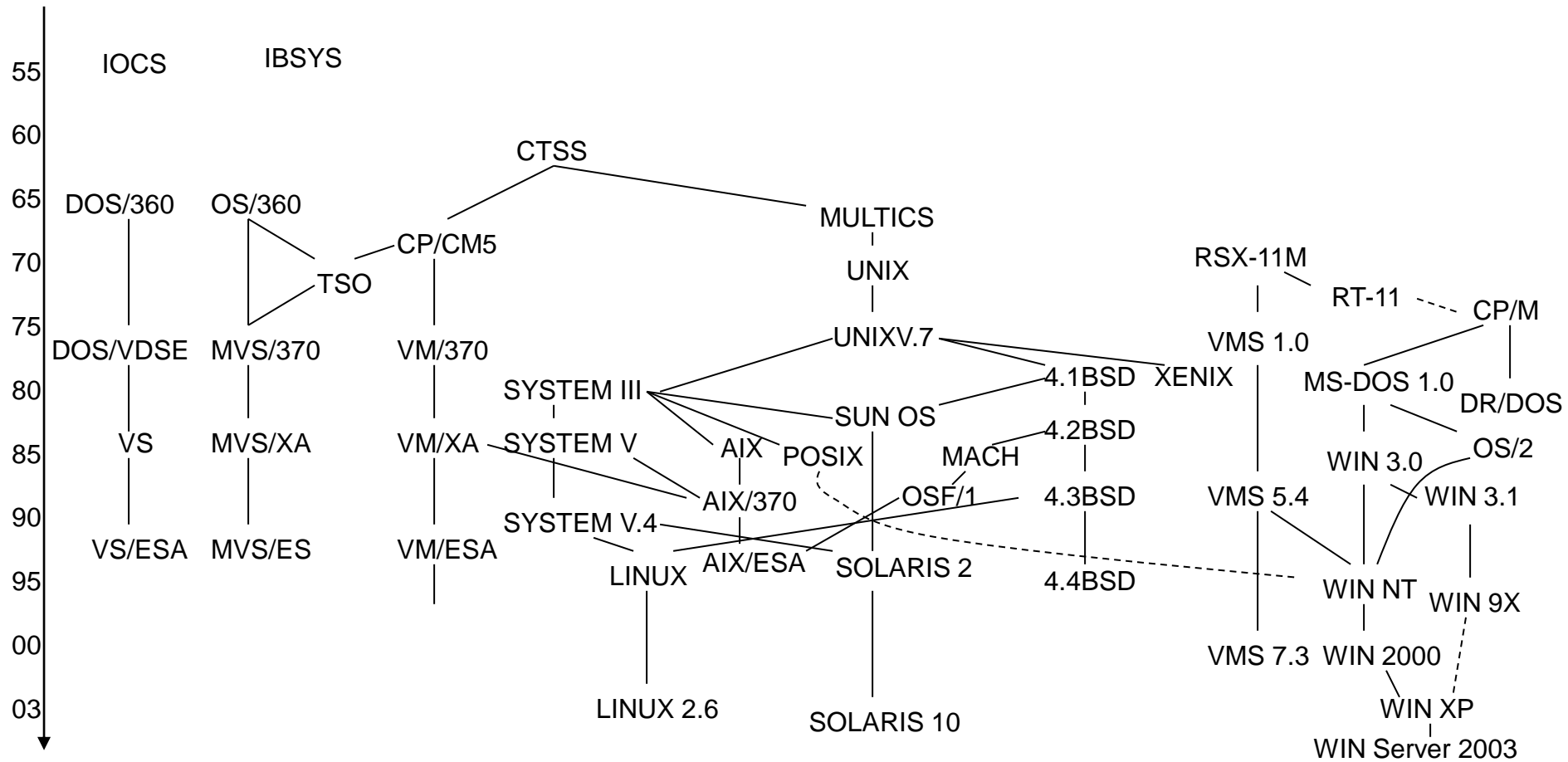
S.O.

- ▶ Los *system-on-chip* y multicores.
  - ▶ Tables, smartphones, ...
  - ▶ Era post-PC.
  - ▶ .net, java, python, php, etc.



- ▶ **Virtualización.**
  - ▶ *Cloud computing, legacy computing, desktop móvil, etc.*
- ▶ **Nuevas formas de interacción.**
  - ▶ Reconocimiento del habla, entrada multitáctil, etc..
  - ▶ Pantallas 3D, holografías, etc.

# Evolución de los sistemas operativos



Principales características:

## 3) Software multidisciplinar

---

- ▶ **Software multidisciplinar:**
  - ▶ Integra trabajos de diferentes áreas:  
Interfaces de usuario, software de sistema, inteligencia artificial, seguridad, Ingeniería Software, etc.



Principales características:

## 4) Software complejo

---

- ▶ **Software complejo:**
  - ▶ Muchas líneas de código.
  - ▶ Muchos equipos de trabajo.

# Principales características:

## 4) Software complejo

- ▶ **Software complejo:**
  - ▶ Muchas líneas de código.
  - ▶ Muchos equipos de trabajo.

### In a Nutshell, Linux Kernel...

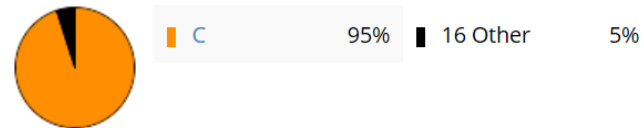
... has had 711,819 commits made by 17,417 contributors representing 16,859,900 lines of code

... is mostly written in C with an average number of source code comments

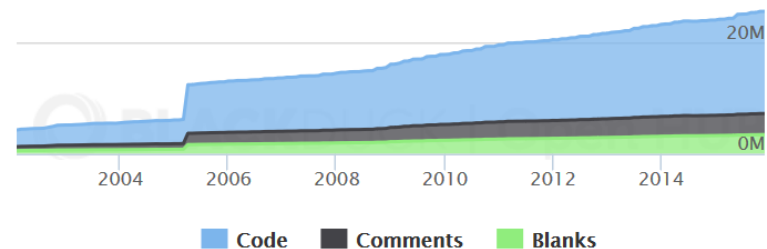
... has a well established, mature codebase maintained by a very large development team with stable Y-O-Y commits

... took an estimated 5,482 years of effort (COCOMO model) starting with its first commit in April, 2005 ending with its most recent commit 2 days ago

### Languages



### Lines of Code

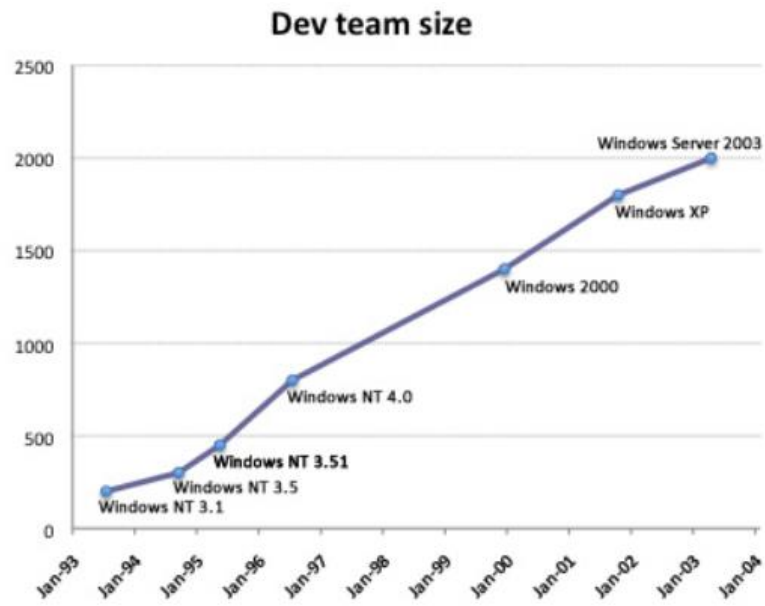


# Principales características:

## 4) Software complejo

---

- ▶ Software complejo:
  - ▶ Muchas líneas de código.
  - ▶ Muchos equipos de trabajo.



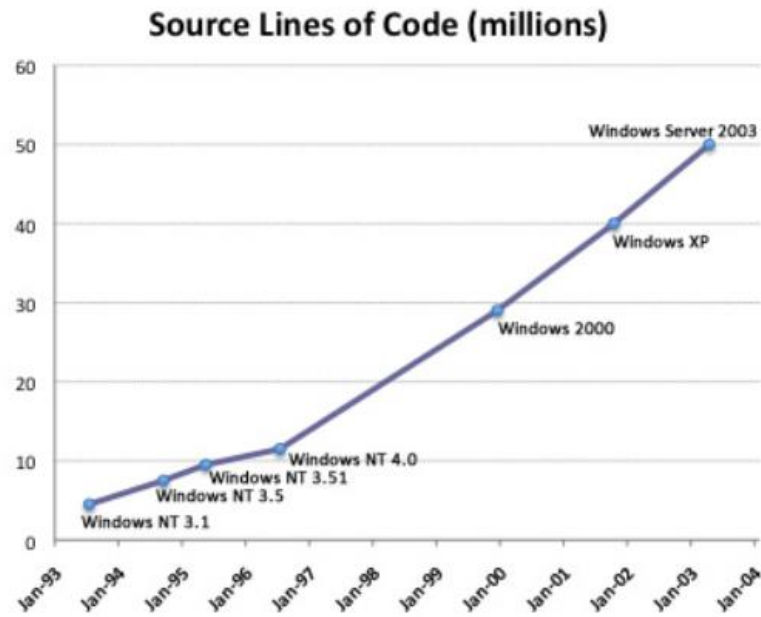


# Principales características:

## 4) Software complejo

---

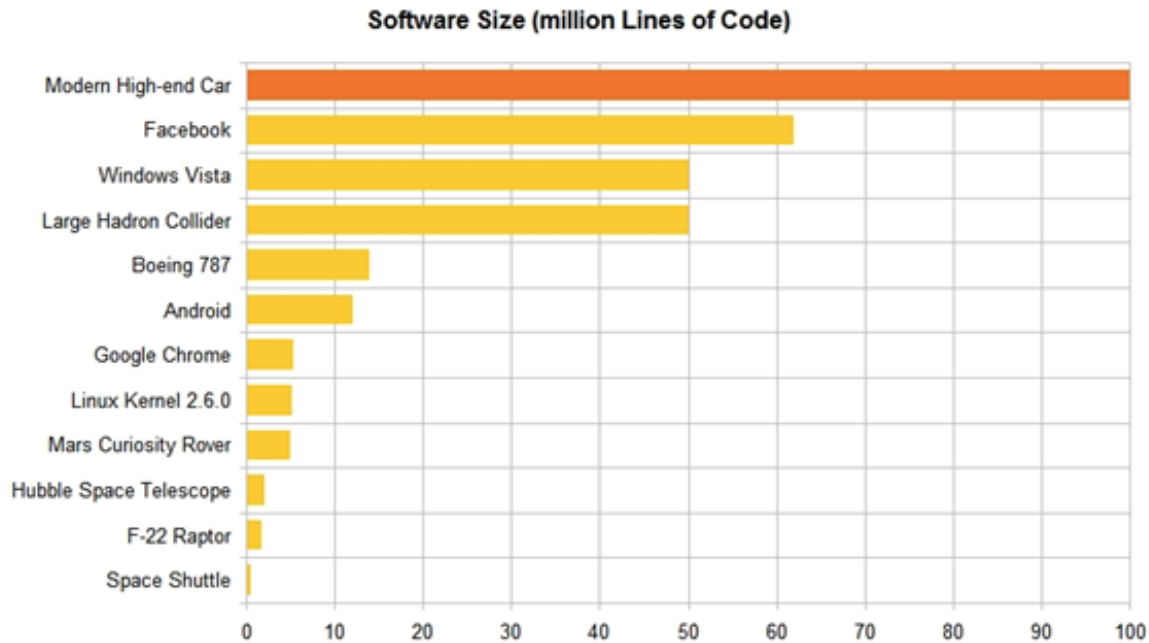
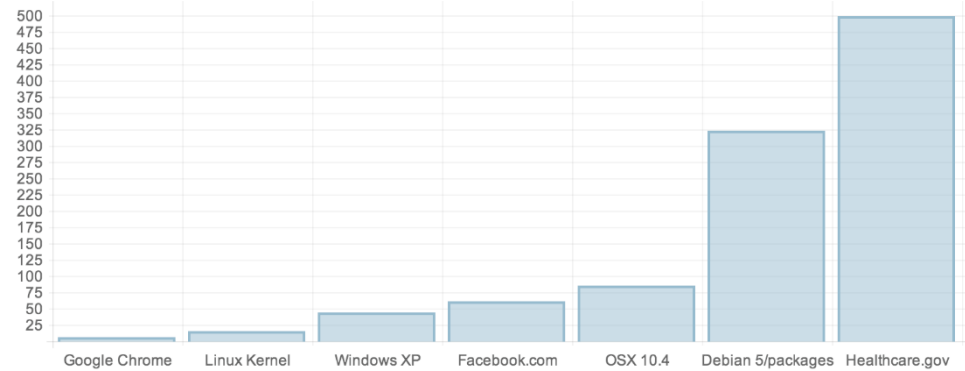
- ▶ Software complejo:
  - ▶ Muchas líneas de código.
  - ▶ Muchos equipos de trabajo.



# Principales características:

## 4) Software complejo

- ▶ Software complejo:
  - ▶ Muchas líneas de código.
  - ▶ Muchos equipos de trabajo.



Principales características:

## 5) Software delicado

---

- ▶ **Software delicado:**
  - ▶ Un fallo en un driver (software controlador de un dispositivo) puede bloquear todo el sistema.
  - ▶ Trata datos de distintas aplicaciones de distintos usuarios que no deben ser perdidos o trasladados a manos incorrectas.

# Principales características:

## 5) Software delicado

---

- ▶ **Software delicado:**
  - ▶ Un fallo en un driver (software controlador de un dispositivo) puede bloquear todo el sistema.
  - ▶ Trata datos de distintas aplicaciones de distintos usuarios que no deben ser perdidos o trasladados a manos incorrectas.

(a) Industry Average: "about 15 - 50 errors per 1000 lines of delivered code." He further says this is usually representative of code that has some level of structured programming behind it, but probably includes a mix of coding techniques.

(b) Microsoft Applications: "about 10 - 20 defects per 1000 lines of code during in-house testing, and 0.5 defect per KLOC (KLOC IS CALLED AS 1000 lines of code) in released product (Moore 1992)." He attributes this to a combination of code-reading techniques and independent testing (discussed further in another chapter of his book).

(c) "Harlan Mills pioneered 'cleanroom development', a technique that has been able to achieve rates as low as 3 defects per 1000 lines of code during in-house testing and 0.1 defect per 1000 lines of code in released product (Cobb and Mills 1990). A few projects - for example, the space-shuttle software - have achieved a level of 0 defects in 500,000 lines of code using a system of format development methods, peer reviews, and statistical testing."

# Principales características:

## 5) Software delicado

---

- ▶ **Software delicado:**
  - ▶ Un fallo en un driver (software controlador de un dispositivo) puede bloquear todo el sistema.
  - ▶ Trata datos de distintas aplicaciones de distintos usuarios que no deben ser perdidos o trasladados a manos incorrectas.



# Principales características

## resumen

---

- ▶ Versátil
  - ▶ Portabilidad
- ▶ Adaptativo
- ▶ Multidisciplinar
- ▶ Complejo
- ▶ Delicado

# Contenidos

---

1. Qué es un sistema operativo.
  1. Definiciones, principales funciones y características
2. Cómo es por dentro.
  1. **Principales objetivos** y estructura

# Objetivos en el diseño de un sistema operativo

---



- ▶ **Rendimiento: eficiencia y velocidad**
  - ▶ Baja sobrecarga, uso adecuado de los recursos
- ▶ **Estabilidad: robustez y resistencia**
  - ▶ Tiempo de funcionamiento, degradación aceptable, fiabilidad e integridad
- ▶ **Capacidad: prestaciones, flexibilidad y compatibilidad**
- ▶ **Seguridad y protección**
  - ▶ Protección entre usuarios
  - ▶ Sistema seguro para 'los malos'
- ▶ **Portabilidad**
- ▶ **Claridad**
- ▶ **Extensibilidad**



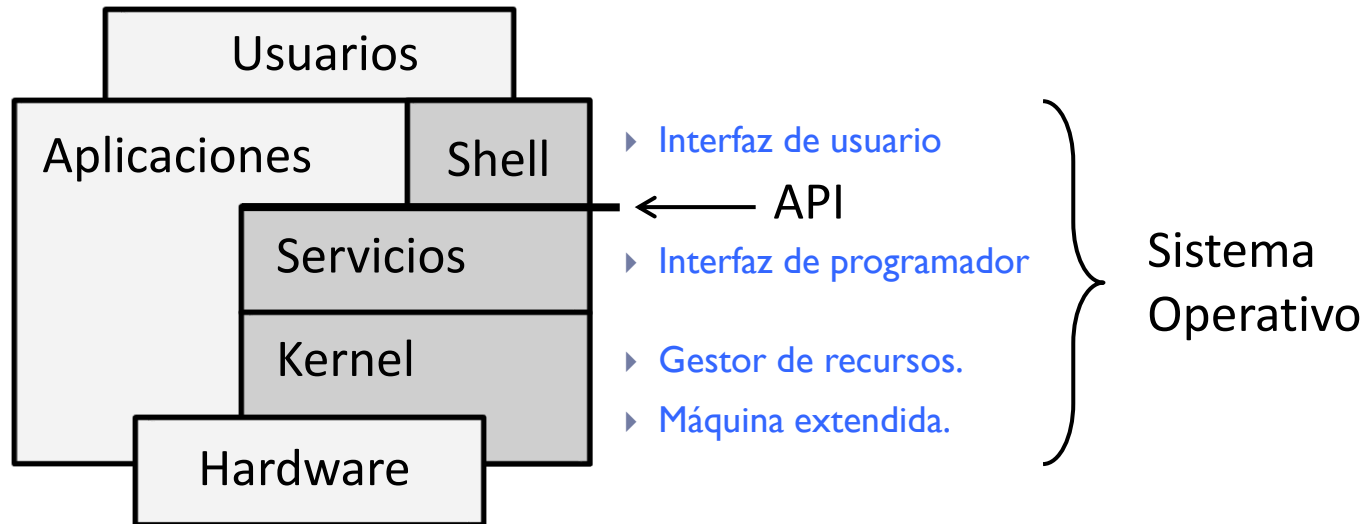
# Contenidos

---

1. Qué es un sistema operativo.
  1. Definiciones, principales funciones y características
2. Cómo es por dentro.
  1. Principales objetivos y **estructura**

# Estructura del sistema operativo

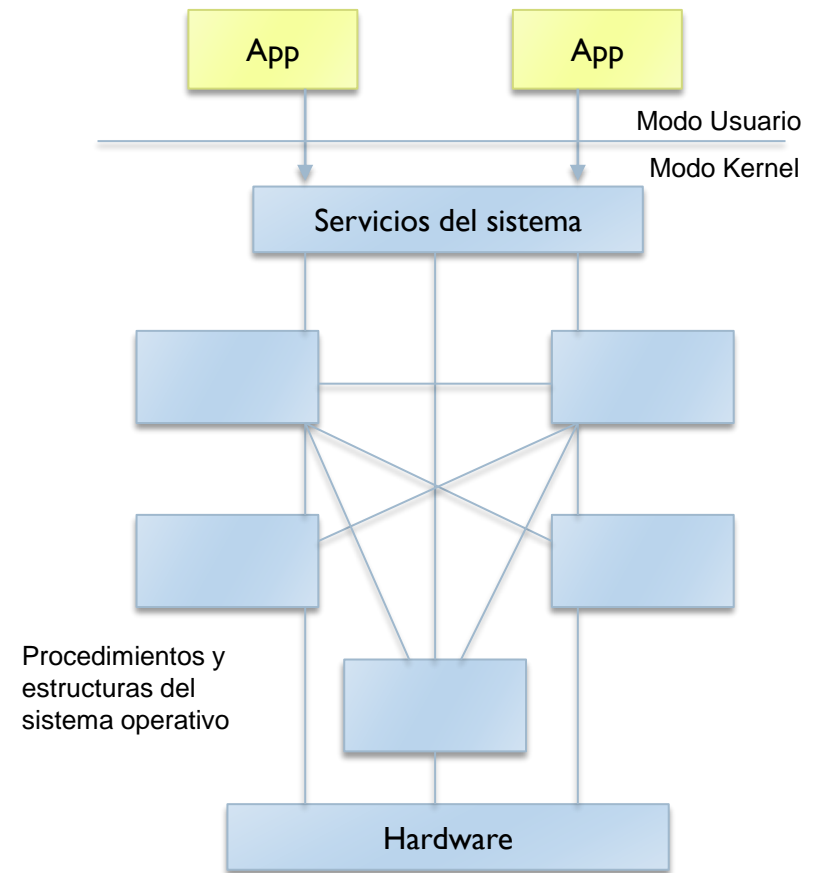
---



# Estructura del Sistema Operativo

## Monolítico (macrokernel)

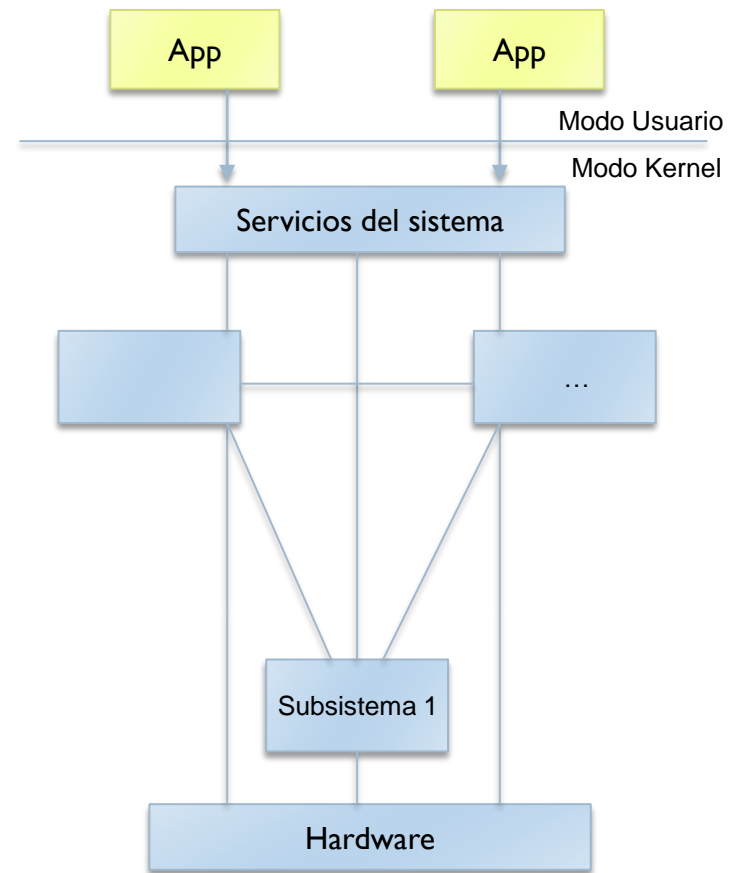
- ▶ Sistema monolítico.
- ▶ No estructurado.
- ▶ Desde cualquier punto del código se puede acceder a cualquier variable o función de otra parte del núcleo (*kernel*)
- ▶ **[!]** muy difícil de mantener, muy sensible a errores



# Estructura del Sistema Operativo

## En subsistemas

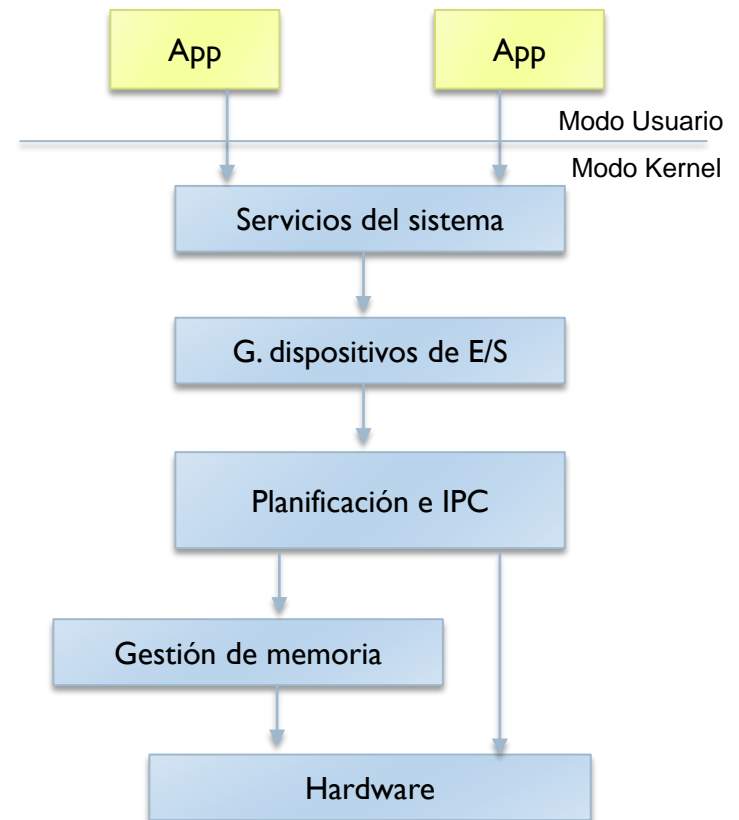
- ▶ Sistema monolítico, compuesto de subsistemas lógicos que ofrecen interfaces bien definidas como puntos de entrada.
- ▶ Se agrupan procedimientos y estructuras de datos relacionadas.
- ▶ Ejemplo:
  - ▶ Linux



# Estructura del Sistema Operativo

## Por capas

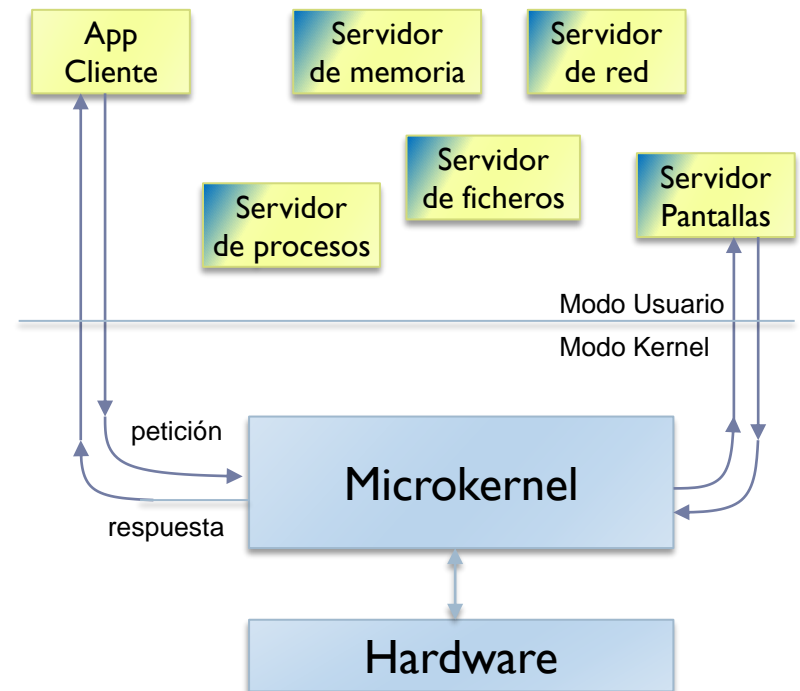
- ▶ Binario monolítico aunque codificado estructurado de forma lógica en capas.
- ▶ Cada capa proporciona acceso únicamente a la interfaz de niveles inferiores.
- ▶ Ejemplo:
  - ▶ **THE** (Dijkstra)
  - ▶ **Multics**, que añadió a la noción de capa la idea de anillos de privilegios



# Estructura del Sistema Operativo

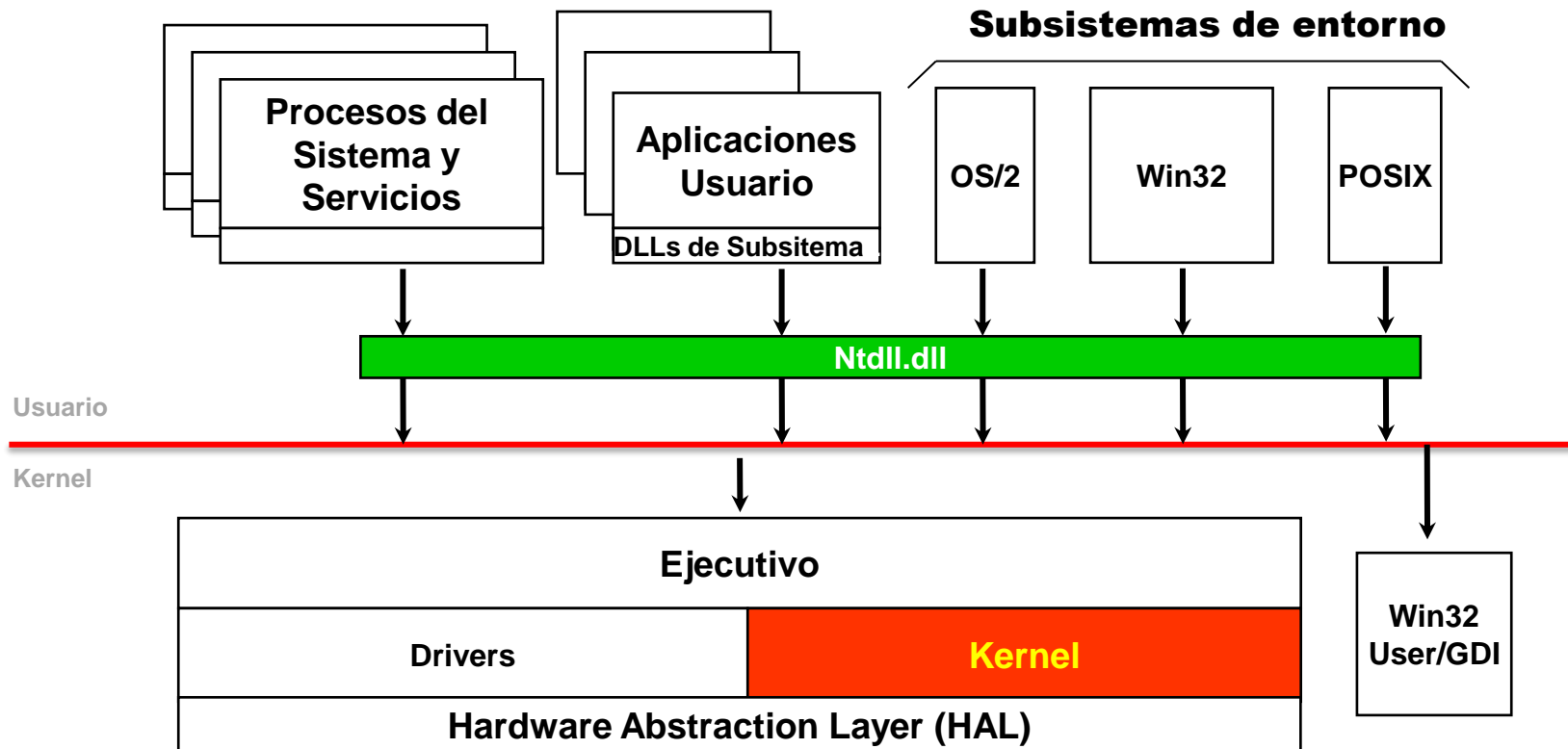
## Microkernel

- ▶ Además de estructurado, los principales componentes se ejecutan como procesos servidores, fuera del kernel.
- ▶ El microkernel tiene:
  - ▶ Planificación y gestión de procesos.
  - ▶ Gestión de memoria virtual básica.
  - ▶ Comunicación entre procesos básica.
- ▶ Ejemplo:
  - ▶ Match, QNX, Minix, L4, etc.



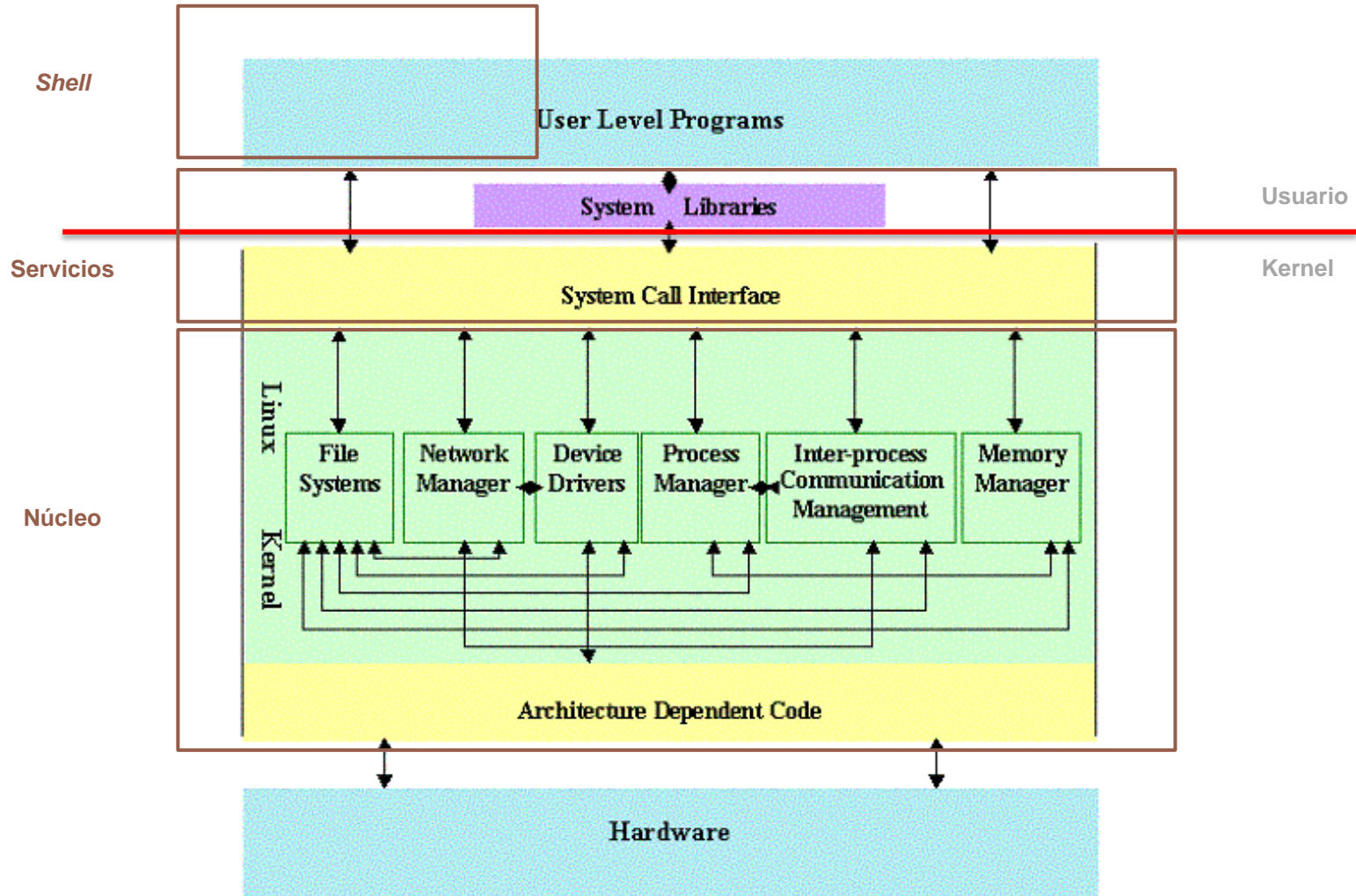
# Estructura del Sistema Operativo

## Windows 2000 (visión simplificada)



# Ejemplo de estructura de subsistemas

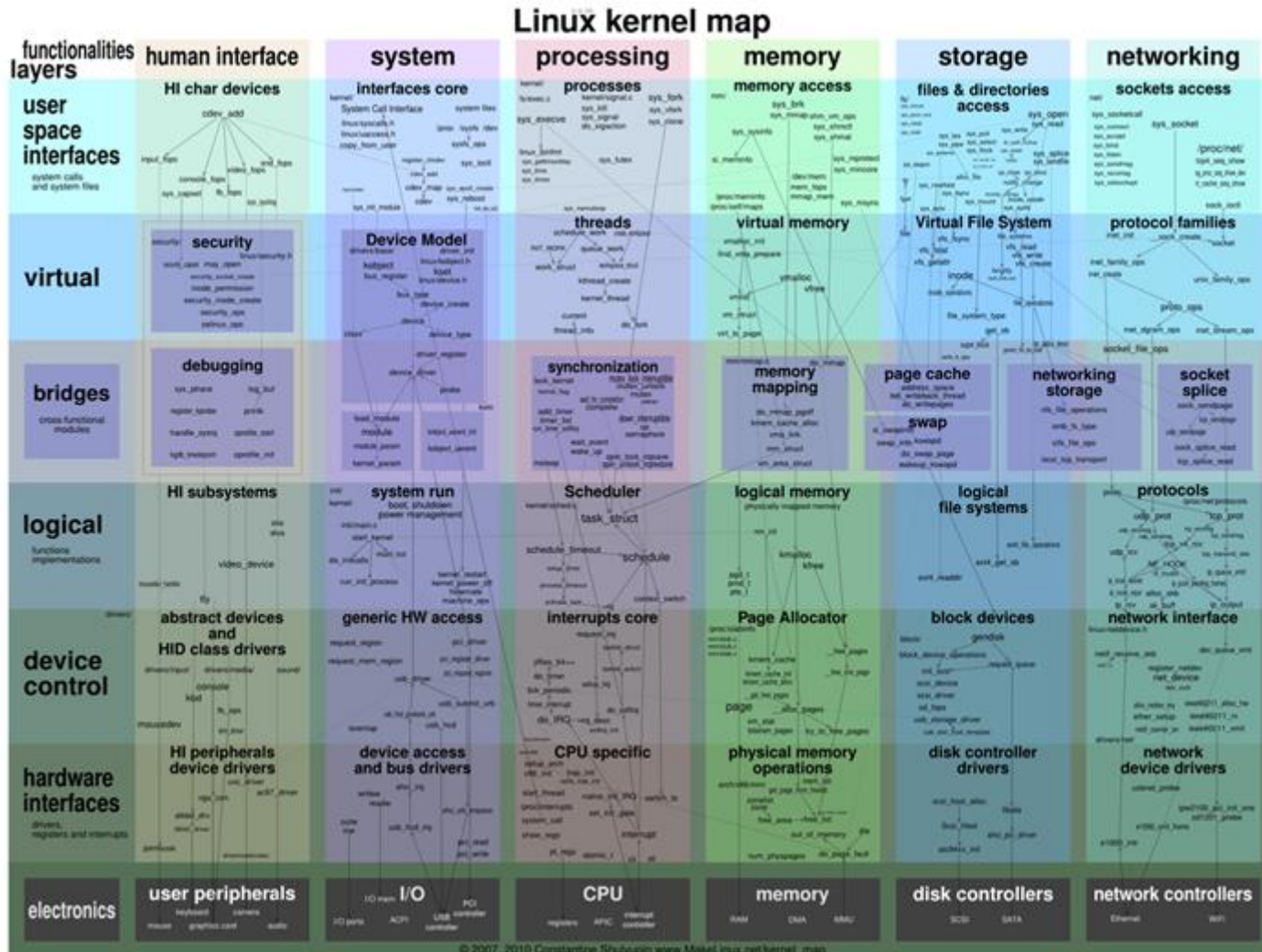
## Linux (versión simplificada)





# SS.OO. Reales

## Linux (versión 'menos' simplificada)



Grupo ARCOS  
Departamento de Informática  
Universidad Carlos III de Madrid

# Lección 1 (a)

## Introducción

Diseño de Sistemas Operativos  
Grado en Ingeniería Informática y Doble Grado I.I. y A.D.E.

