

Sistemas operativos

Los sistemas operativos actuales están compuestos por un conjunto de software muy avanzado que trata de facilitar el empleo del dispositivo al usuario lo máximo posible e intentan malgastar lo menos posible los recursos hardware.

De este modo, los ordenadores incorporan una primera capa de software, el sistema operativo, que interacciona directamente con el hardware del equipo, encargándose de controlar los recursos y presentar al usuario una interfaz más fácil de entender, usar y sobre la que programar.

Sistema operativo: conjunto de programas que tiene como objetivo facilitar el uso del ordenador, encargándose de gestionar los recursos hardware y software del ordenador. Actúa como interfaz entre los programas de aplicación del usuario y el hardware.

Los PCs pueden ejecutar distintos tipos de sistemas operativos.

El sistema operativo es la primera capa de software. Es el primer programa que se carga en el ordenador y es responsable de la forma en que se utilice este. El mismo equipo hardware trabajara de una forma u otra dependiendo del tipo de sistema que se instale en él. El sistema operativo se comunica con el usuario o persona que utiliza el ordenador mediante el interfaz. Se trata de un segmento de una gran evolución, con los fabricantes actualizando sus versiones de sistemas operativos para diferenciarse de su competencia. Además, deben también adaptarse a los cambios que se producen a nivel hardware.

Conceptos importantes que es necesario entender al tratar con sistemas operativos:

- Programa: conjunto de instrucciones ejecutables por el ordenador, formado por algoritmos y estructuras de datos que producen un resultado.
- Proceso: unidad de ejecución de un programa con asignación de recursos del sistema.
- Hilo: cada flujo o tarea independiente en la ejecución de un proceso. Comparte código, datos y recursos del sistema con el resto de hilos, manteniendo su propio contador de programa y su pila (datos privados).
- Modo usuario y modo kernel: la diferencia entre el modo de usuario y el modo kernel es que el modo de usuario es el modo restringido en el que se ejecutan las aplicaciones y el modo de kernel es el modo privilegiado al que entra la computadora cuando accede a los recursos de hardware. La computadora está cambiando entre estos dos modos.

La administración del sistema por parte del sistema operativo se divide en:

Gestión de procesos

El procesador, como recurso fundamental del sistema, ha de repartir su tiempo entre los diferentes procesos que deseen ejecutarse. El sistema operativo debe organizar el paso de estos procesos por el procesador (o procesadores) y sus núcleos, de tal manera que los tiempos de ejecución de las diferentes tareas sigan los objetivos del sistema operativo. Por tanto, el sistema operativo debe gestionar:

- La asignación de procesos a varios procesadores (si dispone de varios).
- El uso de la multiprogramación sobre procesadores individuales y sus núcleos.
- La ejecución de una aplicación o proceso en cuanto a su sincronización con otros procesos o hilos.

Gestión de la memoria

Íntimamente ligado a la gestión de procesos se encuentra la de memoria. Por gestión de memoria se entiende la planificación y gestión global de la memoria principal con extensión a la memoria secundaria. Hoy en día los sistemas disponen de memoria RAM suficiente para albergar el sistema operativo y mucho más software. Pero también se debe planificar cómo actuar en caso de necesitar mayor espacio en memoria empleando el disco. El sistema operativo amplía virtualmente la memoria RAM, tomando prestado del disco duro espacio como si fuese una extensión de la primera (a este concepto se le llama memoria virtual).

Gestión de entradas y salidas.

Acciones como tocar una pantalla táctil, imprimir un documento, acceder a un fichero del disco duro o navegar por Internet requieren que el sistema operativo necesite administrar dichos recursos, ofreciendo soluciones rápidas y de la forma menos costosa posible.

Gestión de almacenamiento secundario. Los discos duros son dispositivos de E/S por sí mismos, pero la gestión de los archivos y directorios en su interior es fundamental. La estructura organizativa de los archivos y su gestión viene determinada por los sistemas de archivos.

Gestión de la seguridad

Se deben evitar actuaciones originadas por errores software, errores hardware o por actuaciones maliciosas de usuarios, ya sean intencionadas o no dando lugar a inconsistencias en el sistema. Por ello el sistema debe garantizar:

- El servicio y la disponibilidad de sus recursos.
- La confidencialidad, protección e integridad del sistema y los datos.
- El control de accesos.
- La autenticidad en las acciones.

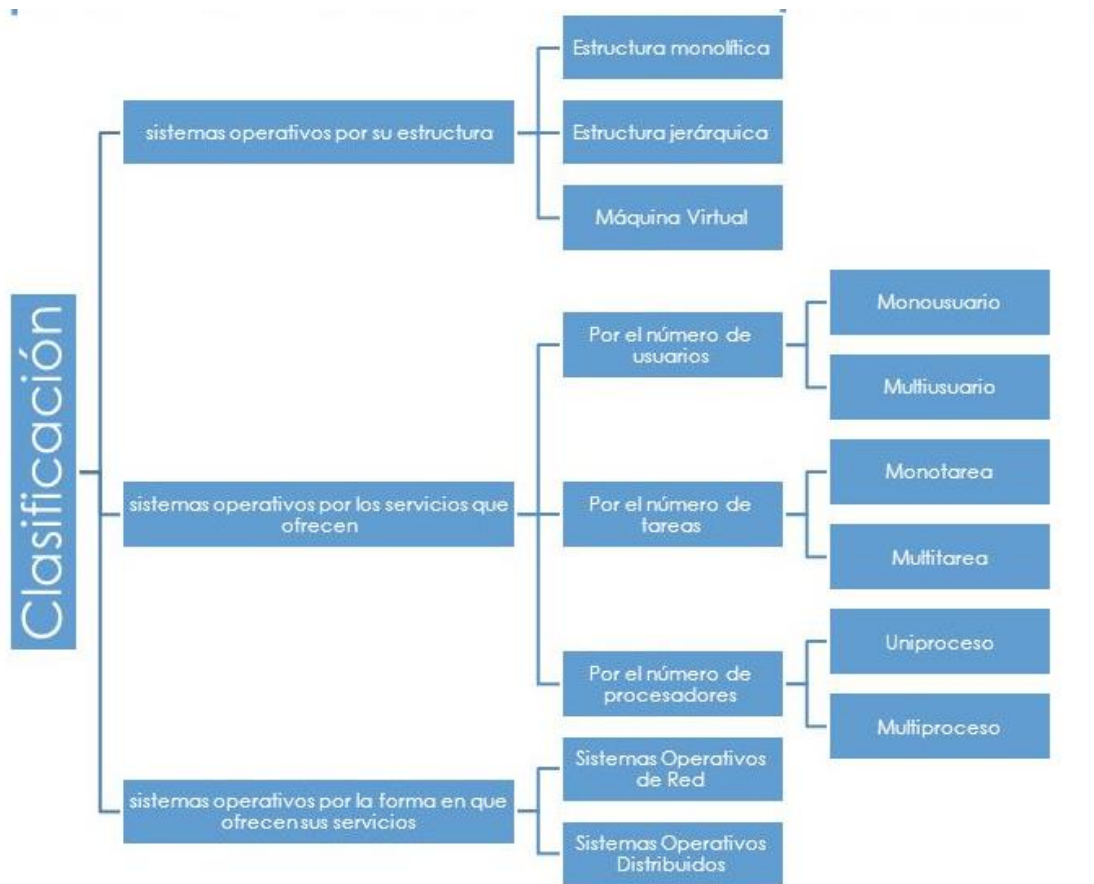
Gestión de errores

Es un elemento fundamental en todo sistema operativo. El control de las acciones que puedan derivarse del software de terceros, el hardware y el sistema operativo es prácticamente imposible. Por ello, el sistema operativo debe gestionar todo tipo de errores de la manera más liviana posible, informando al usuario y salvaguardando de forma prioritaria la seguridad del sistema y los datos.

Gestión de la interfaz de usuario

Todas las acciones encomendadas al sistema operativo tratadas hasta ahora no tendrían sentido sin una interfaz que permita una clara manejabilidad del sistema. Por tanto, los sistemas operativos, ya sean con interfaz gráfica o textual deben ofrecer un soporte que permita una fluida comunicación, así como realizar todas las acciones necesarias para la gestión, administración o explotación del mismo.

Tipos de sistemas operativos



Por el número de usuarios

En función de los usuarios que pueden trabajar simultáneamente:

- **Monousuario:** soportan un solo usuario a la vez. Ejemplo: MS-DOS
- **Multiusuario:** capaces de dar servicio a más de un usuario a la vez, ya sea mediante varios terminales conectados al ordenador (típico de Mainframe) o mediante sesiones remotas en una red. Ejemplos: Windows Server, Unix, Linux. Es independiente del número de procesadores y del número de tareas que cada usuario pueda ejecutar a la vez.

Por el número de tareas

Por el número de tareas simultáneas que pueden ejecutar:

- **Monotarea:** El usuario solo puede ejecutar un proceso en cada momento. Todos los recursos son asignados a un programa hasta completar la ejecución. Ejemplo: MS-DOS
- **Multitarea:** cada usuario puede ejecutar más de un programa concurrentemente, ya que la CPU comparte el tiempo de ejecución entre las diferentes tareas. Ejemplo: Windows, Linux, Unix, MAC OS X

Por el número de procesadores

Según el número de procesadores que puede gestionar:

- **Monoprocesador:** solo permiten un procesador.

- Multiprocesador: pueden trabajar con varios procesadores. Hay dos maneras de gestionarlos:
 - Simétricamente: los procesos son enviados indistintamente a cualquier procesador.
 - Asimétricamente: existe un procesador maestro que distribuye la carga entre ellos.

Por la forma de ofrecer los servicios

- Sistemas centralizados: gran ordenador (mainframe) que realiza todo el procesamiento y los usuarios se conectan a él mediante terminales tontos (sin capacidad de procesamiento) que no disponen de memoria, ni de procesador.
- Sistemas operativos en red: tienen la capacidad de interactuar con los sistemas operativos de otras máquinas de la red para intercambiar recursos. El sistema operativo cliente debe conocer la ubicación de los recursos para acceder a ellos.
- Sistemas operativos distribuidos: abarcan los mismos servicios que los sistemas operativos en red, pero los integran bajo una única máquina virtual a la que el usuario accede de forma transparente sin necesidad de conocer la máquina propietaria del recurso.

Los **sistemas operativos en red** guardan una gran relación con los llamados **sistemas operativos distribuidos**. En ambos casos, hay varias máquinas, varios usuarios y una red que permite su interconexión.

Sin embargo, hay una diferencia fundamental entre estos dos tipos de sistemas. En un sistema en red, **el usuario inicia sesión en la máquina de la red que desea**, que ejecuta su propio sistema operativo.

En cambio, en un sistema distribuido **el usuario no tiene por qué saber a qué ordenador se conecta**. Es el propio sistema el que decide cómo se interconectan y trabajan las distintas máquinas involucradas. Esto es muy importante en cuanto al código: un sistema distribuido es mucho más complejo de diseñar.

En función de requisitos temporales

- De tiempo real: requieren unos plazos muy estrictos para la ejecución. Deben ser muy precisos en el tiempo de respuesta. Por ejemplo, un sistema que controle la temperatura de un equipo y dispare un evento si se supera un umbral para evitar su sobrecalentamiento, no debe demorarse, ya que de lo contrario el equipo se debe averiar.
- Interactivos: para la interacción con el usuario. Su objetivo principal es responder lo más ágilmente posible al usuario, minimizando el tiempo de respuesta.
- Por lotes: El tiempo de respuesta al usuario no es importante. Están pensados para sacar el mayor partido de la potencia de la CPU. El operador carga los trabajos que se ejecutan en serie. El usuario no espera online por el resultado, sino que vuelve más tarde a recoger dichos resultados.

Arquitectura de sistemas operativos

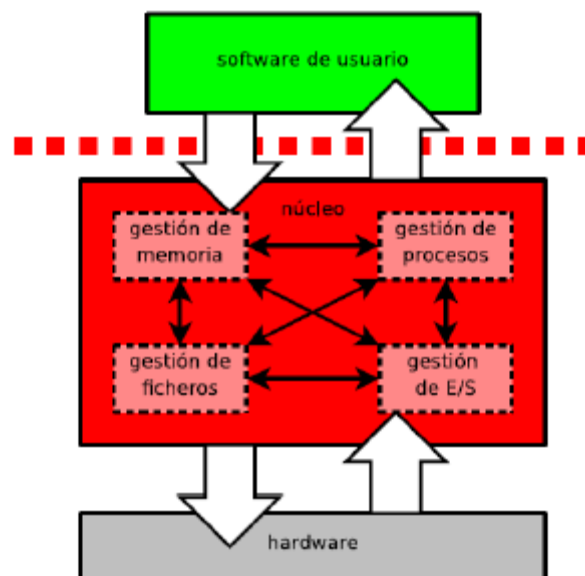
Según la estructura interna del sistema se pueden distinguir diferentes modelos de sistemas operativos. Los cambios en la arquitectura de los sistemas operativos han estado también muy relacionados con la evolución del hardware.

Sistemas monolíticos

En este diseño todo el sistema operativo se ejecuta como un solo programa en modo kernel, compuesto por múltiples rutinas entrelazadas que se pueden llamar unas a otras.

- Ventajas: Son rápidos y eficaces
- Inconvenientes: poco flexibles y modulares. Cualquier cambio en uno de los servicios requiere la recompilación del módulo y el reinicio del sistema.

El ejemplo más típico es Linux. También Unix en sus diferentes variantes: FreeBSD, HP-UX, Solaris, etc.



Sistemas de capas (o jerárquico)

A medida que aumentan las necesidades de los clientes y se fueron volviendo más complejos los sistemas, se hizo necesaria una mayor organización del software de sistema operativo. Aparecen los sistemas de capas.

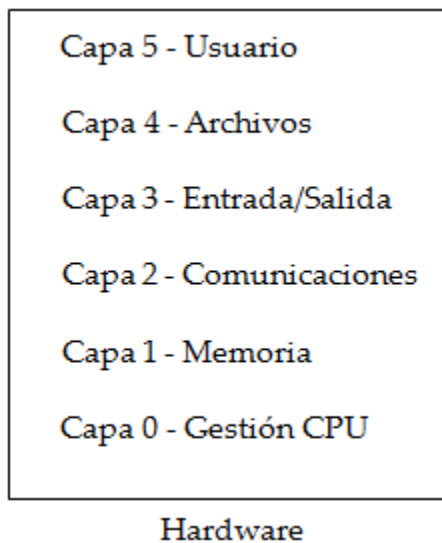
Se trata de una estructura por niveles, en los que cada capa realiza una función lógica definida. Ofrece servicios a la capa superior y utiliza los de la capa inferior.

La capa inferior interactúa con el núcleo.

- Ventajas: Mas modulares y escalables. Cuanto más interna es la capa, más privilegios tiene.

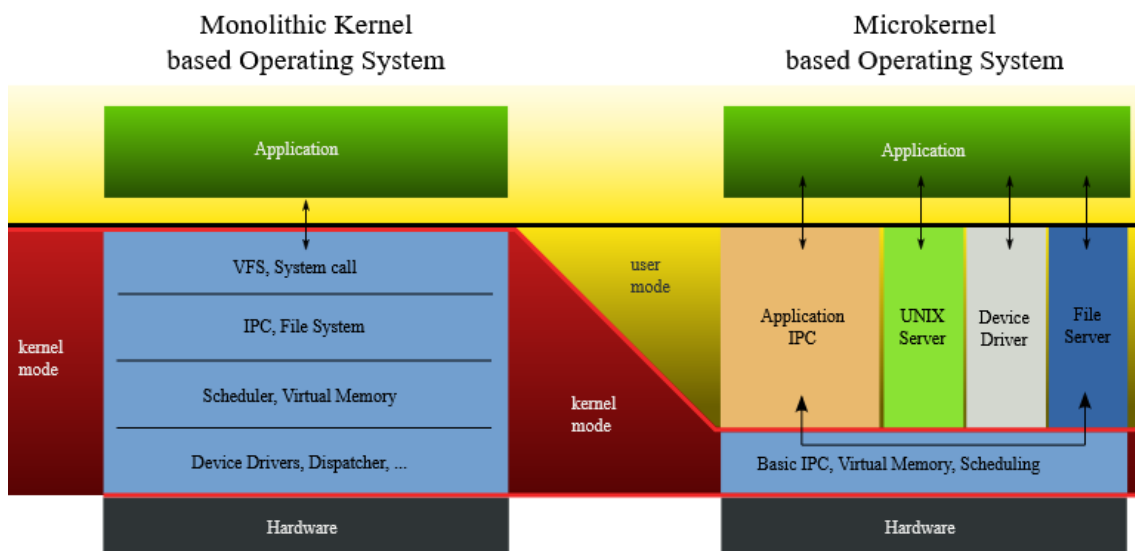
- Inconvenientes: son sistemas complejos

Ejemplos de SSOO de este tipo son THE y MULTICS.



Microkernels

La idea básica de este tipo de sistemas es lograr una alta confiabilidad al dividir el sistema operativo en módulos pequeños y bien definidos, donde solo uno de ellos (el microkernel), se ejecuta en modo kernel con privilegios elevados. El resto de módulos se ejecutan como procesos de usuario ordinarios, de modo que un error en ellos no se traslada al resto del sistema.



Ejemplos: AIX, Symbian, Windows (anterior a NT), Minix.

Ventajas:

- Mayor seguridad, la mayor parte de los procesos se ejecutan en modo usuario.

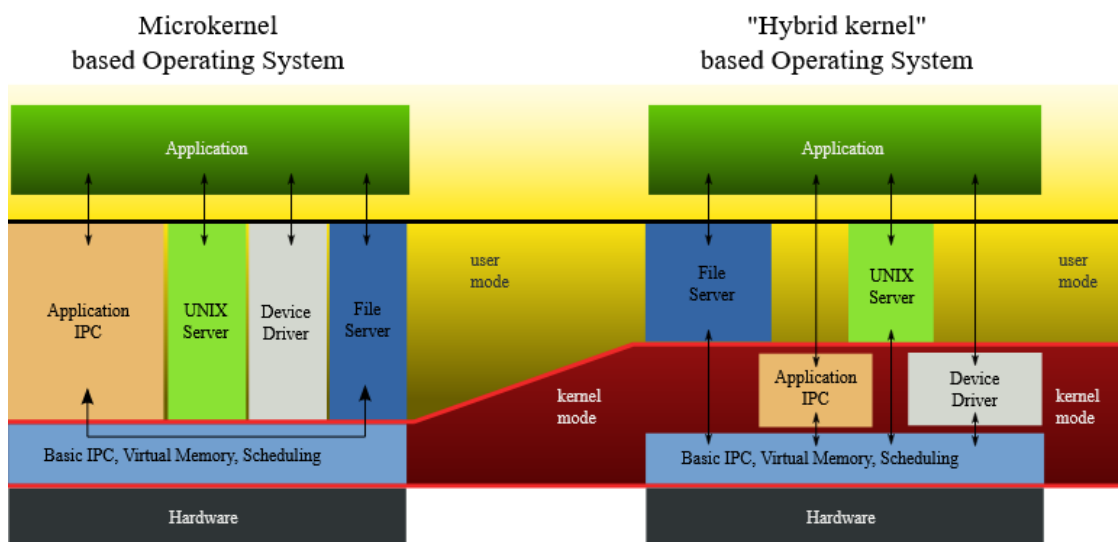
- Mayor estabilidad

Inconveniente principal: puede producirse sobrecarga en la gestión de los procesos y reducirse el rendimiento.

Kernel híbrido (Macrokernel)

Evolución del monolítico y microkernel que aúna lo mejor de ambas arquitecturas, consiguiendo sus ventajas: mayor estabilidad y rendimiento. Es similar al microkernel, pero ejecutan en modo kernel también los drivers de los dispositivos y las funciones de comunicación entre procesos. Es el modelo más usado actualmente.

Ejemplos: Mac OS X, Windows con núcleo NT y posterior.



Máquinas virtuales

En este modelo se crean máquinas virtuales independientes, cada una usada por uno o varios usuarios.

Todas se ejecutan sobre el mismo hardware. Son máquinas réplica de la máquina real, de modo que en cada una de ellas se puede ejecutar un sistema operativo diferente.

El núcleo del sistema operativo se llama monitor de máquina o hipervisor, existiendo dos tipos:

- Tipo 1 o nativo (también bare metal): se ejecuta directamente sobre el hardware. Ejemplos: VMWare ESXi, Microsoft Hyper-V, Xen, KVM
- Tipo 2 o hosted: sobre el hardware se instala un SO anfitrión, y las máquinas virtuales se ejecutan como una aplicación más en este sistema operativo.

Ejemplos: Microsoft Virtual PC, VMWare Workstation, Oracle VirtualBox.

Un hipervisor, conocido también como monitor de máquina virtual (VMM), es un software que crea y ejecuta máquinas virtuales (VM) y que, además, aísla su sistema operativo y recursos de las máquinas virtuales y permite crearlas y gestionarlas.

Cuando el sistema de hardware físico se usa como hipervisor, se denomina "host", y las múltiples máquinas virtuales que utilizan sus recursos se denominan "guests".

El hipervisor utiliza los recursos, como la CPU, la memoria y el almacenamiento, como un conjunto de medios que pueden redistribuirse fácilmente entre los guests actuales o en las máquinas virtuales nuevas.

Ejemplos de sistemas operativos

Entornos de escritorio:

- Windows: Windows 10 (en menor medida versiones anteriores) y Windows 11.
- Linux: Ubuntu (ya en version 21.04), Fedora. Como curiosidad, existe una distribucion de Linux realizada en Galicia: Trisquel.
- Mac OS X (ultima version estable 10.15). Ya lanzada la 11.6

Servidores:

- Windows Server: actualmente en 2022, pero existe todavia gran implantación de 2019, 2016 y 2012 (e incluso anteriores).
- Linux: RedHat, CentOS, Ubuntu, ...
- Unix: HP-UX, AIX, Solaris. Muchos han sido discontinuados, y con el desarrollo y mejoras en el soporte de Linux el uso de Unix ha quedado minimizado.
- Mainframe: z/OS (anteriormente OS/390)

Dispositivos móviles:

- Android: en version 12.
- iOS: ultima version la 15.1
- Existen versiones de Windows, Firefox o Blackberry, pero ya de uso muy residual o discontinuadas.

Gestión de procesos

Como ya se definió anteriormente, conceptos importantes de proceso:

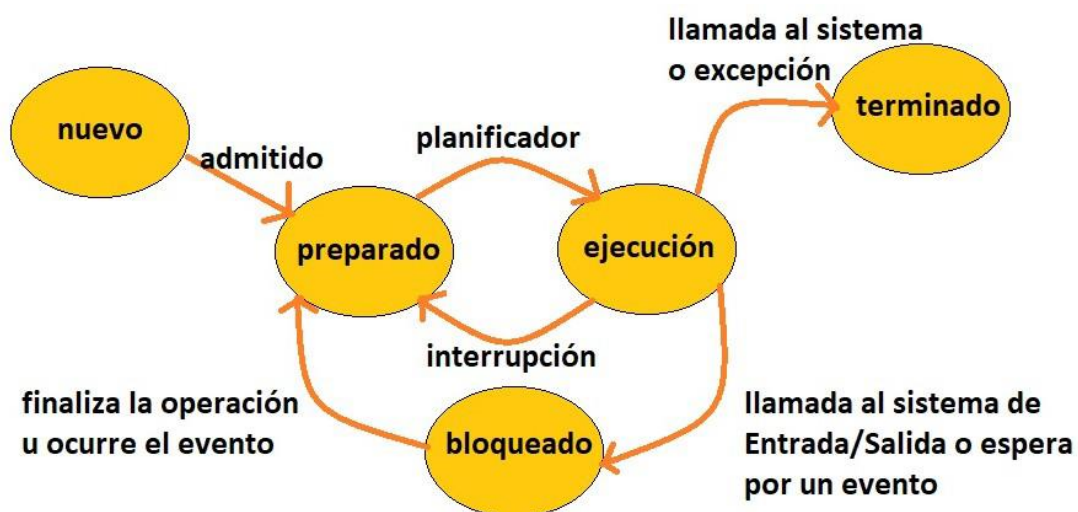
- Programa: conjunto de instrucciones ejecutables por el ordenador, formado por algoritmos y estructuras de datos que producen un resultado.
- Proceso: unidad de ejecución de un programa con asignación de recursos del sistema. Varios procesos pueden ejecutar el mismo programa.
- Hilo: cada flujo independiente en la ejecución de un proceso. Comparte código, datos y recursos del sistema con el resto de hilos, manteniendo su propio contador de programa y su pila (datos privados).

Un sistema operativo que trabaja con hilos se denomina multihilo. En caso contrario son monohilo. La mayoría de los sistemas operativos actuales son multihilo.

En un sistema multiprogramado existen varios procesos en ejecución simultáneamente. Pero no hay un procesador para cada proceso, con lo que el sistema operativo debe repartir el tiempo entre los procesos y dar la sensación al usuario de que todos se ejecutan a la vez.

Los recursos del ordenador son finitos y el sistema operativo debe controlar que el acceso se realiza de forma correcta. Por ejemplo, debe asegurarse que solo un proceso escribe en un recurso de acceso exclusivo (ejemplo, un fichero).

Ciclo de vida genérico de un proceso:



- Nuevo: proceso nuevo creado.

- Listo o preparado: proceso está preparado para ejecutarse y a la espera de que la CPU le asigne un slice de tiempo.
- En ejecución: está actualmente siendo ejecutado por la CPU
- Bloqueado: el proceso se encuentra a la espera de que finalice una operación de E/S.
- Terminado: el proceso ha finalizado.

Tareas importantes del sistema operativo en la gestión de procesos:

- **Comunicación:** los procesos pueden necesitar comunicarse para intercambiar información. Se puede hacer mediante manejo de memoria compartida o mediante el intercambio de mensajes.
- **Sincronización entre procesos:** el sistema operativo debe vigilar que el acceso a los recursos se realiza de forma correcta, para evitar que el sistema se bloquee. Debe proporcionar mecanismos para que los procesos accedan de forma ordenada a los recursos.
- **Planificación de procesos:** el sistema debe asignar la CPU a los procesos que están en estado listo. Dependiendo del tipo de sistema el tipo de planificación utilizado varía. Se describe a continuación con algo más de detalle.

La planificación de los procesos

Los procesos se ejecutan según un orden establecido por el sistema operativo. La forma en que el sistema operativo gestiona los procesos es lo que se conoce como planificación y la herramienta que lo hace recibe el nombre de planificador (scheduler).

OJO: se planifican los hilos, no los procesos. Por simplicidad, hablaremos de planificación del proceso.

Los objetivos del planificador del sistema son:

- Justicia: asegurarse que todos los procesos tengan su turno en la CPU. El reparto dependerá del tipo de sistema, ya que la asignación no debe ser la misma en un sistema de tiempo real que en un sistema interactivo.
- Tiempo de respuesta: mantener el tiempo de respuesta por debajo de un umbral determinado.
- Productividad: maximizar el número de procesos finalizados por unidad de tiempo.
- Evitar aplazamiento indefinido. Los procesos deben finalizar en un tiempo finito.

El sistema debe ser predecible. Si se ejecuta el proceso bajo cargas similares en el sistema, la respuesta debe ser siempre similar.

Existen tres niveles de planificación:

- Nivel alto o a largo plazo: determina que procesos serán admitidos.
- Nivel medio o a medio plazo: Decide qué procesos se suspenden o reanudan y guarda los procesos suspendidos en disco para liberar memoria principal.
- Nivel bajo o corto plazo: Es el que asigna uno de los procesos en estado listo a la CPU.

Para realizar la planificación de los procesos, el sistema operativo utiliza algoritmos de planificación, que pueden ser:

- No apropiativos: una vez que el SO le otorga la CPU a un proceso ya no se la puede quitar, hasta que el proceso finaliza o se queda bloqueado a la espera de una operación de entrada/Salida. Son más sencillos de implementar, y utilizados en sistemas batch y monousuario. Ejemplos: FCFS (First Come First Served) o SJF (Shortest Job First)
- Apropiativos: el SO puede sacar de la CPU al proceso en cualquier momento. Son más complejos de implementar, pero más óptimos en el uso de CPU. Se usan en sistemas multiusuario y multitarea. Ejemplos: Round Robin, prioridades.

En la actualidad los sistemas Windows o Linux utilizan colas multinivel. Disponen de varias colas, cada una con una prioridad. En cada cola utilizan un algoritmo Round-Robin (Ojo: es algo más complejo).

- Windows asigna una prioridad que va del 0 al 31 y puede ser modificada (0 a

15 son de procesos de usuario y 16 a 31 de kernel). La prioridad puede ser modificada.

- Linux asigna una prioridad dinámica, que depende de la prioridad del proceso y del tiempo que lleva en ejecución. El sistema la va reajustando.