

## Actividad 1: Introducción a los Sistemas operativos

### 1. Diferencias en la Definición del Sistema Operativo: Sentido Amplio y Sentido Estricto

El término "sistema operativo" puede entenderse de dos formas principales, dependiendo del nivel de detalle y del contexto en el que se utilice:

#### 1. Definición Amplia y Habitual:

En su concepción más general, un sistema operativo incluye no solo el núcleo (kernel), sino también un conjunto más amplio de herramientas, utilidades y aplicaciones que permiten a los usuarios interactuar de manera efectiva con el hardware y los recursos del sistema. Esto abarca componentes como administradores de archivos, intérpretes de comandos (shells), sistemas de archivos, bibliotecas estándar, configuradores como paneles de control, y utilidades adicionales como terminales, editores de texto y otros programas que forman parte del entorno del sistema operativo. En este sentido, el sistema operativo es visto como un conjunto de aplicaciones accesibles al usuario, que le proporcionan una experiencia completa para la interacción con su equipo.

#### 2. Definición Técnica y Estricta:

Desde un enfoque más técnico, el término "sistema operativo" se refiere exclusivamente al núcleo (kernel), que es el componente más fundamental y esencial del software del sistema. Este núcleo opera de manera invisible para el usuario, ya que no es algo con lo que se interactúa directamente, sino que es el intermediario entre el hardware y las aplicaciones o procesos. El núcleo se encarga de tareas críticas como la gestión de la memoria, la asignación de tiempo de procesador, la comunicación entre procesos y el acceso a los dispositivos de hardware. Por tanto, el usuario no lo percibe directamente, pero su correcto funcionamiento es vital para que las aplicaciones puedan ejecutarse y brindar los servicios que el usuario utiliza.

En el presente módulo, el término "sistema operativo" hará referencia a la definición más técnica, es decir, al núcleo o kernel. En caso de que sea necesario referirse al concepto más amplio, esto será indicado de manera explícita para evitar confusiones.

## 2. Roles y funciones del Sistema Operativo

El sistema operativo cumple dos roles esenciales que están estrechamente relacionados con sus funciones principales:

### 1. Administrador de Recursos: Se explica a si mismo

El sistema operativo actúa como un referee o árbitro entre las aplicaciones (o procesos) que compiten por los recursos del sistema. Esto es fundamental para garantizar que los recursos, como la CPU, la memoria y los dispositivos de entrada/salida, sean gestionados de manera eficiente y sin conflictos. Por ejemplo, cuando dos procesos necesitan utilizar la impresora, el sistema operativo se encarga de decidir cuál tendrá acceso primero, informando al segundo proceso que debe esperar a que la impresora se desocupe. De manera similar, si un proceso solicita más memoria RAM, el sistema operativo verifica qué direcciones de memoria están disponibles, asigna la memoria solicitada, y registra esa asignación para poder reutilizarla posteriormente cuando sea necesario. Esta capacidad de gestionar y coordinar los recursos asegura que múltiples aplicaciones puedan ejecutarse simultáneamente sin interferencias.

### 2. Máquina Extendida:

Más allá de administrar recursos, el sistema operativo proporciona una capa de abstracción entre el hardware y los procesos, simplificando las interacciones y ofreciendo una interfaz uniforme. Esta abstracción permite que las aplicaciones realicen solicitudes de alto nivel al sistema operativo, en lugar de lidiar con la complejidad de las operaciones de bajo nivel del hardware. Por ejemplo, cuando un proceso necesita almacenar datos en un disco, no tiene que preocuparse por controlar qué sectores están ocupados y cuáles están disponibles, o cómo El SO traduce a bajo nivel lo que le dicen los programas que necesitan hacer

comunicarse directamente con la controladora del disco. Sin el sistema operativo, cada programador tendría que implementar esta funcionalidad compleja en sus aplicaciones.

Gracias al sistema operativo, las aplicaciones pueden hacer solicitudes abstractas como "guardar este texto en tal archivo", dejando que el sistema operativo se encargue de traducir esta solicitud a las operaciones de bajo nivel necesarias para comunicarse con el hardware, como dividir los datos en bloques y escribirlos en sectores específicos del disco. Este concepto de "máquina extendida" es clave, ya que proporciona una interfaz más sencilla y práctica para los desarrolladores y usuarios.

Es importante destacar que conceptos como el de "archivo" no existen en el hardware; son abstracciones creadas por el sistema operativo para facilitar el manejo de los datos. En lugar de que las aplicaciones trabajen directamente con bloques o sectores, el sistema operativo permite que interactúen con archivos y carpetas, brindando un modelo más intuitivo y eficiente para los procesos.

En resumen, estos roles explican por qué es indispensable que una PC tenga un sistema operativo: por un lado, necesitamos un gestor que arbitre las solicitudes de recursos entre procesos y, por otro, un sistema que simplifique las interacciones con el hardware, proporcionando una interfaz abstracta y uniforme que facilite tanto el desarrollo como el uso de aplicaciones.

Dentro de este marco, el núcleo del sistema operativo desempeña las siguientes funciones específicas:

**1. Gestión de Procesos:** Supervisa la creación, suspensión y finalización de procesos, garantizando su comunicación y sincronización. Actúa como mediador entre los procesos y los recursos disponibles.

**2. Gestión de Memoria:** Controla la asignación y liberación de memoria, optimizando el uso de la RAM y proporcionando memoria virtual para expandir las capacidades del sistema.

**3. Control de Dispositivos:** Facilita la interacción entre el hardware y el software mediante controladores, que son abstraídos y gestionados por el núcleo para ofrecer acceso uniforme a los dispositivos.

**4. Gestión del Sistema de Archivos:** Organiza y proporciona acceso eficiente a los datos almacenados en dispositivos de almacenamiento.

**5. Seguridad y Protección:** Implementa barreras para evitar accesos no autorizados, asegurando la integridad del sistema y los datos.

### 3. Espacios del Kernel y de Usuario

Una forma de distinguir qué procesos forman parte del sistema operativo es identificar dónde se ejecutan y cómo interactúan con los recursos del sistema. Esta diferenciación está estrechamente vinculada con los modos de ejecución del procesador, que son el modo kernel y el modo usuario:

• **Modo Kernel:** En este modo, las instrucciones en código máquina tienen acceso directo tanto al hardware como a toda la memoria RAM. Esto significa que las operaciones críticas, como gestionar dispositivos o asignar memoria, pueden llevarse a cabo sin restricciones. El espacio del kernel es donde se ejecutan los procesos que forman parte del sistema operativo, como el núcleo (kernel) y los controladores esenciales. Este espacio es exclusivo y está reservado para tareas que requieren control total sobre el sistema.

• **Modo Usuario:** En este modo, las instrucciones del procesador no tienen acceso directo al hardware ni a la memoria del sistema, lo que protege la estabilidad y seguridad del sistema. Los procesos que operan en el espacio de usuario son considerados "normales" y deben realizar solicitudes al sistema operativo, conocidas como llamadas al sistema, para ejecutar operaciones privilegiadas como leer un archivo, asignar memoria o enviar datos a un dispositivo.

En este sentido, otro modo de definir un sistema operativo, además de por sus funciones, es como aquel software que se ejecuta en el espacio del kernel. Esto

incluye el núcleo y otros componentes esenciales que interactúan directamente con el hardware para gestionar recursos y garantizar el funcionamiento correcto del sistema.

En resumen, los espacios del kernel y de usuario y sus respectivos modos de ejecución del procesador son fundamentales para entender cómo el sistema operativo controla el acceso al hardware y mantiene la seguridad del sistema. En el modo kernel, las instrucciones tienen control total sobre los recursos, mientras que en el modo usuario, los procesos están restringidos y dependen del sistema operativo para realizar operaciones más complejas. Esta arquitectura protege al sistema de errores y fallos potenciales, permitiendo un entorno controlado y seguro para las aplicaciones.

#### 4. Tipos de Sistemas Operativos y Relación con los Modos de Operación

Los sistemas operativos pueden clasificarse según cómo distribuyen sus componentes entre el espacio del kernel y el espacio de usuario, lo que influye directamente en su diseño y funcionamiento:

##### 1. Monolíticos:

- El núcleo incluye todas las funciones esenciales y muchas no esenciales del sistema operativo. Todo el sistema operativo opera en el espacio del kernel, lo que proporciona alta eficiencia pero menos modularidad.

- Ejemplo: Linux.

##### 2. Microkernels:

- Solo las funciones más básicas (gestión de procesos, memoria y comunicación) residen en el espacio del kernel. Las demás funciones operan en el espacio de usuario, lo que mejora la modularidad y la seguridad a costa de un rendimiento ligeramente menor.

- Ejemplo: Minix, QNX.

### 3. Híbridos:

- Combinan características de núcleos monolíticos y microkernels.

Mantienen funciones críticas en el espacio del kernel, pero trasladan algunas funciones adicionales al espacio de usuario para lograr un balance entre eficiencia y modularidad.

- Ejemplo: Windows NT, macOS.

Al observar cómo los procesos y funciones se distribuyen entre el espacio del kernel y el espacio de usuario, se obtiene una visión clara de las diferencias entre los diseños de sistemas operativos y su impacto en la seguridad, modularidad y eficiencia.