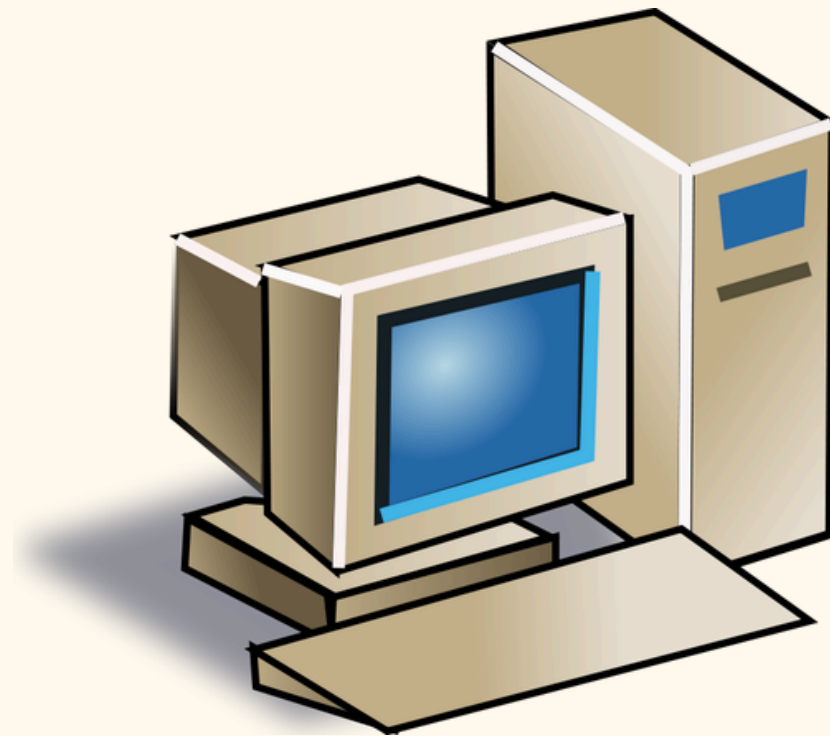


Funcionamiento del sistema operativo

Pedro David Llerenas Gonzalez



Introducción

El monitor de un ordenador, la pantalla de una laptop, el ratón, el teclado, entre otros, son ejemplos de *hardware*. Por otro lado, el *software* es aquel componente no físico con el cual podemos interactuar por medio del *hardware*. Por ejemplo, el navegador que utilizamos, una calculadora, un editor de texto, etc. Sin embargo, estos no (siempre) interactúan directamente entre sí. Es decir, hay un componente no físico que conecta estos conceptos.

El *sistema operativo* es un tipo de software (específicamente, software de sistema) que permite la comunicación entre el hardware y otro pedazo de software. Hay otros softwares que interactúan directamente con el hardware, como pueden ser los drivers. Estos se encargan de las solicitudes y envío de información de, por ejemplo, un ratón.

A lo largo de los años, los sistemas operativos han ido evolucionando. Lo podemos observar por medio de Android, el sistema operativo más utilizado en dispositivos móviles (“Mobile Operating System Market Share Worldwide | Statcounter Global Stats”). La calidad y velocidad en la que se puede manipular el contenido ha cambiado drásticamente. Estas mejoras han sido gracias a la implementación de la *multiprogramación*, que consiste en compartir el CPU (y GPU, si existe) y la RAM de manera equitativa entre los programas. Es decir, cada programa tiene cierta cantidad de tiempo para alterar ciertas partes del hardware, permitiendo tener diversas aplicaciones corriendo al mismo tiempo sin que ninguna se detenga esperando el proceso de otras. Notemos que esto también trae desventajas, ya que como la memoria se comparte, puede ocurrir que otros programas intenten modificar la memoria utilizada por otra aplicación (Licklider).

El sistema operativo

Un sistema operativo es un tipo de software que permite al usuario utilizar los recursos del hardware por medio de otro programa. En general, un software está hecho para un sistema operativo en específico, ya que no todos hacen la comunicación entre la aplicación y el hardware de la misma manera. Podemos ver esto cuando intentamos descargar alguna aplicación, y nos muestran opciones dependiendo de nuestro sistema operativo. Ya sea Linux, Windows, o Mac.

El sistema operativo también es aquel que se encarga de organizar la manera en que las aplicaciones interactúan con el hardware. La eficiencia en la que lo hace depende puramente en la manera que fue implementada por los creadores del sistema operativo. Es decir, puede haber diferencia en la manera que Windows y Linux deciden separar los tiempos de ejecución de dos programas.

A pesar de que las implementaciones sean distintas, hay tres niveles esenciales de un sistema operativo.

1. **Kernel:** la parte del sistema operativo que se encarga de recibir y dirigir información directamente del hardware. Este es el que tiene permiso de leer y escribir en la RAM o disco duro. También maneja el flujo de información que puede llegar de un dispositivo como el ratón o un teclado.
2. **Interfaz de usuario:** la parte nos permite interactuar de manera sencilla con el ordenador. Es decir, los pixeles que forman un botón, el cursor, los iconos que nos dicen que acción se realiza al darle clic.

3. **Interfaz de programación de aplicaciones (API):** lo que permite la conexión entre varios dispositivos, por medio de funciones que exponen o reciben información de manera estructurada para el procesamiento interno de cada dispositivo. Por ejemplo, cuando presionamos una tecla, el API es el que se encarga de enviar exactamente qué tecla se presionó, sea como sea la estructura de la información, otro programa recibe esa información, y realiza alguna acción, como puede ser, mover un personaje dentro de un juego.

Funciones básicas del sistema operativo

- **Bootear:** el proceso de preparar el hardware para ser manipulado por el usuario, y asegurarse que está en buen estado para ser utilizado.
- **Seguridad de datos:** el sistema operativo intenta prevenir las acciones maliciosas de una aplicación tratando de corromper información que no le pertenece. Los *virus* son aquellos que logran burlar el sistema operativo, dándoles control de la memoria y/o procesos que puede realizar.
- **Manejo de disco duro:** el disco duro es donde se almacenan los recursos que frecuentamos y que son esenciales para correr una aplicación.
- **Proveedor de interfaz:** los iconos que observamos e interactuamos cuando instalamos un sistema operativo por primera vez vienen directamente del mismo. Anteriormente, las interfaces eran simplemente la consola. Con las mejoras de hardware a lo largo de los años, el sistema operativo ha sido capaz de utilizar más recursos para mejorar la experiencia de usuario.

Tipos de sistemas operativos

Existen 5 principales tipos de sistemas operativos. Cada uno con una especialidad distinta.

- **De uno y varios procesos:** un sistema operativo de un proceso solo puede correr un programa a la vez, mientras que el alterno tiene la capacidad de correr diversos al mismo tiempo. Los de varios procesos se caracterizan en *preventivo* y *cooperativo*. El preventivo parte el CPU, dedicando una pieza a cada programa. Los cooperativos le proporcionan un tiempo limitado a cada programa.
- **Uni- y multi-usuario:** los sistemas de un usuario no distinguen entre usuarios, mientras que los multiusuario permiten distintas sesiones y personalización única por usuario.

- **Distribuido:** el sistema operativo se encuentra descentralizado. En este caso, un proceso es realizado hasta su finalización. Ejemplos comunes son los servicios de Amazon de la nube.
- **Embebido:** diseñados principalmente para su uso en máquinas pequeñas (o móviles) con cantidades de memoria menores a las de un ordenador. La eficiencia en estos sistemas operativos es crucial.
- **De tiempo real:** son sistemas que responden al input de manera inmediata. Ejemplos pueden ser coches autónomos.

Abstracciones de un sistema operativo

Para prevenir los posibles riesgos de seguridad que trae correr distintas aplicaciones simultáneamente, los sistemas crean una abstracción de la memoria real. Esto hace que cada programa vea una versión de la memoria, sin revelar exactamente qué hay detrás de ella, y se prepara para las lecturas de memoria mediante la *virtualización*, que le presenta al programa direcciones que no corresponden exactamente al lugar en el hardware que se encuentra. El *heap* se utiliza para asignar memoria de manera dinámica y la que maneja el usuario. Por ejemplo, cuando usamos `malloc()` en C, aquí es donde se hace espacio para lo que necesitamos.

Conclusiones

Los sistemas operativos realizan tareas importantes que nos permiten realizar manejar software y hardware. Las implementaciones de estos son increíblemente robustas, y deben ser cuidadosamente programadas para la protección de información. Esto lo logra mediante la abstracción de la memoria real, negando la vista real a los programas que intenten leer o escribir en memoria que no les pertenece.