

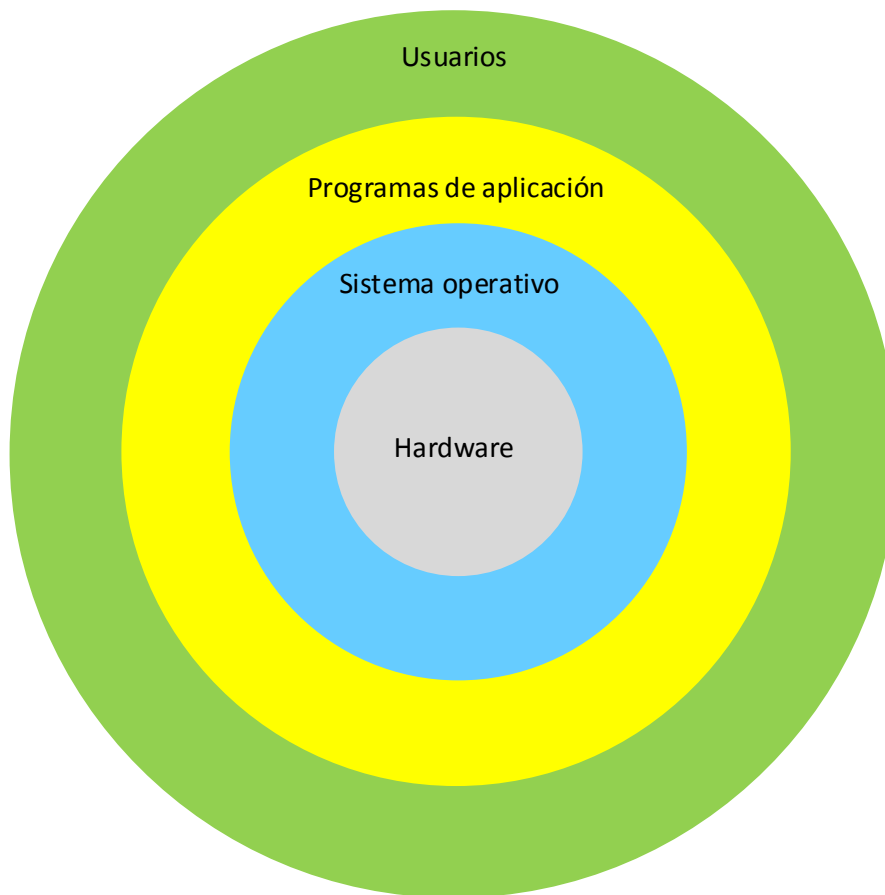
Sistemas Operativos

(Consultar [P. Norton, Introducción a la Computación](#))

Antiguamente llamados Sistemas Administrativos (porque esa es la idea, administrar los recursos de hardware y software de un sistema de procesamiento de datos).

Es un conjunto de programas que guían a una computadora en el cumplimiento de sus tareas, auxilian a los programadores y programas con funciones de apoyo e incrementan la utilidad del hardware.

Los Sistemas Operativos son la interfaz entre los usuarios y el hardware y entre los programas de aplicación y el hardware:



Cómo se realiza el proceso de la carga del Sistema Operativo

Cuando se enciende una computadora, ésta pasa por varias etapas antes de poder ser utilizada:

Primer Paso: [Autocomprobación](#). Se identifican los dispositivos físicamente conectados, la cantidad de memoria disponible y si ésta está en buen estado. Esto es realizado por un programa localizado en dispositivos de memoria ROM, llamado la rutina POST.

Segundo Paso: [Búsqueda](#) de un [programa de inicio](#) en la computadora, que verifica si un SO está presente e inicia su proceso de carga. Este programa, el Master Boot Record (MBR), suele encontrarse en el primer sector del disco duro, en la pista más externa.

Tercer Paso: el programa de inicio **transfiere** el control a una porción del SO para finalizar la carga.

Tipos de sistemas operativos

- a) Monousuario: atiende solo a un usuario.
- b) Multiusuario: atienden los procesos de muchos usuarios simultáneamente.
- c) Multitarea: tiene la capacidad de atender múltiples tareas simultáneamente, no es necesariamente multiusuario.

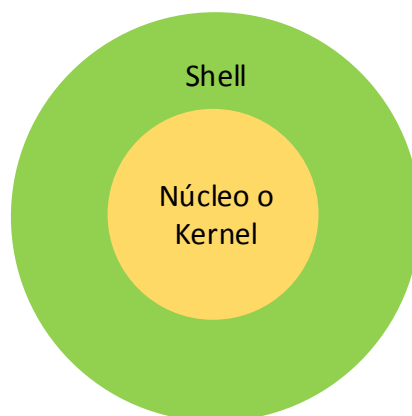
Por ejemplo, los sistemas operativos actuales de PCs, que tenemos en nuestras casas, son monousuario y multitarea. En el pasado, el sistema operativo DOS, era monousuario y no multitarea.

Tareas (Funciones) de un sistema operativo

1. Interpreta los comandos del operador, que describen el trabajo que debe realizarse.
Los comandos se pueden introducir desde la "línea de comandos" o desde "consola" en forma textual, o bien pueden ser introducidos con ayuda de la interfaz gráfica del sistema operativo. Estos son capturados e interpretados por el so y luego deben ser ejecutados. Ejemplo: mostrar el directorio (contenido de una carpeta), formatear un disco o una memoria usb, mostrar la fecha y hora del sistema, borrar un archivo, etc.
2. Da control de la computadora a los programas de aplicación y maneja la transición entre una tarea y otra.
Los programas pueden hacer uso de los recursos de hardware por medio del so. Cuando un programa necesita, por ejemplo, escribir en un archivo en disco, es el so el que se encarga de la comunicación y de la transferencia de datos hacia el dispositivo físico.
3. Programa y desarrolla cronológicamente las funciones de entrada y salida.
Un programa podría estar requiriendo del uso de un dispositivo, por ejemplo, para salida de datos, pero debe esperar a que este dispositivo termine de ser utilizado por una requisición previa de otro programa. El so administra los trabajos de impresión por medio del spool de impresora. Algo similar sucede con el uso de los demás dispositivos cuando varias aplicaciones quieren hacer uso simultáneo de ellos.
4. Asigna memoria y otros recursos en un medio de multiprogramación.
Multiprogramación se refiere a correr múltiples programas a un mismo tiempo. Cuando se va a iniciar la ejecución de un programa, el so le asigna una zona de memoria para ser cargado. Lo mismo sucede a medida que un programa va requiriendo más espacio de memoria para contener sus datos (memoria dinámica) (o al principio: memoria estática).
5. Impide que los programas interfieran entre ellos.
Se puede ver como una consecuencia de punto anterior. Por ejemplo, el so debe evitar que dos o más programas utilicen la misma zona de memoria, o el recurso de impresora al mismo tiempo, o el uso de la cpu en un medio de multiprogramación.

6. Proporciona servicios de depuración y diagnóstico de error.
Mensajes mandados a consola como respuesta a los comandos introducidos: de operaciones exitosas o fallidas; archivos de texto donde registra las operaciones diarias (login), para seguimiento; etc.
7. Asigna dispositivos de entrada y salida a los archivos lógicos que hacen referencia los programas.
Los archivos en disco son manejados por las aplicaciones por medio de flujos de Entrada/Salida. Esto es, pequeños buffers o zonas de memoria asignadas por el so cuando se abre un archivo desde un programa en ejecución.
8. Administra la forma en cómo se almacenan y se recuperan los datos.
Por ejemplo, los programadores indican por medio de sus instrucciones, que se necesita almacenar o recuperar información del disco. Pero el so es el que lleva estas operaciones a bajo nivel (mediante las rutinas del BIOS), operaciones que están relacionadas directamente con el hardware. Es el so quien conoce el tipo de formato del dispositivo de almacenamiento.
9. Hace efectiva la disciplina necesaria para correr muchos programas simultáneos en un medio de tiempo compartido.
En un medio de tiempo compartido cada programa utilizará la cpu en una fracción de tiempo. Cada vez que la cpu deba soltar un programa y tomar el siguiente, hay un intervalo de tiempo en el que debe guardar el "estado" en el que se encontraba la cpu al momento de soltar cada programa, para luego reanudar cuando lo retome nuevamente. Esto es similar a cuando una persona tiene que dejar de hacer una actividad para hacer otra y luego continuar con la anterior.
10. Asigna tareas a los procesadores en un medio de multiprocesamiento.
Un medio de multiprocesamiento es un sistema donde hay varias cpu (procesadores). En estos casos los sistemas operativos tienen un diseño especial para poder correr en estos ambientes. Los so de estos entornos deben tener la capacidad de poder efectuar la distribución de tareas hacia las diferentes cpu.

Componentes de un sistema operativo



El sistema operativo Unix tiene dos componentes fundamentales:

- El Núcleo (Kernel en Unix – Linux).
- El Shell.

EL KERNEL constituye el núcleo del sistema operativo. Actúa como [interfaz con el hardware del sistema](#). Se encarga de realizar las [funciones de más bajo nivel](#), como son:

1. Manejo y control de los dispositivos físicos.
2. Control de interrupciones.
3. Control de procesos y tiempos.
4. Control de archivos.
5. Control de memoria.

Está escrito fundamentalmente en lenguaje "C" (90%) y el resto en lenguaje ensamblador.

EL SHELL

- Es el intérprete de comandos.
- Proporciona una serie de comandos a través de los cuales el usuario puede interactuar con el sistema operativo.
- Funciona como interfaz del sistema operativo con los usuarios.
- Puede interactuarse con el Shell por medio de una ventana especial conocida como la terminal, o a través del entorno gráfico que proporciona el sistema operativo.
- El sistema operativo permite la creación de archivos de lotes de comandos para realizar tareas específicas: los archivos bash en Linux o batch en Windows. Al ejecutar estos archivos, se ejecutan estos comandos, uno tras otro, lo que permite realizar la tarea. En Linux, estos archivos se conocen como shellscripts.

Clasificación de los comandos del Sistema Operativo

Los comandos de un SO se clasifican en: Comandos [Internos](#) y Comandos [Externos](#).

En el proceso de carga, una porción del SO es cargada a memoria y otra permanece en el medio de almacenamiento.

Comandos internos: son los comandos que se cargan a memoria y permanecen allí durante el tiempo que la computadora esté encendida. La ejecución de este tipo de comandos es más rápida.

Comandos Externos: son los comandos que permanecen en el medio de almacenamiento. La ejecución de este tipo de comandos es más lenta puesto que, al ser invocados, deben ser buscados primero en el dispositivo físico, cargados a memoria y por último, ejecutados.

¿Cómo hacen los diseñadores de los SO para definir cuáles comandos serán internos y cuáles serán externos? A través del tiempo, los repertorios de comandos de los sistemas operativos crecen. Así también, comandos que en versiones previas eran externos, se vuelven internos en versiones posteriores. Estos cambios los realizan los diseñadores de los SO dependiendo de la frecuencia de uso y demanda de los usuarios. Mientras más frecuentemente se utilice determinado comando, será más necesario que sea interno, ya que esto incrementa la eficiencia del sistema para responder a las necesidades de los usuarios.