

# INTRODUCCIÓN A LOS SISTEMAS OPERATIVOS

Rodrigo García Carmona  
Universidad San Pablo-CEU  
Escuela Politécnica Superior



# OBJETIVOS

- Presentar el **concepto** de sistema operativo.
- Describir la **funcionalidad** implementada en un sistema operativo.
- Dar una visión general de la **evolución** de los sistemas operativos, justificando cuándo y por qué se introdujeron ciertas funciones.
- Entender los conceptos de **llamada a sistema, servicio y programa**.
- Presentar la **shell** o **intérprete de órdenes** como el ejemplo más importante de programa de sistema.

# CONTENIDO

- Concepto de sistema operativo
- Funciones del sistema operativo
- Evolución de los sistemas operativos
- Llamadas a sistema
- Programas de sistema

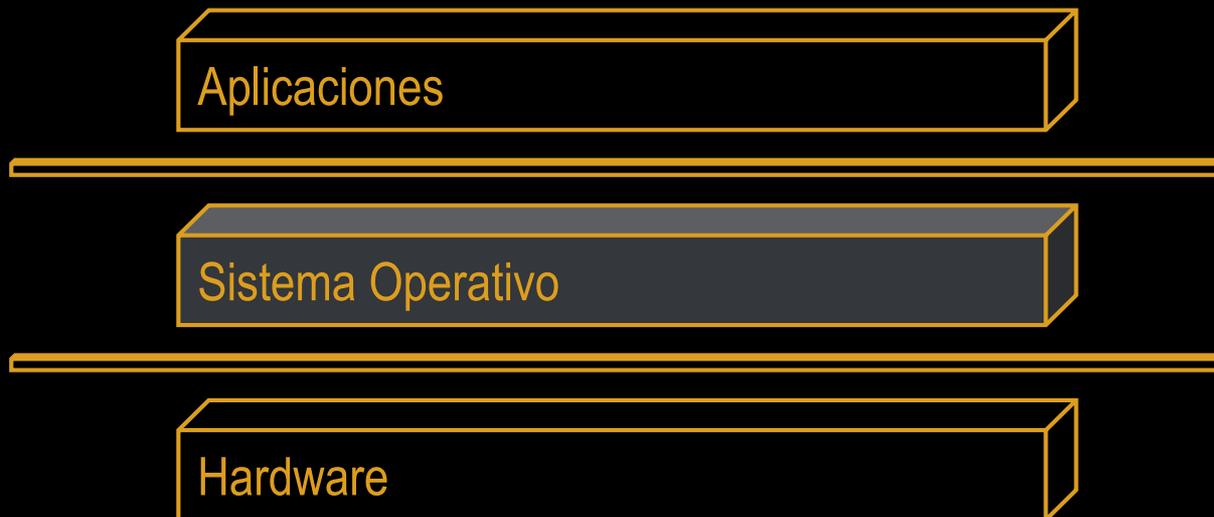
## Bibliografía

- W. Stallings:  
**Sistemas Operativos.**
  - Capítulo 2.
- A.S. Tanenbaum:  
**Modern Operating Systems.**
  - Capítulos 1 y 3.

# CONCEPTO DE SISTEMA OPERATIVO

# CONCEPTO DE SISTEMA OPERATIVO

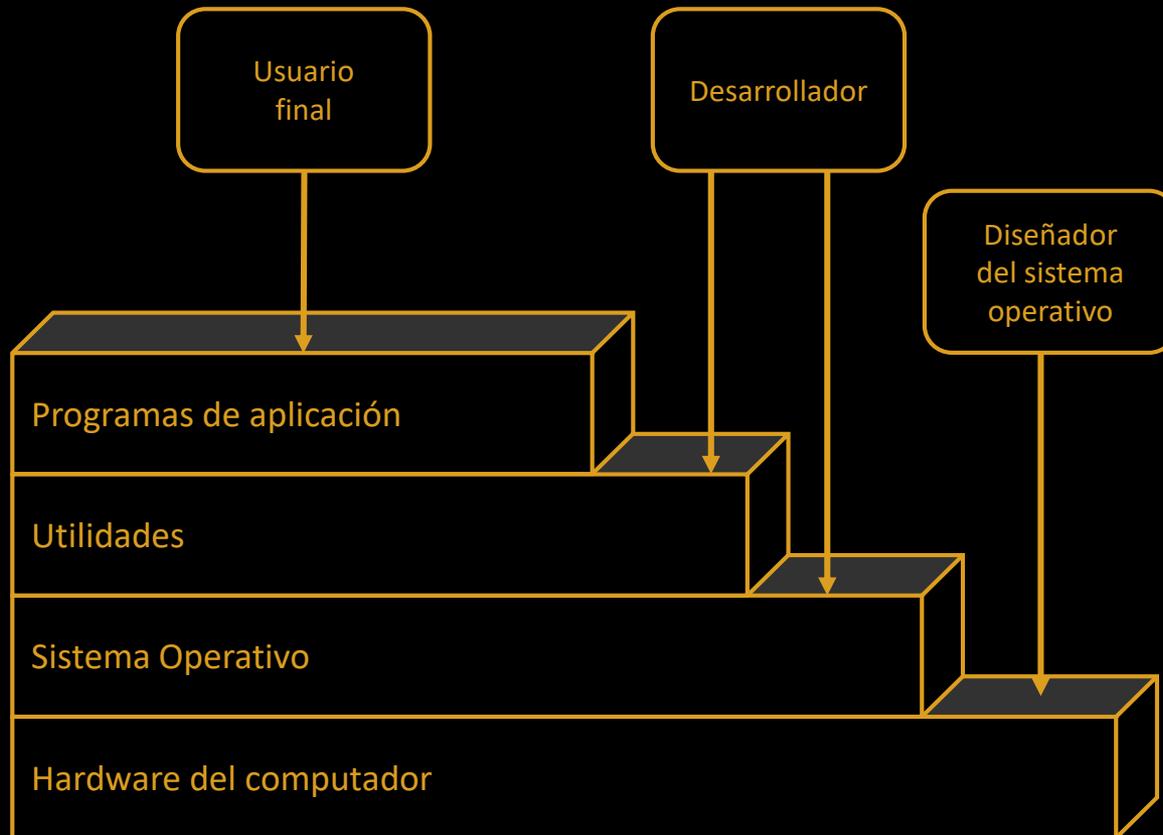
- Capa de software que actúa entre las aplicaciones de usuario y el hardware del computador.
- Abreviada SO (*OS, Operating System* en inglés).



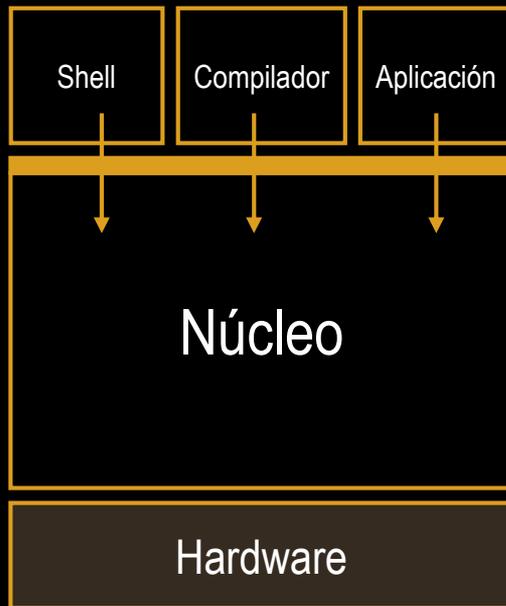
# OBJETIVOS DE UN SISTEMA OPERATIVO

- **Comodidad:** Facilitar a los usuarios el uso de la máquina, proporcionando un entorno cómodo para el desarrollo de programas y el acceso a sus recursos.
  - Hace que un ordenador sea más sencillo de utilizar.
- **Eficiencia:** Fomentar que los recursos de un sistema se aprovechen de una manera más eficiente.
  - Es difícil conseguir que se cumplan comodidad y eficiencia a la vez.
  - La arquitectura de los ordenadores y el SO se influyen mutuamente.
- **Capacidad de evolución:** Permitir el desarrollo de nuevas funciones sin interferir en los servicios actuales.

# NIVELES EN UN ORDENADOR



# PARTES DE UN SISTEMA OPERATIVO



- **Núcleo del SO:** se carga al arrancar la máquina y se mantiene siempre en memoria.
- **Procesos de usuario:** aplicaciones hechas por desarrolladores que se ejecutan sobre el SO. Solicitan los servicios del SO invocando **llamadas a sistema**.
- **Llamadas al sistema:** conjunto de servicios que el SO ofrece a los procesos de usuario.
- **Aplicaciones de sistema:** utilidades del SO que se ejecutan como procesos de usuario:
  - Editores, compiladores y montadores.
  - Sistema de ventanas, aplicaciones para manipulación de ficheros.
  - Aplicaciones de acceso a la red.

# MODOS DE EJECUCIÓN

- Los procesadores actuales tienen como mínimo dos modos de funcionamiento o modos de ejecución:
  - **Supervisor o privilegiado:** En este modo es posible acceder a todo el juego de instrucciones del procesador. Con ello se pueden utilizar todos los recursos, y realizar ciertas acciones tales como programar los controladores de dispositivo, atender interrupciones, programar la unidad de gestión de memoria, etc.
  - **Usuario:** En este modo no se pueden utilizar las instrucciones del procesador relacionadas con la entrada/salida, la gestión del modo de ejecución, la gestión de memoria o cualquier instrucción privilegiada.
- En el modo supervisor únicamente se ejecuta el sistema operativo. Por tanto es el único que tiene control y acceso a todos los recursos de un equipo.
- En el modo usuario se ejecutarán todos los demás programas. Para utilizar los recursos no accesibles en modo usuario (aquellos que requieran instrucciones privilegiadas), se deberán realizar **llamadas al sistema**.

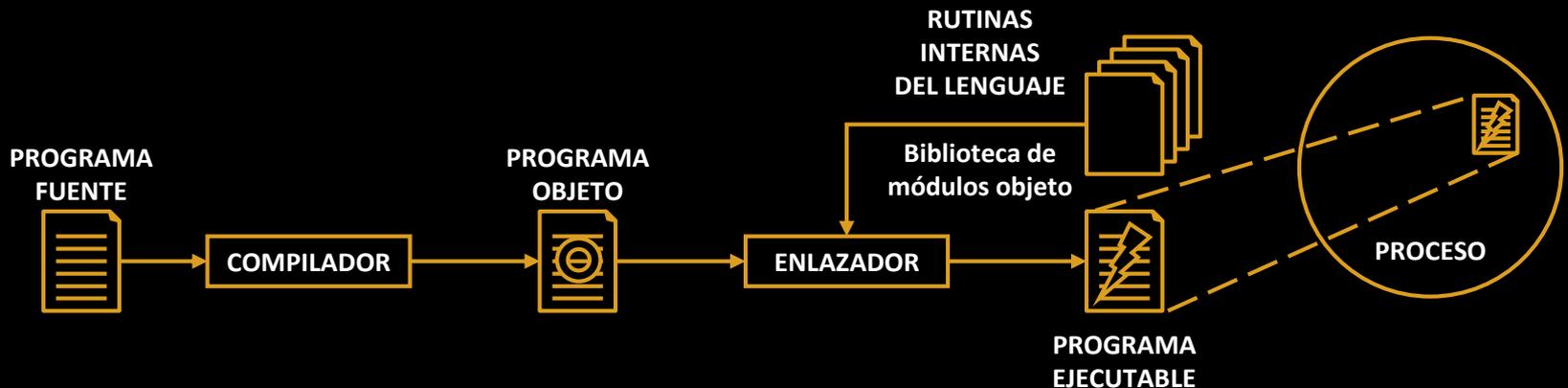
# FUNCIONES DEL SISTEMA OPERATIVO

# FUNCIONES DEL SISTEMA OPERATIVO

- Funciones que lleva a cabo un sistema operativo:
  - Gestión de procesos.
  - Gestión de memoria.
  - Gestión del sistema de ficheros.
  - Gestión de dispositivos de entrada/salida.
  - Gestión de la red.
  - Protección.
- Cada una de ellas viene caracterizada por una serie de abstracciones que se han desarrollado para resolver los obstáculos de los problemas prácticos.

# PROGRAMAS Y PROCESOS

- Es importante diferenciar entre:
  - **Programa:** secuencia de instrucciones. Ente pasivo. Se almacena en ficheros. “Lo que se escribe”.
  - **Proceso:** ejecución de un programa. Ente activo. Se encuentra en memoria. “Lo que se ejecuta”.



# GESTIÓN DE PROCESOS

- Problemática asociada a los procesos:
  - **Ejecución concurrente:**
    - Simular la ejecución en paralelo de programas.
    - Sólo se dispone de un único procesador.
    - Es necesario repartir el tiempo de procesador entre los procesos.
  - **Planificación de procesos:**
    - Determinar cómo se reparte el tiempo de procesador.
    - Dar prioridad a quién la merece.
- Funciones para la gestión de procesos:
  - **Creación y eliminación de procesos.**
  - **Comunicación y sincronización de procesos.**



# GESTIÓN DE MEMORIA

- Problemática asociada a la memoria:
  - **Aislamiento del proceso:** cada proceso debe tener una zona de memoria independiente que no puede ser interferida por otros procesos.
  - **Asignación de la memoria:** controlar las zonas de memoria libres y ocupadas y la zona de memoria asignada a cada proceso.
  - **Memoria virtual:** permite a los procesos ver y ocupar una memoria que “no existe”. Crea la ilusión de una memoria más grande.
- Funciones para la gestión de la memoria:
  - **Transparencia frente al programador:** la asignación y la liberación se llevan a cabo automáticamente con la creación y eliminación de procesos.
  - **Particionamiento de la memoria.**
  - **Compartición entre procesos.**

# GESTIÓN DEL SISTEMA DE FICHEROS

- **Fichero:** Abstracción para almacenar y organizar información **persistente**.
  - Organización en forma lógica.
  - Con vida más allá de un proceso.
- Problemática asociada a los ficheros:
  - **Organización del sistema de ficheros:** directorios.
  - **Asignación de espacio:** asignar espacio en disco a la información de manera no necesariamente contigua.
  - **Gestión de espacio:** del espacio libre y ocupado en disco.
  - **Gestión del acceso compartido.**
- Funciones para la gestión de ficheros:
  - **Operaciones sobre ficheros y directorios:** creación, eliminación, lectura, escritura. copia, ...
  - **Manejo de hardware de almacenamiento.**

# GESTIÓN DE DISPOSITIVOS DE ENTRADA/SALIDA

- **Dispositivos de entrada/salida (E/S) o *input/output* (I/O):**
  - Introducen/reciben información de un ordenador.
  - Enorme diversidad.
- Es necesario proporcionar una interfaz de “alto nivel”:
  - Sencilla de utilizar.
  - Uniforme.
  - Abstraída de las peculiaridades del dispositivo
- Problemática asociada a la gestión de dispositivos de E/S:
  - **Desarrollo de controladores (*drivers*):**
    - Ocultan la complejidad del hardware.
    - Interfaz uniforme.
- Funciones para la gestión de dispositivos de E/S:
  - **Uso de los dispositivos:** proporcionar acceso exclusivo, spooling, buffering....

# GESTIÓN DE LA RED

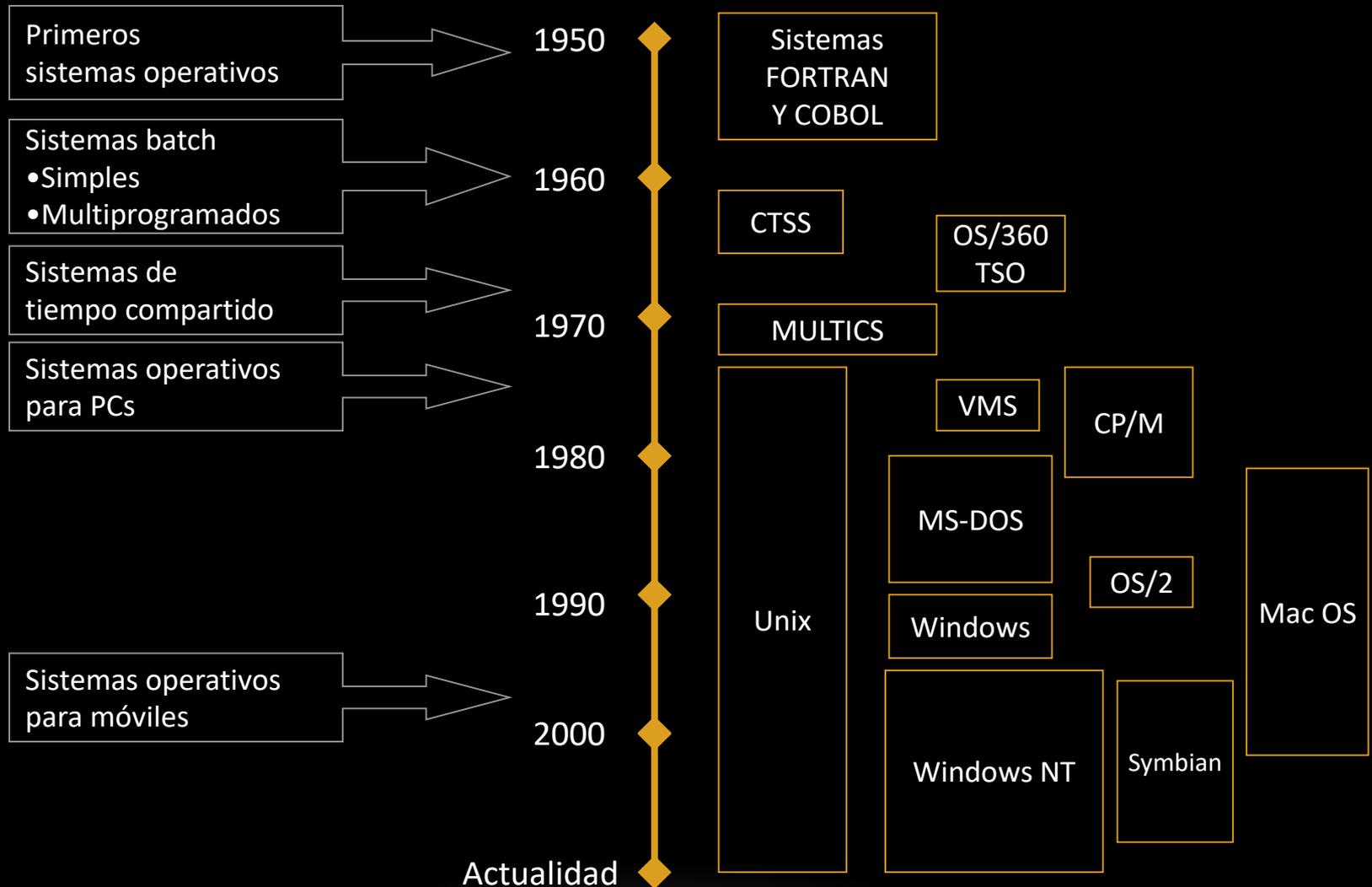
- Es raro el ordenador no enlazado a ninguna red.
- Comprende varios niveles:
  - **Drivers de las tarjetas de red:**
    - Igual que en la gestión de dispositivos de E/S.
    - Ethernet, WiFi, Modem...
  - **Protocolos de comunicación:**
    - Controlan el acceso a la red.
    - Proporcionan una API para la comunicación entre procesos remotos.
    - TCP/IP, NetBios, PPP...
  - **Aplicaciones para el uso de la red:**
    - Programas de aplicación.
    - Construidos sobre la API de comunicación.
    - Acceso a recursos remotos.
    - WWW, BitTorrent, Streaming de video...

# PROTECCIÓN

- Se superpone a todas las funciones anteriores.
- Mecanismo para permitir/denegar/controlar el uso de un recurso.
- Problemática asociada a la protección:
  - **Diseño de un modelo y una política** de protección para definir qué accesos son legales y cuáles son ilegales.
  - **Implementación de un mecanismo ejecutivo** que vigile el cumplimiento de las reglas de protección definidas.
  - **Seguridad:** mecanismo para garantizar la privacidad y estado de la información frente a ataques intencionados.
- Funciones de protección:
  - Funciones para definir **propietarios** de recursos o niveles de protección.
  - Capacidad para **hacer cumplir (enforce)** la protección establecida.

# EVOLUCIÓN DE LOS SISTEMAS OPERATIVOS

# HISTORIA DE LOS SISTEMAS OPERATIVOS



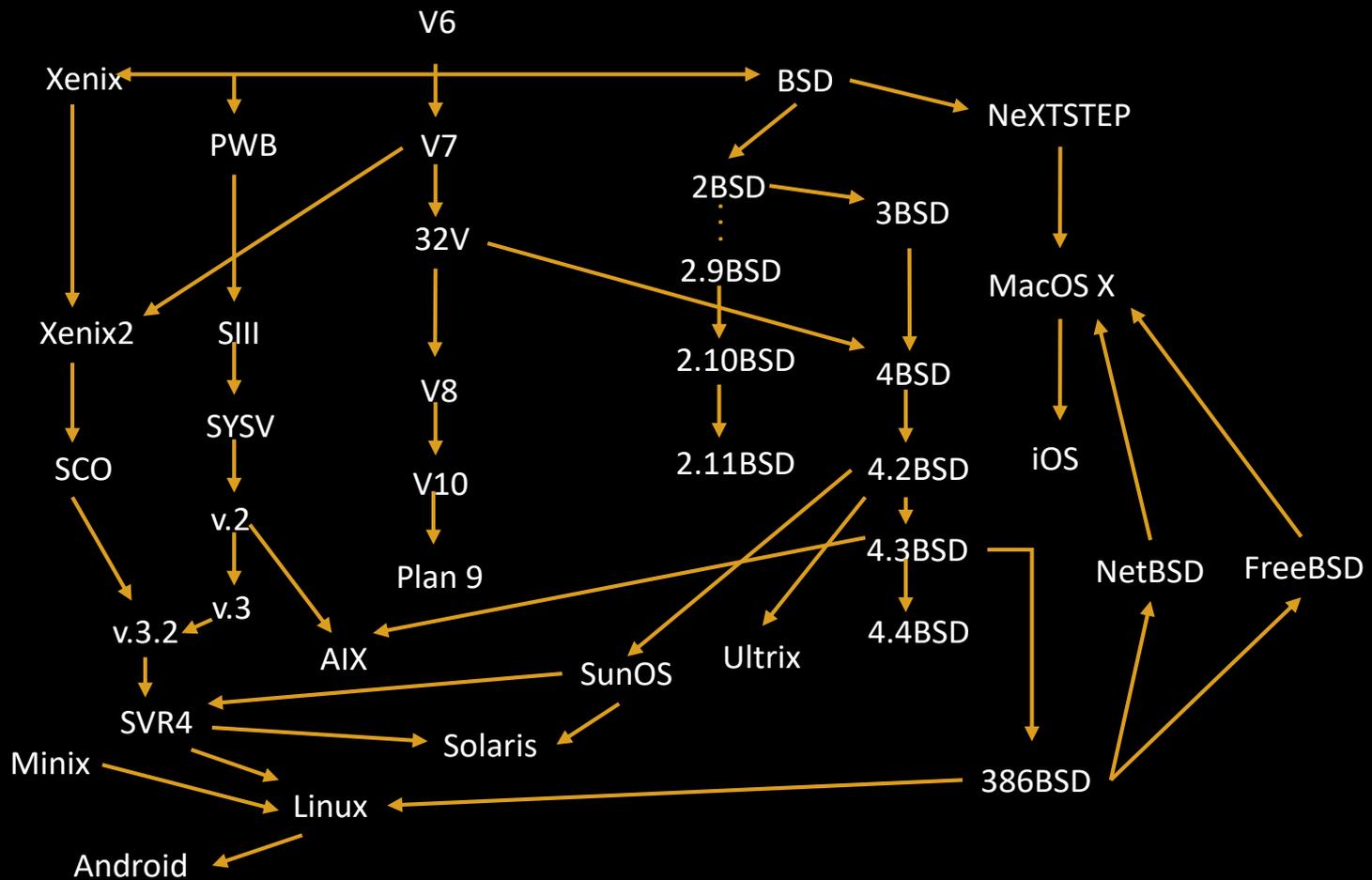
# SISTEMAS UNIX

- **Miniordenadores:** máquinas de arquitectura más sencilla que los mainframes pero con mayor potencia de cálculo y menos usuarios.

PDP-1, ..., PDP-7, ..., PDP-11

- **Objetivos:**
  - Adaptar los sistemas de tiempo compartido MULTICS a miniordenadores.
  - Utilización de lenguajes de alto nivel para el diseño y desarrollo de SO.
- **Aportaciones de Unix:**
  - Diseño sencillo y potente con dos únicas abstracciones: proceso y fichero.
  - Introduce el modelo de sistema de ficheros actual.
  - Introduce un potente intérprete de órdenes.
  - Entorno de programación muy completo.
  - Base para el desarrollo de la primera red de computadores y TCP/IP.
  - Introduce el concepto de sistema abierto.

# EVOLUCIÓN DE LOS SISTEMAS UNIX



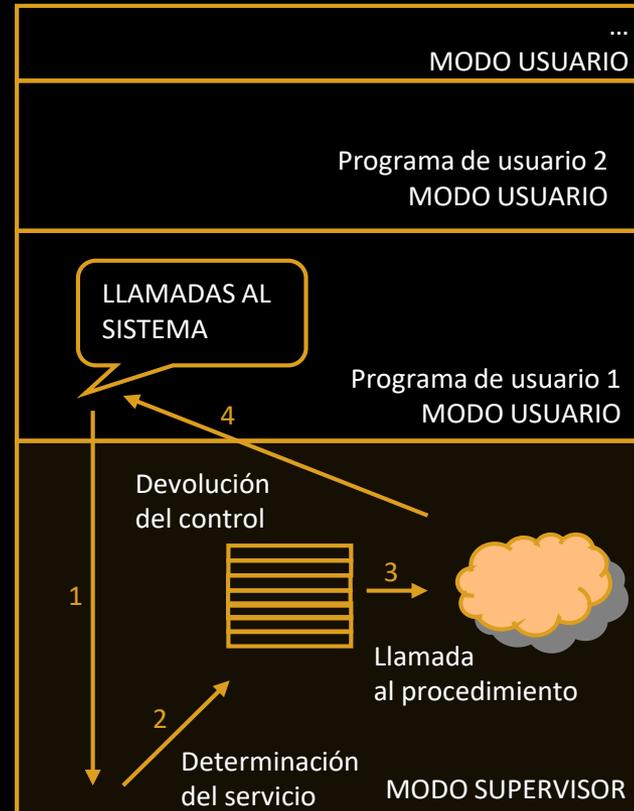
# TENDENCIAS ACTUALES

- **Sistemas operativos en red:**
  - Permiten el acceso a recursos remotos.
  - Proporcionan protocolos de comunicación y gran variedad de aplicaciones para acceso a la red.
- **Sistemas distribuidos:**
  - Permiten el acceso a recursos remotos de manera transparente (obviando el hecho de la distribución).
  - Proporcionan entornos de programación distribuida y acceso transparente a bases de datos remotas.
- **Sistemas multimedia:**
  - Sistemas que incorporan interfaces audiovisuales.
  - Desarrollo de nuevas interfaces usuario - máquina.
  - Desarrollo de protocolos de transmisión en tiempo real.
- **Sistemas empotrados:**
  - En dispositivos que no tienen interacción con el usuario.
  - En algunos casos son de tiempo real.
  - De propósito específico.

# LLAMADAS A SISTEMA

# LLAMADAS A SISTEMA

- Conjunto de servicios básicos que proporciona el sistema operativo.
- Definen la interfaz entre un proceso y el sistema operativo.
- Permiten al SO supervisar las operaciones “peligrosas”.
- El SO es el único que puede funcionar en modo supervisor.
- Las llamadas a sistema se proporcionan como funciones de biblioteca de lenguajes de programación.



# PROGRAMAS DE SISTEMA

# PROGRAMAS DE SISTEMA

- **No son, estrictamente hablando, parte del sistema operativo.**
- Utilidades del sistema operativo que se ejecutan como **procesos de usuario** y proporcionan un entorno más cómodo.
- Como cualquier otro programa, realizan llamadas al sistema. Para acceder a las funciones del sistema operativo.
- **Clasificación** (ejemplos de UNIX):
  - **Intérpretes de órdenes:** sh, ksh, bash,...
  - **Tratamiento de ficheros y directorios:** mkdir, cp, mv, ls, ...
  - **Filtros:** sort, grep, head, tail, ...
  - **Comunicaciones:** mail, ftp, rlogin, ...
  - **Sistema de ventanas:** X11

# INTÉRPRETE DE ÓRDENES

- En inglés *shell*.
- Interfaz primaria entre el usuario y el sistema operativo.
- Lee de su entrada estándar una orden introducida por un usuario, la analiza y la ejecuta.
- Suele ser un programa que se ejecuta como un proceso de usuario.
- Ejemplos: bash o tsch de Unix, command.com de MS-DOS
- **Dos tipos de órdenes:**
  - **Órdenes externas:**
    - El intérprete de órdenes crea un proceso para ejecutar la orden.
    - La orden es cualquier fichero ejecutable.
    - *Ejemplo:* cp origen destino
  - **Órdenes internas:**
    - El intérprete de órdenes ejecuta él mismo la orden.
    - Su código contempla las acciones necesarias para ejecutarla.
    - *Ejemplo:* cd.