

Sistema Operativo



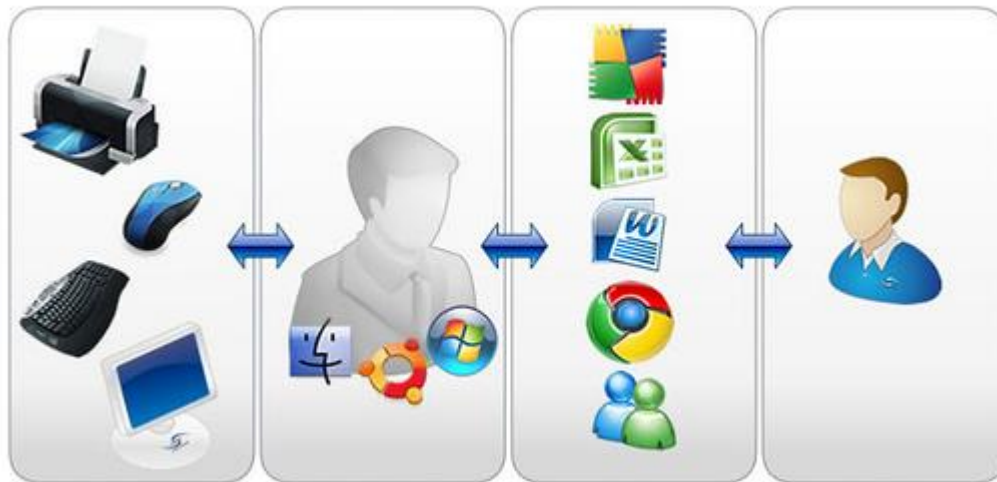
Ing. Eddie Angel Sobrado Malpartida

Definición

- Es un conjunto de **programa**, **rutinas**, **funciones software**, etc que hace de interfaz entre el **usuario** y el **hardware** de un sistema informático.

Objetivos principales

- ✓ Facilitar el uso del sistema informático
- ✓ Ofrecer un entorno adecuado para la ejecución de programas en un sistema informático



Definición

- El SO. **gestiona los recursos** de hardware disponibles, entre ellos el tiempo del procesador y la memoria.
- La gestión del tiempo del procesador permite al programador de aplicaciones escribir múltiples subprogramas como si cada uno fuera el único que utiliza la CPU.
 - ✓ Una parte del OS se encarga de asignar tiempo de ejecución a todos los programas que tiene cargados en base a un juego de reglas conocido de antemano. A estos subprogramas se los llama tareas.
- Con esto se logra la ilusión de que múltiples programas se ejecutan simultáneamente, aunque en realidad sólo pueden hacerlo de a uno a la vez (en sistemas con un sólo núcleo, como es el caso general de los sistemas embebidos).

Cómo se administra el tiempo del CPU?

- El encargado de esta gestión es un componente del SO llamado *scheduler* o programador de tareas. Su función es determinar **qué tarea debe estar en ejecución** a cada momento.
- Ante la ocurrencia de ciertos eventos revisa si la tarea en ejecución debe reemplazarse por alguna otra tarea. En este reemplazo se produce un **cambio de contexto de ejecución**.



Contexto

- Se llama **contexto de ejecución** al conjunto de recursos que identifican el estado de ejecución de una tarea:
 - ✓ IP (instruction pointer)
 - ✓ SP (stack pointer)
 - ✓ Registros del CPU
 - ✓ Contenido de la pila en uso

Cambios de contexto

- Cuando el **scheduler** determina que debe cambiarse el contexto de ejecución, invoca a otro componente del SO llamado **dispatcher** para que **guarde el contexto** completo de la tarea actual y lo **reemplace por el de la tarea entrante**.
- Por esta razón debe reservarse un bloque de memoria de datos para cada tarea. Esto limita la cantidad de tareas simultáneas del sistema (pueden sin embargo eliminarse y crearse nuevas tareas en tiempo de ejecución).

Cambios de contexto

- Estos **cambios de contexto** se realizan de forma transparente para la tarea, no agregan trabajo al programador. Los SO trabajan en dos modos:
 - ✓ En modo **cooperativo**, estos cambios solo ocurren cuando la tarea en ejecución relega voluntariamente el uso del CPU.
 - ✓ En cambio en modo **preemptive** el scheduler tiene la facultad de remover una tarea sin el consentimiento de la misma.

En este caso debe preverse que algunas operaciones no deben ser interrumpidas, a estos pasajes del programa se los llama secciones críticas. Los SO permiten inhibir de este modo los cambios de contexto cuando es necesario.

Servicios del sistema operativo

- El sistema brindará un entorno de ejecución de programas dónde se dispondrá de un conjunto de servicios que serán accesible mediante una interfase bien definida.

Servicios del sistema operativo

- **Ejecución de programas:** capacidad del sistema de cargar un programa en memoria y ejecutarlo.
- **Operaciones de E/S:** dado que los programas no pueden ejecutar operaciones de E/S directamente, el sistema operativo debe proveer medios para realizarlas.
- **Manipulaciones del sistema de archivos:** capacidad de los programas de leer, escribir, crear y borrar archivos.
- **Comunicaciones:** intercambio de información entre procesos ejecutando en la misma computadora o en otra conectada a través de una red. Implementada mediante memoria compartida o por paso de mensajes.
- **Detección de errores:** asegurar un cómputo correcto mediante la detección de error en el CPU, memoria, dispositivos de E/S o en los programas de usuario.

Componentes del Sistema Operativo

- Administración de procesos
- Administración de memoria principal
- Administración de memoria secundaria
- Administración de dispositivos de E/S
- Administración de archivos
- Sistema de protección
- Redes
- Sistema de interprete de comandos

Componentes del Sistema Operativo



- Los distintos componentes **interactúan** unos con otros

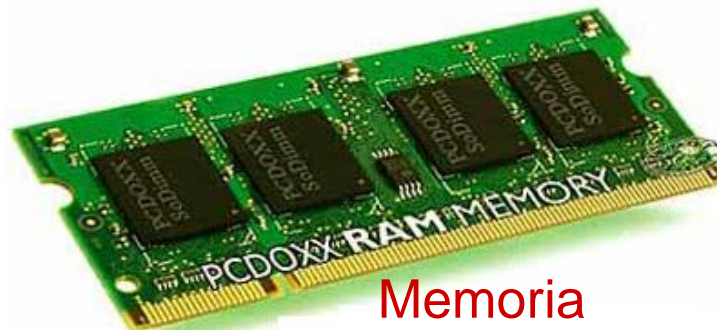
1. Administración de procesos

- Un **proceso** es un **programa en ejecución**. Un proceso necesita ciertos **recursos**, incluyendo:
 - ✓ tiempo de CPU
 - ✓ Memoria
 - ✓ Archivos y
 - ✓ dispositivos de E/Spara lograr su tarea.
- El **sistema operativo** es responsable de las siguientes actividades en la administración de procesos:
 - ✓ **creación** de procesos y borrado.
 - ✓ **suspensión** y **reanudación** de procesos.
 - ✓ mecanismos de: **sincronización** de procesos **comunicación** entre procesos



2. Administración de Memoria Principal

- La memoria es un **array de words** y **bytes**, cada uno con su dirección propia. Es un repositorio de datos de rápido acceso compartido por el CPU y los dispositivos de E/S
- La **memoria principal** es un dispositivo de almacenamiento volátil. Pierde su contenido en caso de interrupción del sistema y falla.
- El **sistema operativo** es responsable de las siguientes actividades respecto a la administración de memoria :
 - ✓ Llevar cuenta de que partes de la **memoria** están siendo **utilizadas** y por quien.
 - ✓ Decidir que procesos **cargar en memoria** cuando este está disponible.
 - ✓ Alojar y desalojar espacio de memoria cuando sea requerido.



Memoria

3.Administración de memoria secundaria

- Dado que la **memoria principal** es volátil y muy pequeña para almacenar todos los datos y los programas en forma permanente, el sistema de cómputo provee **memoria secundaria** para respaldar la memoria principal.
- La mayoría de las computadoras modernas utilizan **discos** como medio de almacenamiento en línea, para programas y datos.
- El **sistema operativo** es responsable de las siguientes actividades respecto a la administración de la memoria secundaria:
 - ✓ Administración del espacio de memoria libre.
 - ✓ Alojamiento de memoria
 - ✓ Planificación del disco



Memoria
Secundaria

Memoria virtual y MMU

- La **memoria virtual** combina la RAM del equipo con espacio temporal en el disco duro. Cuando queda poca RAM, la memoria virtual mueve datos de la RAM a un espacio llamado **archivo de paginación**. Al mover datos al archivo de paginación y desde él, se libera RAM para que el equipo pueda completar la tarea.
- Si el **sistema operativo**, con asistencia de un hardware llamado MMU (o memory-management unit), proveen memoria virtual, entonces cada uno de los '**n**' procesos puede tener su propio espacio de memoria contigua de tamaño **M**, sin que sea necesario tener una memoria de hardware (o sea, memoria física) de tamaño $n \times M$ (a menos que los procesos llenen sus memorias).

4. Administración de entradas y salidas

- El sistema de entradas y salidas consiste de:
 - ✓ un sistema de buffer-caching
 - ✓ Una interface general de drivers de dispositivos
 - ✓ Drivers para dispositivos específicos

5. Administración de archivos

- Un **archivo** es una colección de información relacionada definida por su creador. Comúnmente, los archivos representan **programas** (en formato fuente y objeto) y **datos**.
- El sistema operativo es responsable de las siguientes actividades respecto a la administración de archivos:
 - ✓ **Creación** de archivos y **borrado**.
 - ✓ **Creación** de directorio y **borrado**.
 - ✓ Soporte de primitivas de **manipulación de archivos** y directorios.
 - ✓ **Mapeo** de archivos en memoria secundaria.
 - ✓ Respaldo de archivos en un medio de almacenamiento estable (no-volátil).



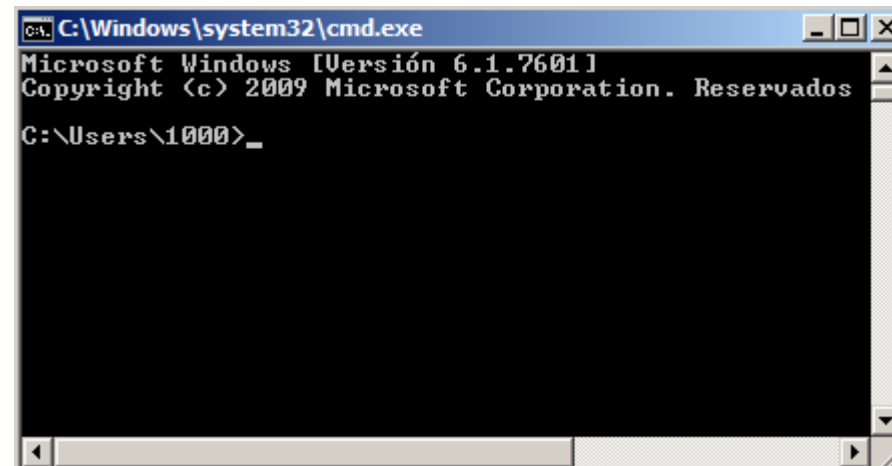
6. Sistema de Protección

- La protección es un mecanismo control de acceso de los programas, procesos o usuarios al sistema o recursos.
- El sistema de protección debe:
 - ✓ distinguir entre usos autorizados y no-autorizados.
 - ✓ especificar el tipo de control de acceso impuesto.
 - ✓ proveer medios para el aseguramiento de la protección.



7. Sistema de Interprete de comandos

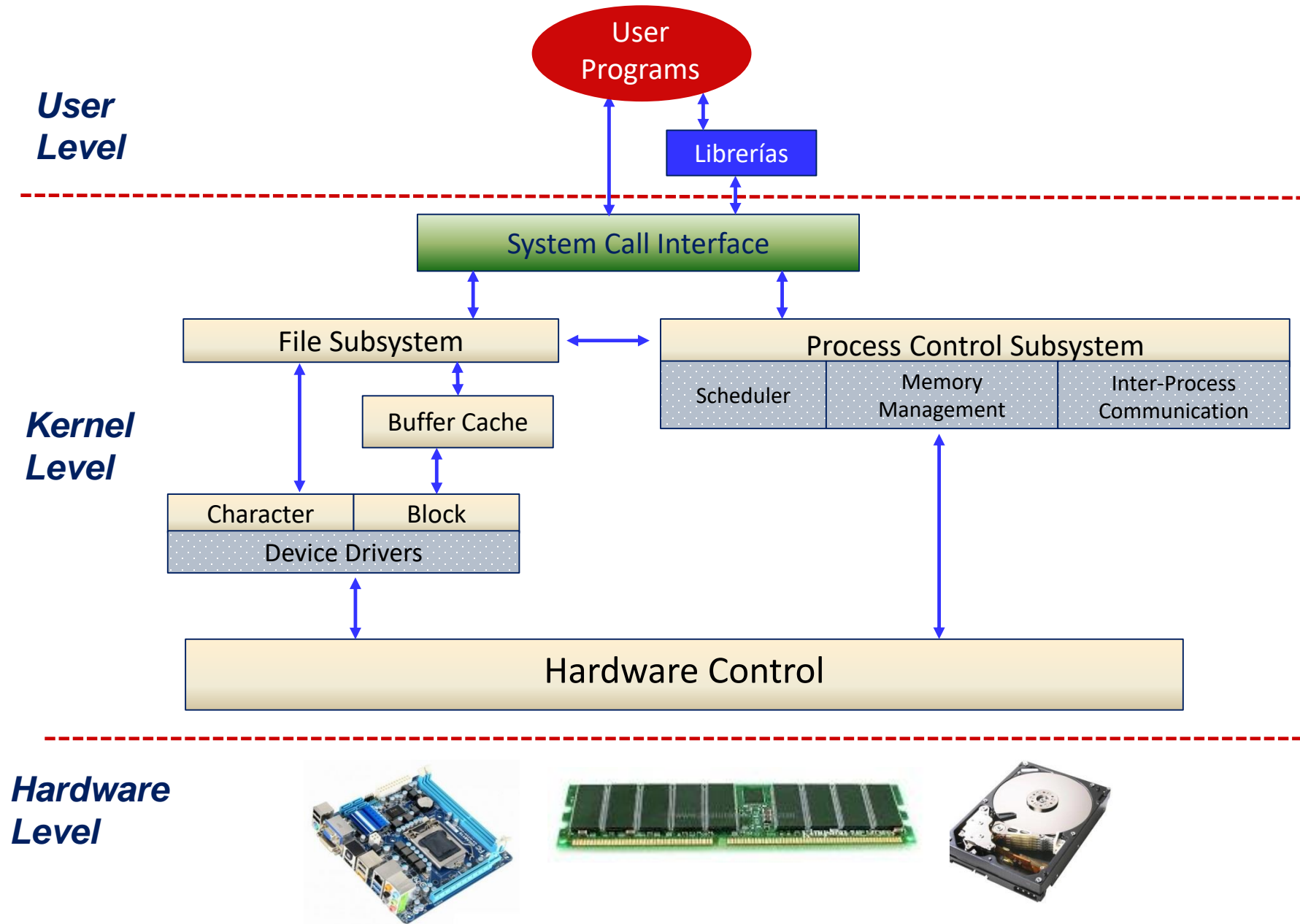
- Muchos comandos son proporcionados por el sistema operativo mediante estatutos de control que tienen las siguientes funciones
 - ✓ creación de procesos y administración
 - ✓ administración de E/S
 - ✓ administración de almacenamiento secundario
 - ✓ Administración de memoria principal
 - ✓ acceso al sistema de archivos
 - ✓ protección
 - ✓ acceso a la red



Estructura del sistema

- ✓ Nivel 5: programas de usuario
- ✓ Nivel 4: buffering para dispositivos de E/S
- ✓ Nivel 3: driver de dispositivo para la consola del operador
- ✓ Nivel 2: administración de memoria
- ✓ Nivel 1: administración del CPU
- ✓ Nivel 0: hardware

Estructura del sistema



Llamadas al sistema (System Call)

- Los llamados al sistema (system calls) son una interfaz, provista por el núcleo, para que los procesos de usuarios accedan a los diferentes servicios que brinda el sistema operativo.
- Los servicios son invocados por los procesos en modo usuario, cuando ejecutan lo hacen en modo monitor, y al retornar vuelven al modo usuario.
- Típicamente a los system calls se les asocia un número que los identifica (en Linux son aproximadamente 350).

La arquitectura del sistema operativo

¿Cuál es la mejor arquitectura de sistema operativo?

- Este debate produjo un enconado encuentro en el grupo de noticias de MINIX, entre Andy Tanenbaum y Linus Torvalds, padre de LINUX.
- A continuación listamos las arquitecturas clásicas de sistema operativo.

✓ La arquitectura plana

✓ La arquitectura monolítica: Windows, Linux

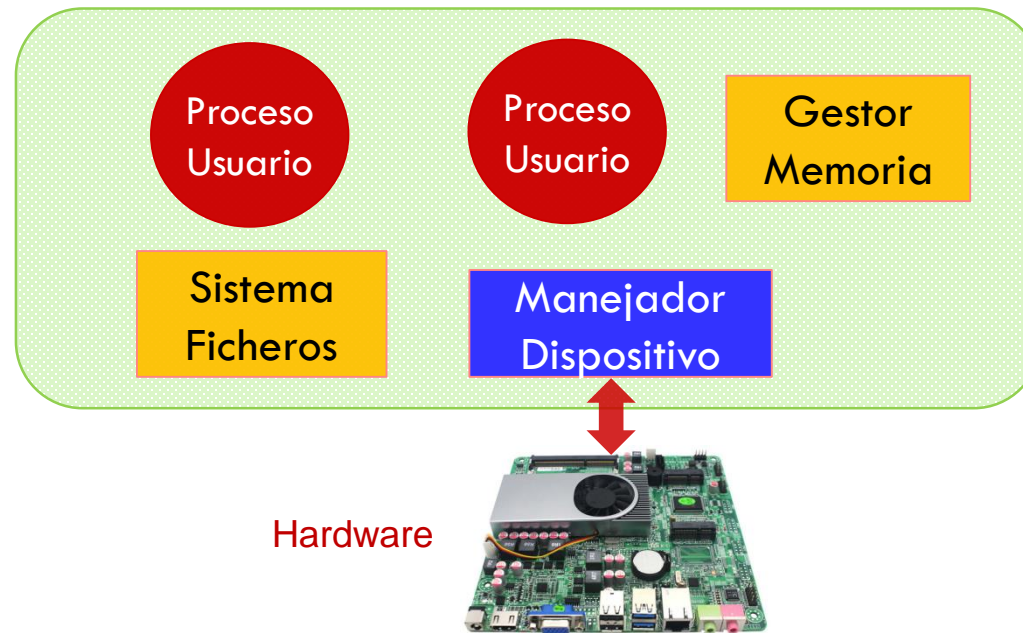
✓ La arquitectura microkernel: MINIX y QNX

La arquitectura del sistema operativo

Arquitectura plana.

- Adoptada tradicionalmente por muchos sistemas de tiempo real.
- Integra todos los **componentes de la aplicación** en el mismo espacio de direccionamiento que el **sistema operativo**.

Sistema en tiempo real



La arquitectura del sistema operativo

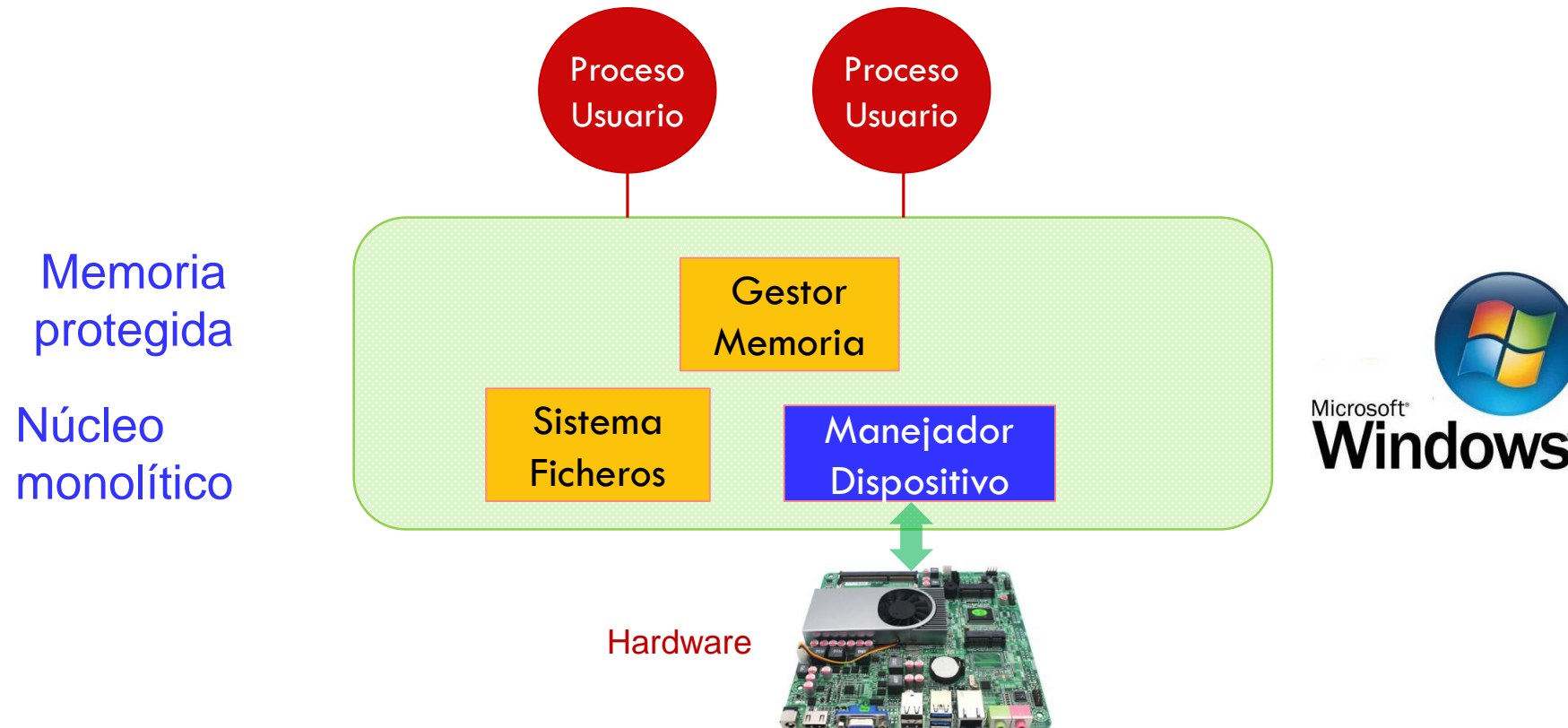
Arquitectura plana

- Suele emplearse en sistemas **empotrados pequeños** o procesadores **sin facilidades de protección de memoria**, como procesadores digitales de señal (DSPs).
- La fiabilidad de todo él depende de cada nuevo componente. La **carencia de protección** conlleva el que si un proceso falla, puede escribir sobre código o datos del núcleo, de modo que todo el sistema se corrompe. Además, en cuanto el sistema crece, la probabilidad del fallo se multiplica.
- Cada fallo es difícil de localizar. Incluso las mejores herramientas pueden ser incapaces de aislarlo y es preciso que una persona conozca bien todo el sistema completo.

La arquitectura del sistema operativo

Arquitectura monolítica: Windows

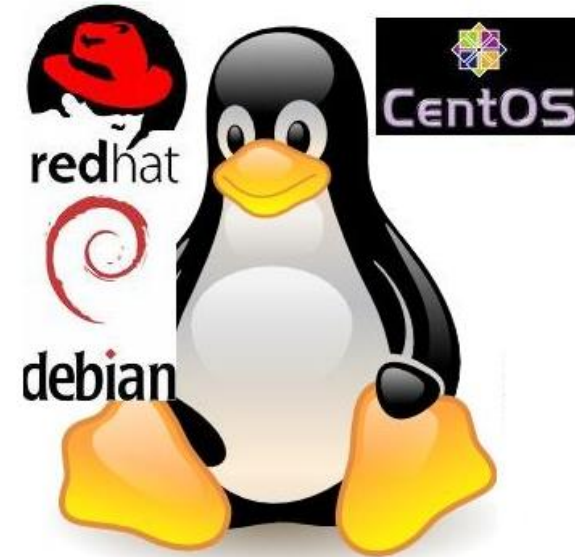
- Es un intento de paliar los problemas de la arquitectura plana
- Su aportación estriba en que los procesos de usuario ejecutan en espacios de direccionamiento diferentes al del sistema operativo



La arquitectura del sistema operativo

Arquitectura monolítica: Linux

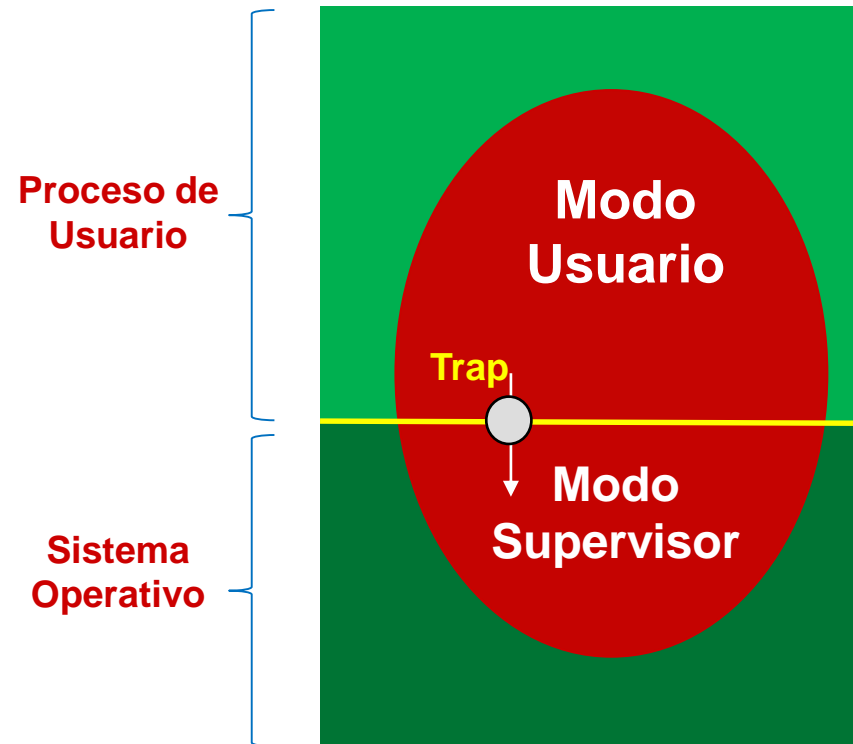
- Las implementaciones de UNIX han respondido tradicionalmente este diseño
- Es una colección de procedimientos compilados en un único programa objeto donde cada procedimiento es visible a todos los demás.
- Se aísla al sistema de los errores de los procesos de usuario, pero nuevos dispositivos aparecen en el mercado continuamente y es preciso escribir manejadores para soportarlos. De nuevo el sistema crece y la probabilidad del fallo aumenta.



La arquitectura del sistema operativo

Arquitectura monolítica: Linux

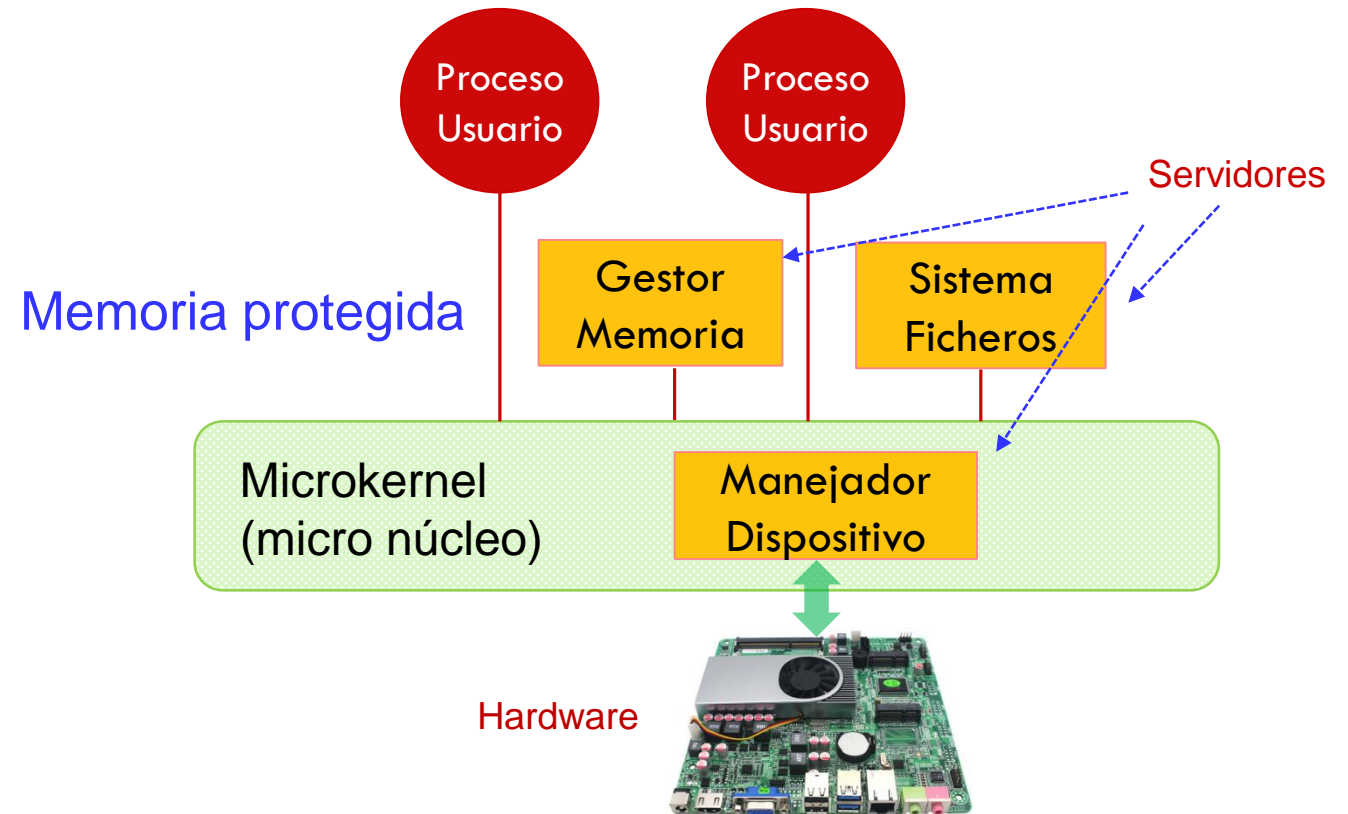
- El programa de usuario lleva a cabo las **llamadas al sistema** mediante **interrupciones software** o *traps*.
- ¿Por qué es necesaria una instrucción de interrupción software y no una instrucción de salto? Debido a que esta instrucción de interrupción software conmuta el procesador de *modo usuario a modo supervisor*.
- Sólo en **modo supervisor** se permite al procesador ejecutar **instrucciones privilegiadas**
 1. Accesos a las posiciones de memoria asignadas a los adaptadores de dispositivo
 2. Copia de datos entre espacios de direccionamiento diferentes.



La arquitectura del sistema operativo

Arquitectura micronúcleo: MINIX

- La tendencia en los sistemas operativos modernos es disponer de **un núcleo lo más pequeño posible**, procurando que algunas de sus funciones (**cuantas más mejor**) sean desplazadas fuera del mismo y sean implementadas como programas de usuario.
- Este nuevo núcleo reducido se viene denominando **microkernel**.



La arquitectura del sistema operativo

Arquitectura micronúcleo: MINIX

- MINIX hace uso del paradigma cliente-servidor.
- El sistema de ficheros y el gestor de memoria se programan como servidores
- Los manejadores de los dispositivos también toman la forma de servidores; no obstante, se mantienen integrados en el núcleo para facilitar su acceso a los dispositivos físicos.
- La labor del microkernel es soportar el paso de mensajes entre clientes y servidores
- Una de las ventajas de un sistema microkernel es que las partes se dividen por fronteras bien definidas, por lo que resultan más comprensibles y manejables.
- Una segunda ventaja es su idoneidad para construir sistemas distribuidos,

La arquitectura del sistema operativo

Arquitectura micronúcleo: QNX

- QNX, como MINIX, construye los manejadores de dispositivo como procesos, pero con la ventaja adicional de que los desplaza fuera del núcleo en espacios de direccionamiento propios. Este es el diseño más fiable de todos porque es el que proporciona mayor protección

