

CONTENIDO

<i>Introducción a los Sistemas Operativos.....</i>	<i>2</i>
Concepto	2
Objetivos.....	2
<i>Funciones de los Sistemas Operativos.....</i>	<i>4</i>
Los primeros ordenadores.....	5
Secuencia automática de trabajos.....	7
<i>Tipos de Sistemas Operativos.</i>	<i>8</i>
Sistemas Operativos por su estructura (visión interna).	8
Tipos de Sistemas Operativos según su visión interna.	10
Sistemas Operativos según su visión externa.	13
Sistemas Operativos de escritorio.....	13
Sistemas Operativos en Red y servidores.....	13
Sistemas Operativos distribuidos.	13
Sistemas Operativos por su disponibilidad.	16
Sistemas operativos propietarios.	16
Sistemas operativos libres.	16
Sistemas Operativos por su tipo de licencia.	17
O.E.M.....	17
Retail	17
VLM (Licencias por volumen).....	17
MSDN (Licencias de educación.).....	18

INTRODUCCIÓN A LOS SISTEMAS OPERATIVOS.

El ordenador es un sistema programable formado por un conjunto de elementos hardware que necesitan instrucciones que le indiquen cómo utilizar los recursos. El conjunto de instrucciones o programas es lo que conocemos como soporte lógico o software. Un ordenador, sin software que lo programe, es básicamente un bloque de metal inútil, pero con el software puede almacenar, procesar y obtener información, editar textos, controlar el entorno, etc.

CONCEPTO

Sin duda alguna, la utilización de los recursos mediante programas es muy complicada, puesto que cada dispositivo es diferente y con gran cantidad de características a controlar. Por ello, una de las primeras acciones a llevar a cabo es el diseño y codificación del software que nos facilite el manejo de estos recursos, evitando, en lo posible, que debamos poseer profundos conocimientos del hardware, cediéndole esta tarea a un reducido número de profesionales que serán los que construyan dicho software. Una vez realizado este esfuerzo de diseño, cabe pensar por que no se completa un poco más con el fin de dotar a los usuarios de unas cuantas funciones adicionales, que no sólo faciliten el uso de estos recursos, sino que además los potencien lo más posible. Pues bien, este software así diseñado, cuya finalidad es gestionar adecuadamente los recursos para que realicen el trabajo que se les ha encomendado, y que, además, potencien las funciones de los mismos, es lo que denominaremos sistema operativo, pudiéndolo definir como:

Un sistema operativo es un conjunto de programas que, ordenadamente relacionados entre sí, contribuyen a que el ordenador lleve a efecto correctamente el trabajo encomendado.

OBJETIVOS.

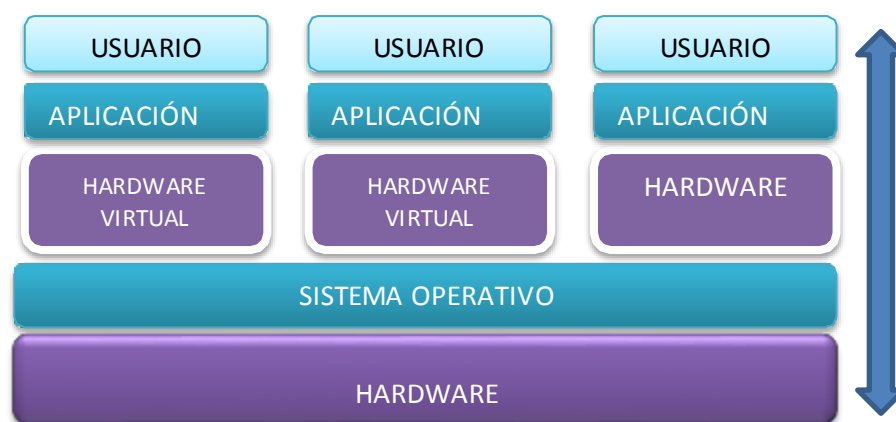
Desde el punto de vista del usuario, el sistema operativo consiste en una **serie de programas** y funciones que **ocultan** los detalles del **hardware**, ofreciéndole una vía sencilla y flexible de acceso al mismo, teniendo dos **objetivos** fundamentales:

- **Seguridad:** El sistema operativo debe actuar contra cualquier manipulación extraña, ya sea accidental o premeditada que pudiera dañar la información, perjudicar a otros usuarios o provocar un funcionamiento indeseado del sistema. Por ejemplo, hay ciertas instrucciones que pueden parar la máquina y otras que realizan operaciones directamente sobre el hardware, que debemos evitar que se utilicen por los programas.



Para ello, algunos sistemas proporcionan dos estados, llamados estado protegido (Sistema o **Kernel**), en el cual se ejecuta el **sistema operativo**, y estado no protegido (Usuario o **User**), que es el destinado a la ejecución de los **programas de usuario** y de aplicación. De esta manera se impide que los programas de los usuarios puedan tener contacto directo con el hardware, o puedan forzar un incorrecto funcionamiento del sistema.

- **Abstracción:** La tendencia actual del software y de los lenguajes de programación es ocultar lo más posible los detalles de más bajo nivel, intentando dar a los niveles superiores una visión más sencilla, global y abstracta, ofreciéndoles operaciones para manipular dichas estructuras ocultas, desconociendo por completo la gestión interna de las mismas. Sobre estas estructuras se construyen otras que abstraen a las anteriores, y así sucesivamente. Gracias a la **abstracción**, los sistemas operativos **enmascaran** los **recursos físicos**, permitiendo su manejo con funciones más generales que ocultan las básicas, constituyendo verdaderos recursos ficticios o virtuales, que mejoran y son más potentes que los físicos.



Desde el punto de vista de un programa o usuario, la máquina física se convierte, gracias al sistema operativo, en una **máquina virtual**, también conocida como máquina extendida, que presenta la ventaja respecto a la física de ofrecer más funciones de las que normalmente soportaría esta última. Desde el punto de vista del usuario, el sistema operativo proporciona servicios que no están presentes en la máquina subyacente. Estos servicios incluyen las facilidades de carga y ejecución de programas, interacción entre el usuario y los programas, permitiendo que se ejecuten varios al mismo tiempo, gestión de la contabilidad para facturar los servicios y almacenamiento de datos y programas.

Como resumen, podemos decir que el **sistema operativo** persigue alcanzar la **mayor eficiencia** posible del **hardware** y **facilitar** el **uso** del mismo a los **usuarios** y a las aplicaciones.

FUNCIONES DE LOS SISTEMAS OPERATIVOS.

Las funciones de los sistemas operativos son diversas y han ido evolucionando de acuerdo con los progresos que la técnica y la informática han experimentado. Como principales funciones, podríamos enumerar las siguientes:

- ▶ **Gestión de procesos.** Hay que diferenciar entre los conceptos programa y proceso. Un programa es un ente pasivo, que cuando se carga en memoria y comienza a ejecutarse, origina uno o varios procesos.
- ▶ **Gestión de la memoria.** La gestión de memoria, suele ir asociada a la gestión de procesos. Para ejecutar un proceso es necesario asignarle unas direcciones de memoria exclusivas para él y cargarlo en ellas, cuando el proceso finalice su ejecución es necesario liberar las direcciones de memoria que estaba usando.
- ▶ **Gestión de ficheros.** Un fichero es una **abstracción** para definir una **colección de información no volátil**. Su objetivo es proporcionar un modelo de trabajo sencillo con la información almacenada en los dispositivos de almacenamiento. Estos ficheros deben tener espacio asignado en los dispositivos, deben estar protegidos entre ellos, deben organizarse según unos determinados esquemas... todo esto es la gestión de ficheros.
- ▶ **Gestión de los dispositivos de E/S.** La gestión de la entrada salida (E/S) tiene como objetivo proporcionar una interfaz de alto nivel de los dispositivos de E/S sencilla de utilizar.
- ▶ **Gestión de la red.** El sistema operativo es el encargado de gestionar los distintos niveles de red, los drivers (controladores) de los dispositivos involucrados en la red, los protocolos de comunicación, las aplicaciones de red, etc.
- ▶ **Protección y seguridad.** Mecanismos para permitir o denegar el acceso a los usuarios y a sus procesos a determinados recursos (ficheros, dispositivos de E/S, red, etc.).

Para comprender mejor porqué existen dichas funciones y cuáles son sus objetivos, las iremos estudiando mientras hacemos un breve recorrido a través de la historia de los ordenadores y la informática, ya que nos ayudara a comprender mejor el concepto de sistema operativo.

Los objetivos fundamentales de los sistemas operativos respecto a conseguir la mayor **eficiencia y facilidad de uso** posibles, no son siempre compatibles, ya que cualquier sistema que deba ser eficiente, normalmente no será fácil de usar, mientras que, si es fácil de usar, se deberá ofrecer a los usuarios muchas facilidades y ayudas, incluyendo muchos pasos e información que para un usuario experto no serían necesarias, lo que implica, obviamente, una pérdida de eficiencia.

LOS PRIMEROS ORDENADORES.

Los primeros ordenadores tenían un gran tamaño, eran extremadamente caros y muy difíciles de usar. Estas enormes máquinas ocupaban normalmente amplias salas y eran gestionadas por el usuario desde una consola, único medio de acceder a dicho ordenador. Cada usuario tenía asignados períodos de tiempo durante los cuales sólo él podía utilizar el ordenador, siendo el dueño absoluto de la máquina.



Cuando a un usuario le llegaba su tiempo de máquina, tenía que apresurarse a introducir en el ordenador todas las fichas perforadas que conformaban su programa, ejecutar el programa en el ordenador, vigilar su funcionamiento y esperar a que todas las operaciones se terminaran (si había suerte, antes que se le terminara su tiempo de máquina).

Estos ordenadores se basaban en dos factores: sus dispositivos de entrada/ salida y su habilidad para ejecutar un programa, pero no disponían de recursos lógicos adicionales, como pudieran ser medios de almacenamiento secundario por lo que, los usuarios debían introducir sus programas en el ordenador cada vez que se deseaba ejecutar el trabajo correspondiente.

En el caso de que al programador se le acabara el tiempo de máquina concedido sin haber terminado el trabajo, éste debía suspenderlo en el estado en que se encontrara en ese instante, recopilar todo el material obtenido, retirarse a su mesa de trabajo, y estudiarlo hasta que tuviera otra vez la oportunidad de disponer del ordenador. Por el contrario, si el programador acababa antes del final del tiempo asignado, el ordenador quedaba inactivo hasta el siguiente período de tiempo concedido a otro programador.

Podemos deducir que el aprovechamiento de los recursos del ordenador era escasísimo, y por tanto carísimo, además de no ser satisfactorio para los usuarios; de ahí los esfuerzos para mejorar su rendimiento.

Prestando una mínima atención al procedimiento anterior, se pueden pensar varios mecanismos para obtener un mayor aprovechamiento del ordenador, principalmente incidiendo en los tiempos muertos del sistema que podrían utilizarse para llevar a cabo otros trabajos mientras se realizan las correcciones, estudios, etcétera, sobre un programa ejecutado que tuviese errores.

Para resolverlo, los propietarios de los sistemas contrataron a una o varias personas especializadas para ejecutar las rutinas de carga y descarga, con el fin de mantener el sistema con la máxima ocupación posible, recibiendo los trabajos de los usuarios para su ejecución. De esta forma, al recibir dichos trabajos, los reunía y ejecutaba secuencialmente consumiendo únicamente el tiempo que realmente necesitasen y evitando en gran medida los tiempos de inactividad del procesador.

Esta persona se conoce como el operador del ordenador, que es un técnico dedicado únicamente a su manipulación para realizar los trabajos encomendados. A partir de este momento, **el programador o usuario dejó de tener acceso directo al sistema.**

En el caso de que existiesen errores en la ejecución de los trabajos, se hacía un volcado en código binario de la memoria para su entrega al programador, y se ejecutaba inmediatamente el siguiente trabajo, disminuyendo considerablemente los tiempos muertos y, por lo tanto, mejorando el rendimiento.

Además, otra de las misiones del operador, era agrupar los trabajos que tuvieran necesidades de recursos físicos y lógicos similares para que se ejecutasen como un grupo. Supongamos que tenemos varios programas que deben ser traducidos con el compilador Fortran y otros con Cobol; si se reúnen todos los programas Fortran y se compilan como un grupo, sólo habrá que cargar el compilador una vez, obteniendo un considerable ahorro de tiempo.

A este modo de trabajo se le conoce como “proceso por lotes” o en inglés “batch”. Es una forma de trabajo en la que los programas no pueden ser interactivos, debido a que el usuario no está presente cuando el trabajo se ejecuta.



SECUENCIA AUTOMÁTICA DE TRABAJOS.

A pesar del ahorro de tiempo inactivo y de la agrupación de trabajos, aún persistían breves períodos de inactividad, ya que, si un trabajo se paraba por algún error, el operador debía observar la consola y tomar nota de todo lo sucedido para comunicárselo al programador. Analizando el trabajo del operador se observó que era bastante mecánico, y que se podía automatizar en gran parte, pensando que podía diseñarse un programa que estuviese permanentemente residente en la memoria del ordenador y que fuese el que realizase muchas de esas operaciones, surgiendo la secuencia automática de trabajos.

Se diseñó un pequeño programa que transfería automáticamente el control de un trabajo a otro. Este programa se denominó **Monitor Residente**, que se puede considerar como el germen de un Sistema Operativo. El monitor **residía permanentemente en memoria**. En el momento de encender el ordenador se daba control al monitor, el cual, a su vez, daba control al primer trabajo, de manera que cuando terminaba su ejecución, el monitor tomaba el control de nuevo activando el siguiente trabajo, y así sucesivamente. Es decir, controlaba la secuencia de los trabajos a realizar.

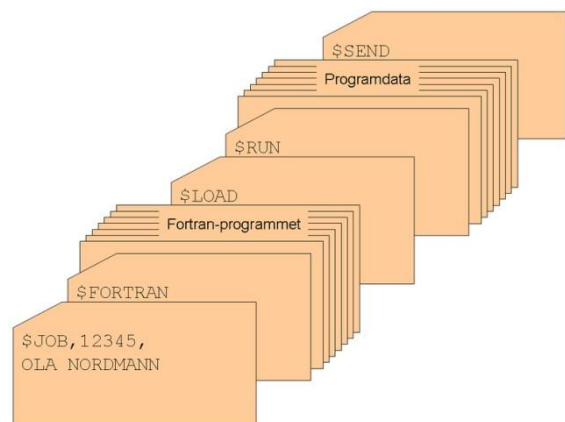
Para que el monitor supiera qué programa debía ejecutar, y qué datos iba a tratar, se añadieron al paquete de tarjetas que contenía el programa unas tarjetas de control con las directivas necesarias para dicho monitor. Estas tarjetas debían ceñirse a un lenguaje estricto de comandos denominado Lenguaje de Control de Trabajos (JCL: Job Control Language).

Para diferenciar estas tarjetas del resto del programa, con objeto de identificarlas y poder ser tratadas adecuadamente, la primera columna debía contener algún símbolo especial que variaba de un sistema a otro, pero que normalmente era "\$" o "//", tal y como se ve en la figura.

Además, debía poder gestionar la entrada y salida de datos desde el exterior al ordenador y viceversa, con el fin de poder llevar a cabo correcta y rápidamente el trabajo encomendado, sin necesidad de intervención de los usuarios u operadores.

A partir de este momento, los operadores tenían la misión de cargar y descargar las tarjetas en los lectores y perforadores correspondientes, instalar y extraer las cintas magnéticas, mantener las impresoras con suficiente papel, y todas aquellas labores que son necesarias para que el sistema esté operativo físicamente.

Vemos aquí como por primera vez en el ordenador se introducen y procesan **programas** que no tienen una utilidad "directa", es decir, que no buscan obtener un resultado en concreto pedido por un programador, sino que **facilitan el uso** de la máquina.



TIPOS DE SISTEMAS OPERATIVOS.

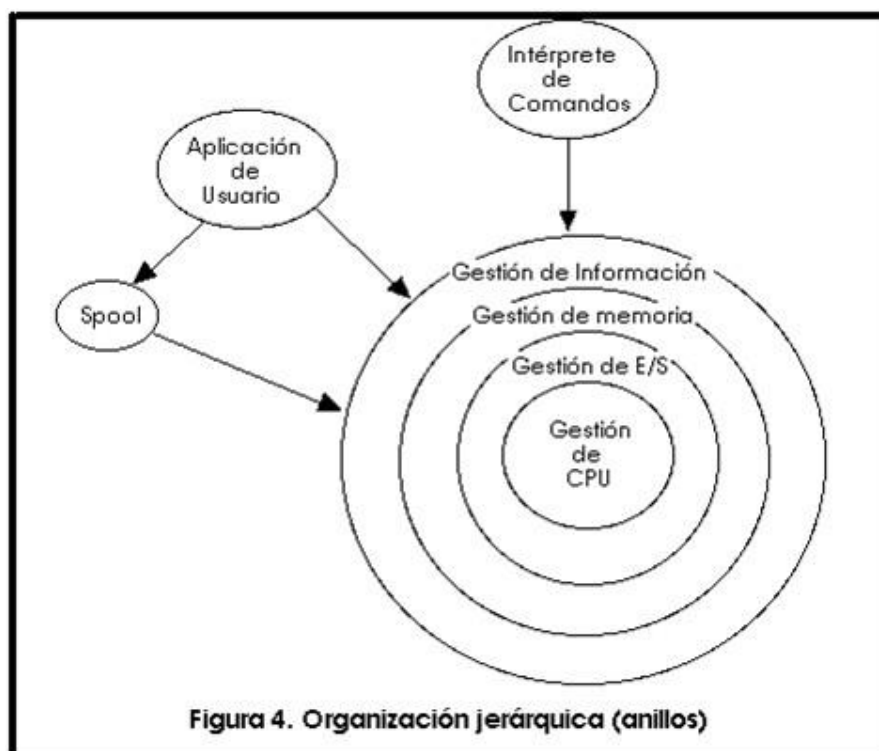
En este punto vamos a describir algunas características que permiten clasificar a los sistemas operativos. Básicamente veremos tres clasificaciones: sistemas operativos por su estructura (visión interna), sistemas operativos por los servicios que ofrecen y sistemas operativos por la forma en que ofrecen sus servicios (visión externa).

SISTEMAS OPERATIVOS POR SU ESTRUCTURA (VISIÓN INTERNA).

Si estudiamos los sistemas operativos atendiendo a su estructura interna, veremos que existen dos tipos fundamentales, los sistemas de estructura **monolítica** y los sistemas de estructura **jerárquica**.

En los sistemas operativos de estructura **monolítica** nos encontramos con que el sistema operativo está formado por **un único programa** dividido en rutinas, en donde cualquier parte del sistema operativo tiene los mismos privilegios que cualquier otra. Estos sistemas tienen la ventaja de ser muy rápidos en su ejecución (solo hay que ejecutar un programa) pero cuentan con el inconveniente de carecer de la flexibilidad suficiente para soportar diferentes ambientes de trabajo o tipos de aplicaciones. Es por esto que estos sistemas operativos suelen ser hechos a medida, para solucionar un problema en concreto y no para trabajar de forma generalista.

A medida que fueron creciendo las necesidades de los usuarios y se perfeccionaron los sistemas, se hizo necesaria una mayor organización del software, del sistema operativo, donde una parte del sistema contenía partes más pequeñas y esto organizado en forma de niveles. Se dividió el



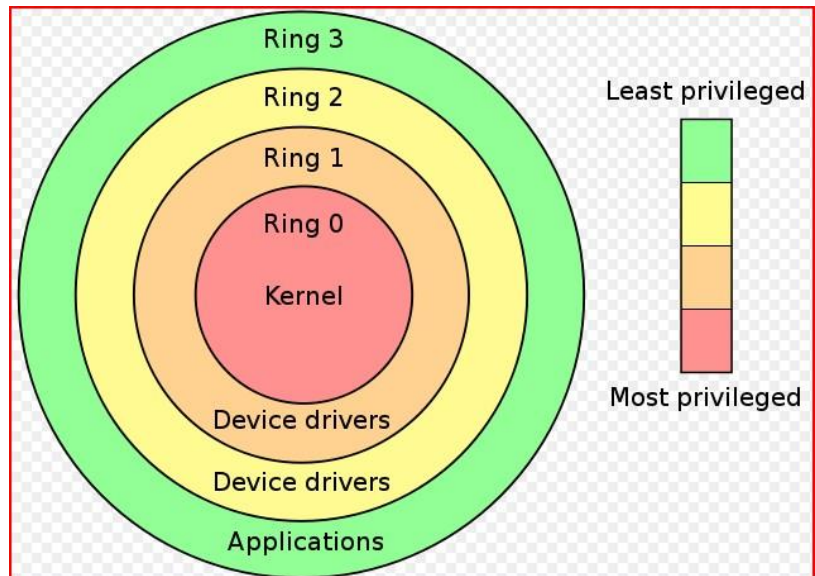
sistema operativo en pequeñas partes, de tal forma que cada una de ellas estuviera perfectamente definida y con un claro interface con el resto de elementos.

Se puede pensar también en estos sistemas como si fueran 'multicapa'.

En la estructura anterior se basan prácticamente la mayoría de los sistemas operativos actuales.

Otra forma de ver este tipo de sistema es la denominada de anillos concéntricos o "rings"

En el sistema de anillos, cada uno tiene una apertura, conocida como puerta o trampa (trap), por donde pueden entrar las llamadas de las capas inferiores. De esta forma, las zonas más internas del sistema operativo o núcleo del sistema estarán más protegidas de accesos indeseados desde las capas más externas. Las capas más internas serán, por tanto, más privilegiadas que las externas.



Cada capa supervisa a la capa que tiene por encima, de modo que para que algo se ejecute en la capa 5, por ejemplo, debe recibir permiso y supervisión de la capa 4, que esta supervisada por la 3, y así sucesivamente. Evidentemente cuanto más al "exterior" de la estructura se ejecute un programa, más lento va a ser su funcionamiento ya que va a recibir un gran número de supervisiones. Por el contrario, cuanto más en el interior se ejecute un proceso, mayor será su velocidad.

En el centro de esta estructura se encuentra el Kernel o Núcleo del sistema operativo, que es su parte más importante.

TIPOS DE SISTEMAS OPERATIVOS SEGÚN SU VISIÓN EXTERNA.

Según el número de usuarios que soporta concurrentemente:

- ▶ **Monousuarios.** Los sistemas operativos monousuarios son aquéllos que soportan a un usuario a la vez, sin importar el número de procesadores que tenga la computadora o el número de procesos o tareas que el usuario pueda ejecutar en un mismo instante de tiempo. Las computadoras personales típicamente se han clasificado en esta sección.
- ▶ **Multiusuario.** Los sistemas operativos multiusuario son capaces de dar servicio a más de un usuario a la vez, ya sea por medio de varias terminales conectadas a la computadora o por medio de sesiones remotas en una red de comunicaciones. No importa el número de procesadores en la máquina ni el número de procesos que cada usuario puede ejecutar simultáneamente.

Según el número de tareas que puede ejecutar concurrentemente:

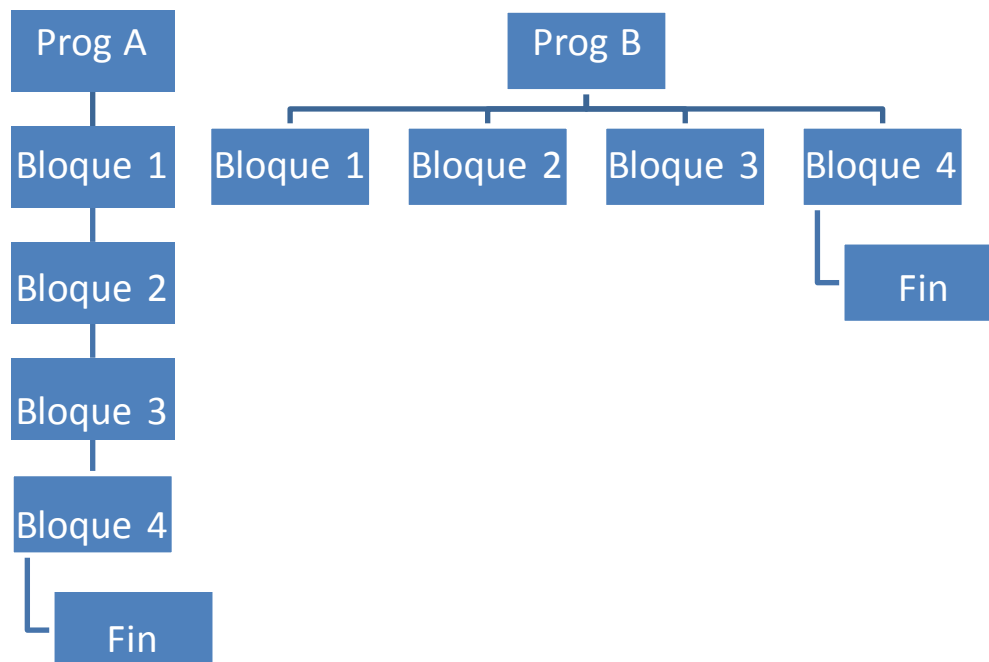
- ▶ **Monotareas.** Los sistemas monotarea son aquellos que sólo permiten una tarea a la vez por usuario. Puede darse el caso de un sistema multiusuario y monotarea, en el cual se admiten varios usuarios al mismo tiempo, pero cada uno de ellos puede estar haciendo solo una tarea a la vez.
- ▶ **Multitareas.** Un sistema operativo multitarea es aquél que le permite al usuario estar realizando varias labores al mismo tiempo. Por ejemplo, puede estar editando el código fuente de un programa durante su depuración mientras compila otro programa, a la vez que está recibiendo correo electrónico en un proceso en background (segundo plano). Es común encontrar en ellos interfaces gráficas orientadas al uso de menús y el ratón, lo cual permite un rápido intercambio entre las tareas para el usuario, mejorando su productividad.

Según el número de procesadores que puede gestionar:

- ▶ **Uniproceto.** Un sistema operativo uniproceto es aquél que es capaz de manejar solamente un procesador de la computadora, de manera que si la computadora tuviese más de uno le sería inútil. Por ejemplo, Windows 98 es un sistema operativo Uniproceto.
- ▶ **Multiproceto.** Un sistema operativo multiproceto es capaz de manejar más de un procesador en el sistema, distribuyendo la carga de trabajo entre todos los procesadores que existan en el sistema. Generalmente estos sistemas trabajan de dos formas: simétricamente o asimétricamente.
 - Cuando se trabaja de manera **asimétrica**, el sistema operativo selecciona a uno de los procesadores el cual jugará el papel de procesador maestro y servirá como pivote para distribuir la carga a los demás procesadores, que reciben el nombre de esclavos. El único procesador que realmente tiene acceso a todos los recursos del sistema es el maestro, que relega en los esclavos los trabajos que le van llegando. Es un sistema simple de construir y donde es muy fácil añadir más procesadores esclavos.

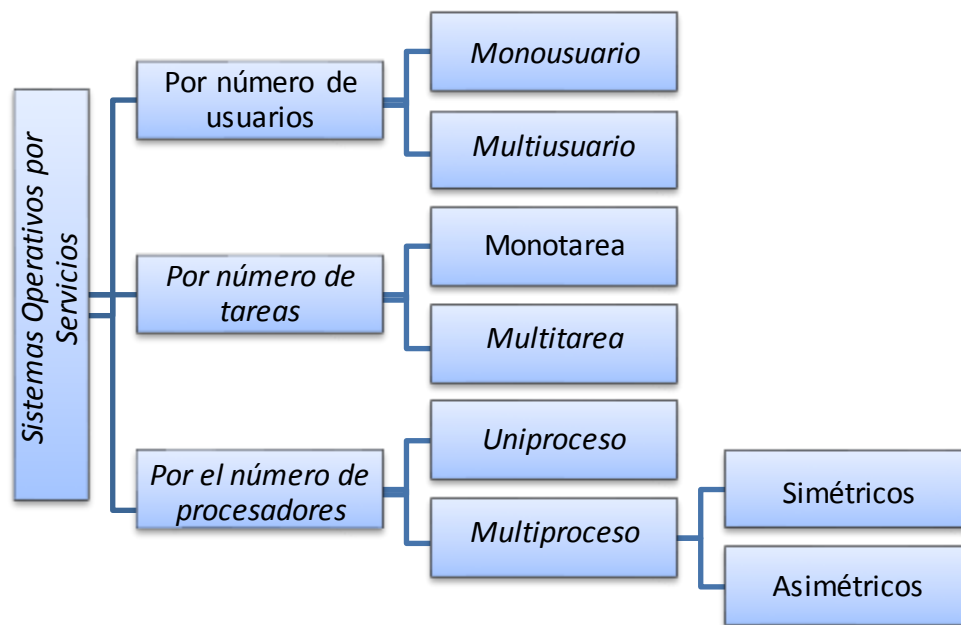
- Cuando se trabaja de manera **simétrica**, los procesos o partes de ellos (threads, hebras o hilos) son enviados indistintamente a cualquiera de los procesadores disponibles, teniendo una mejor distribución y equilibrio en la carga de trabajo bajo este esquema. Se dice que un thread es la parte activa en memoria y corriendo de un proceso, lo cual puede consistir de un área de memoria, un conjunto de registros con valores específicos, la pila y otros valores de contexto. Es un sistema mucho más difícil de construir, y es también tremendamente complicado añadir más procesadores, pero tiene la gran ventaja de ser muchísimo más práctico, ya que cada procesador tiene acceso a todos los recursos y las cargas de trabajo se pueden dividir de forma mucho más rápida.

Un aspecto importante a considerar en estos sistemas es que la aplicación debe construirse específicamente para aprovechar varios procesadores. Existen aplicaciones que fueron hechas para correr en sistemas uniproseso que no aprovechan el multiproseso, ya que el código debe contener secciones de código paralelizable (que se puedan correr en paralelo), los cuales son ejecutados al mismo tiempo en procesadores diferentes.



Así, Prog A se ejecutará en 4 ciclos de reloj sin importar si se ejecuta en un sistema monoprocesador o no, mientras que Prog B se ejecutará en 4 ciclos de reloj en un sistema monoprocesador, pero en 1 ciclo de reloj en un sistema con 4 procesadores.

Resumen de los tipos de sistemas operativos según su visión externa.



SISTEMAS OPERATIVOS SEGÚN SU FUNCIÓN.

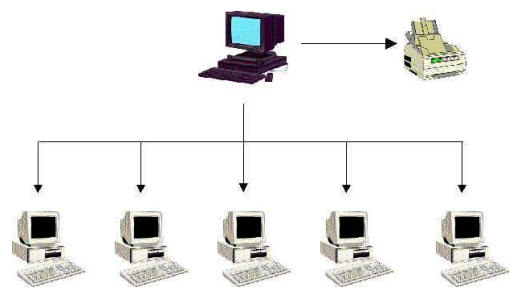
Esta clasificación se refiere a la función que el sistema va a realizar.

SISTEMAS OPERATIVOS DE ESCRITORIO.

Estos sistemas operativos se utilizan en los equipos personales, estaciones de trabajo, portátiles, etc. También se suelen conocer como sistemas operativos clientes. Windows 7 por ejemplo, es un sistema operativo de escritorio. Suelen ser sistemas operativos preparados para permitir un uso fácil por parte del usuario, destacan en multimedia, juegos, sonido, ofimática, etc.

SISTEMAS OPERATIVOS EN RED Y SERVIDORES.

Los sistemas operativos de red se definen como aquellos que tiene la capacidad de interactuar con sistemas operativos en otras computadoras por medio de un medio de transmisión con el objeto de intercambiar información, transferir archivos, ejecutar comandos remotos y un sin fin de otras actividades. Lo importante es hacer ver que el usuario puede acceder a la información no solo de su máquina, sino a la de cualquier máquina de la red, y esto se consigue gracias a que utiliza un sistema operativo de red.



Hoy en día todos los sistemas operativos de escritorio son sistemas operativos de red también, cosa que no ocurría anteriormente. Normalmente solemos llamar sistemas operativos en red a los sistemas operativos que funcionan como servidores en una red, como es el caso del Windows Server o Linux Server.

SISTEMAS OPERATIVOS DISTRIBUIDOS.

Un sistema distribuido se define como una colección de equipos informáticos separados físicamente y conectados entre sí por una red de comunicaciones distribuida; cada máquina posee sus componentes de hardware y software de modo que el usuario percibe que existe un solo sistema (no necesita saber qué cosas están en qué máquinas). El usuario accede a los recursos remotos de la misma manera en que accede a recursos locales ya que no percibe que existan varios ordenadores, sino que solo es capaz de ver uno formado por todos los anteriores.

Una ventaja fundamental de los sistemas distribuidos, es que permiten aumentar la potencia del sistema informático, de modo que 100 ordenadores trabajando en conjunto, permiten formar un único ordenador que sería 100 veces más potente que un ordenador convencional.

Los sistemas distribuidos son muy confiables, ya que si un componente del sistema se descompone otro componente debe de ser capaz de reemplazarlo, esto se denomina Tolerancia a Fallos.

El tamaño de un sistema distribuido puede ser muy variado, ya sean decenas de hosts (red de área local), centenas de hosts (red de área metropolitana), y miles o millones de hosts (Internet); esto se denomina escalabilidad. De hecho, si un ordenador formando por un sistema distribuido se queda “corto” para las necesidades de la empresa, basta con instalar más.

La computación distribuida ha sido diseñada para resolver problemas demasiado grandes para cualquier supercomputadora y mainframe, mientras se mantiene la flexibilidad de trabajar en múltiples problemas más pequeños. Esta forma de computación se conoce como **grid**.

Los grandes retos de cálculo de hoy en día, como el descubrimiento de medicamentos, simulación de terremotos, inundaciones y otras catástrofes naturales, modelización del clima/tiempo, grandes buscadores de internet, el programa SETI, etc. Son posibles gracias a estos sistemas operativos distribuidos que permiten utilizar la computación distribuida.

El modelo de computación de ciclos redundantes, también conocido como computación zombi, es el empleado por aplicaciones como Seti@Home, consistente en que un servidor o grupo de servidores distribuyen trabajo de procesamiento a un grupo de computadoras voluntarias a ceder capacidad de procesamiento no utilizada. Básicamente, cuando dejamos nuestro ordenador encendido, pero sin utilizarlo, la capacidad de procesamiento se desperdicia por lo general en algún protector de pantalla, este tipo de procesamiento distribuido utiliza nuestra computadora cuando nosotros no la necesitamos, aprovechando al máximo la capacidad de procesamiento. La consola PS3 también cuenta con una iniciativa de este tipo.



Otro método similar para crear sistemas de supercomputadoras es el **clustering**. Un cluster o racimo de computadoras consiste en un grupo de computadoras de relativo bajo costo conectadas entre sí mediante un sistema de red de alta velocidad (gigabit de fibra óptica por lo general) y un software que realiza la distribución de la carga de trabajo entre los equipos. Por lo general, este tipo de sistemas cuentan con un centro de almacenamiento de datos único. Los clusters tienen la ventaja de ser sistemas redundantes, si falla un equipo se resiente un poco la potencia del cluster, pero los demás equipos hacen que no se note el fallo.

En un **cluster** normalmente todos los equipos están ubicados en una **misma red** de área **local**, mientras que en un **grid** los equipos suelen estar distribuidos por **todo el mundo**.

Algunos sistemas operativos que permiten realizar clustering o grid, son: Amoeba, BProc, DragonFly BSD, Génesis, Kerrighed, Mosix/OpenMosix, Nomad, OpenSSI, Plurid.

Un cluster que usamos habitualmente, es el que forma Google. Se estima que en 2010 usaba unos 450.000 ordenadores, distribuidos en varias sedes por todo el mundo y formando clusters en cada una de dichas sedes.

Cada cluster de Google está formado por miles de ordenadores y en los momentos en que se detecta que el sistema está llegando al límite de su capacidad, se instalan cientos de ordenadores más en pocos minutos, aumentando así la potencia de cada cluster. Estos equipos normalmente con ordenadores x86 como los que solemos usar nosotros, corriendo versiones especiales de Linux, modificadas por la propia Google para que permitan la formación de estos clusters. (Buscar en Google “Google centro de datos” para ver algunas informaciones sobre ellos).

SISTEMAS OPERATIVOS POR SU DISPONIBILIDAD.

Dividimos aquí los sistemas operativos por la forma en que se ponen disponibles a los usuarios.

SISTEMAS OPERATIVOS PROPIETARIOS.

Se les denomina propietarios porque son sistemas propiedad de la empresa que los desarrolla. La empresa no vende en realidad el sistema operativo, sino una licencia de uso del mismo. No se tiene acceso al código fuente del sistema, o por lo menos, no se tiene permiso para modificarlo libremente.

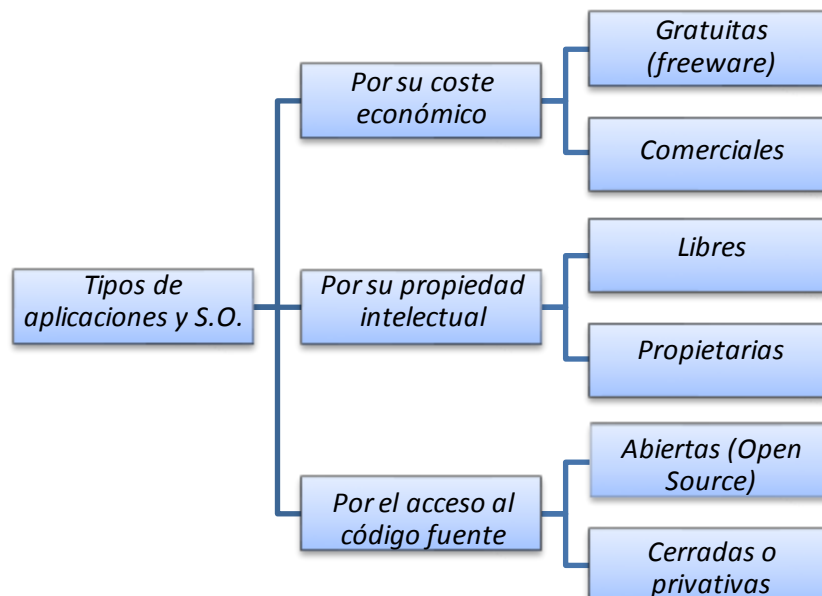
También está prohibido distribuir estos sistemas, o usarlos de formas no autorizadas por la empresa desarrolladora. Toda la familia Windows es un claro ejemplo de sistema operativo propietario.

SISTEMAS OPERATIVOS LIBRES.

Son sistemas operativos en los que se ha renunciado a cualquier tipo de propiedad intelectual. Son sistemas que pueden usarse libremente, ser distribuidos, permiten que se acceda a su código fuente y permiten que esté sea modificado de la forma que queramos.

No hay que confundir el hecho de que sean libres con el hecho de que sean gratuitos. Posteriormente trataremos en profundidad el tema de las licencias de software.

En general, tanto los sistemas operativos como las aplicaciones normales, pueden definirse según su disponibilidad en alguno de estos apartados:



SISTEMAS OPERATIVOS POR SU TIPO DE LICENCIA.

Dentro de los sistemas operativos comerciales, propietarios y privativos, nos podemos encontrar con diversos tipos de licencia de uso:

O.E.M.

OEM (abreviatura del inglés “**O**riginal **E**quipment **M**anufacturer”, en español sería fabricante de equipamiento original). Este tipo de licencias se las otorga el desarrollador del sistema operativo al fabricante de hardware, de modo que cuando nosotros compramos uno de sus productos, este viene con una licencia de uso del sistema operativo de tipo OEM. La particularidad de este tipo de licencias, es el que el sistema operativo viene preparado para ese hardware específicamente, de modo que no tenemos realmente una licencia de uso del sistema operativo, sino una licencia de uso del sistema operativo únicamente para ese hardware en concreto.

Estas licencias son las más económicas, y suelen poseer restricciones especiales, aparte de venir sin manuales ni caja.

RETAIL.

Es la licencia que compramos directamente del desarrollador. Somos propietarios de la licencia, podemos instalarlo en cualquier tipo de hardware compatible, podemos revender la licencia o cederla, etc.

Normalmente solo permiten su uso en una sola máquina a la vez. Vienen con su caja y manuales.

En las licencias de tipo retail, normalmente podemos elegir entre una licencia completa, o una licencia de actualización, que permite actualizar un sistema anterior al nuevo, por un coste algo más reducido.

VLM (LICENCIAS POR VOLUMEN).

Para una empresa con cientos de ordenadores, es complicado controlar las licencias individuales de cada una de sus máquinas. Existe la posibilidad de contratar un tipo de licencia especial con el desarrollador, de modo que, con una única clave de licencia, podemos utilizar varias máquinas a la vez. Es habitual que existan licencias de 25 usos concurrentes, 50, etc.

Son las licencias más caras evidentemente, aunque son bastante más económicas que comprar cada una de las licencias individualmente.

MSDN (LICENCIAS DE EDUCACIÓN).

Son unas licencias especiales de Microsoft que permiten su uso únicamente para actividades educativas y de formación. Cualquier uso de estas licencias en equipos que desarrollen actividades fuera de este ámbito, es ilegal. Existen también licencias de este tipo para empresas de desarrollo, academias, etc.

