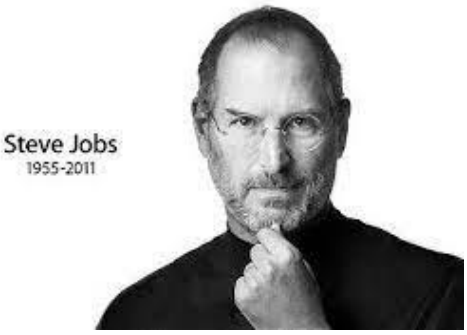
Tema 1: Introducción a los sistemas operativos



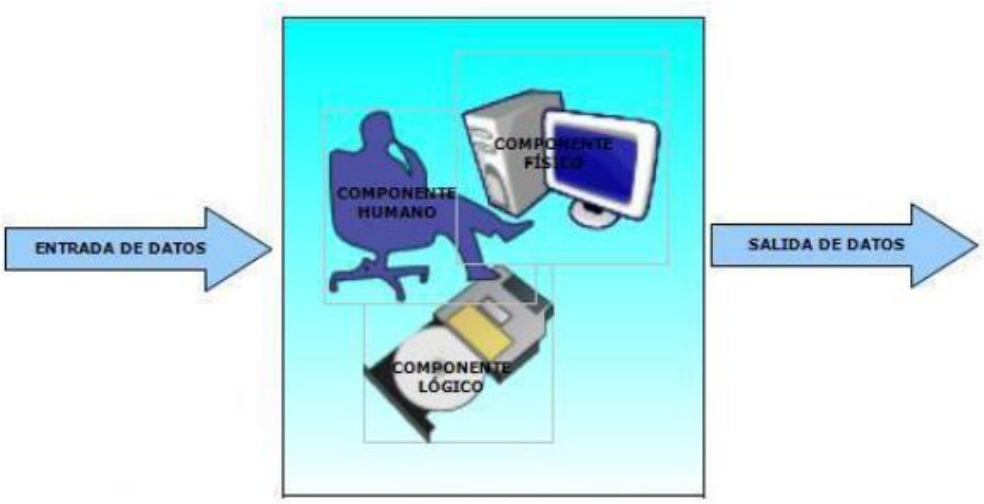
Ken Thompson y Dennis Ritchie,
creadores de Unix y del lenguaje C

Tabla de contenido

1.- Estructura de un Sistema Informático

Para entender la definición de un sistema informático habrá que definir unos conceptos previos como:

- **Informática**: es el conjunto de conocimientos científicos y técnicas que hacen posible el tratamiento automático de la información por medio de los ordenadores.
- **Ordenador**: máquina electrónica dotada de una memoria de gran capacidad y de métodos de tratamiento de la información, capaz de resolver problemas aritméticos y lógicos gracias a la utilización automática de programas registrados en ella. Formará parte del hardware o componentes físicos encargos de tratar la información
- **Programa** informático: es el conjunto de instrucciones que ha de ejecutar un ordenador para realizar una tarea dada. Un programa es una secuencia de instrucciones u ordenes que permiten a un ordenador procesar una información conocida como datos de entrada (*input*) para producir una información de salida (*ouput*) o resultados.
- **Lenguaje** de programación: Un programa está escrito en un lenguaje de programación determinado. Formará parte del software o componente lógico encargado de procesar la información



Podemos considerar un **Sistema informático** (S.I.) a un conjunto de elementos interconectados o relacionados para el tratamiento de información. El más básico es un sólo ordenador que recibiendo datos del exterior y mediante un programa informático almacenado en su memoria procesará los datos para emitir unos resultados. Otros S.I. más complejos son las redes (varios ordenadores conectados entre sí). Sin la intervención humana el sistema informático no podría operar ya que necesita de personas que lo manejen, diseñen, implanten y exploten.

Las **computadoras se pueden clasificar** como:

- De uso **general**: ejecutan todo tipo de aplicaciones.
- De uso **específico**: ejecutan aplicaciones con un único propósito de servicio.
- **Supercomputadora**: procesan grandes cantidades de información en poco tiempo.
- **Macrocomputadores** o mainframes: ordenadores grandes y rápidos, son capaces de controlar cientos de usuarios simultáneamente, utilizados para controlar grandes redes de comunicación, soportan más programas que las supercomputadoras.
- **Minicomputadoras**: se encuentran entre los mainframes y las estaciones de trabajo, permiten el multiproceso (varios procesos a la vez o en paralelo) y pueden soportar hasta unos 200 usuarios a la vez. Se utilizan para almacenamiento de información como bases de datos y para aplicaciones multiusuario en red, como servidores de pequeñas redes.
- **Microcomputadoras** o computadores personales (PC): son ordenadores de uso profesional o personal, pueden ser de sobremesa o portátil, cuando se conectan a una red actúan con un software con función de estación de trabajo dentro de una LAN (red de área local)

2.- Arquitectura de un Sistema Operativo

Un sistema informático tiene muchos recursos (tiempo CPU, memoria, periféricos, etc). El Sistema Operativo actúa como gestor de estos recursos y los asigna a programas y usuarios específicos, según las necesidades, para que realicen sus tareas. En resumen, es un interfaz entre la máquina y el usuario.

Un sistema operativo (S.O.) o **software de base**, consiste en un software formado por un conjunto de programas que sirve para controlar e interactuar con el sistema, proporcionando control sobre el hardware (administración de dispositivos) y dando soporte a otros programas como los que forman el llamado software de aplicación. Por destacar algunas de las tareas que realiza son:

- la administración de los dispositivos periféricos
- Control de temperatura del microprocesador
- Se encarga de la transferencia de datos entre la memoria principal y los dispositivos de almacenamiento.

Los S.O. se pueden encontrar en la mayoría de los aparatos electrónicos que utilicen microprocesadores. Es el primer programa que se carga en el ordenador como responsable de la forma en que se utilice éste. El mismo equipo hardware trabajará de una forma u otra dependiendo del tipo de sistema que se instale en él. El S.O. se comunica con el usuario o persona que utiliza el ordenador mediante el llamado **interface** (API) que se puede presentar en un entorno de trabajo en modo texto o gráfico (en forma de ventanas de diálogo), de esta manera el administrador o usuario de la máquina puede configurar su sistema para que actúen de una cierta manera y adaptarla a sus necesidades.

Objetivos de un S.O.:

- Comodidad: un S.O por medio de la interfaz hace que el ordenador sea más fácil de utilizar por el usuario. Interfaz de usuario son todos aquellos procedimientos que ofrece el sistema operativo para facilitar el trabajo entre el usuario y él mismo. La interfaz entre el usuario y el sistema tendrá que basarse necesariamente en un lenguaje, ya sea textual (MS-DOS) o gráfico (Windows – Linux).

- Eficiencia: un S.O gestiona los recursos para que se aprovechen de una manera más eficaz, como gestor de todos los elementos que componen el sistema, la función del S.O. es proporcionar a los programas una asignación ordenada y controlada de los procesadores, memoria y periféricos, es decir, los recursos del sistema.
- Evolución: un S.O debe de ser capaz de evolucionar, de forma que permita añadir nuevas funciones al sistema sin interferir en los servicios que brinda. Los ordenadores evolucionan muy rápido y el sistema operativo debe controlar cada uno de los nuevos dispositivos.

    - Para admitir nuevo tipo de hardware y actualizaciones del ya existente.
    - Para ofrecer nuevos servicios como respuesta a las demandas del usuario o a las necesidades de los administradores del sistema.
    - Para poder corregir fallos que se descubren con el paso del tiempo en el S.O.

Evolución de los S.O.

- Procesos en serie – Años 40-50 - Transistores.

  El trabajo en ejecución tenía el control total de la máquina y no se comenzaba otro trabajo hasta que el primero terminaba u ocurría algún error. En ambos casos, el S.O. continuaba con el siguiente. El usuario no tenía acceso a la máquina. Era el operador del computador quién agrupaba secuencialmente los trabajos (en tarjetas o cintas).

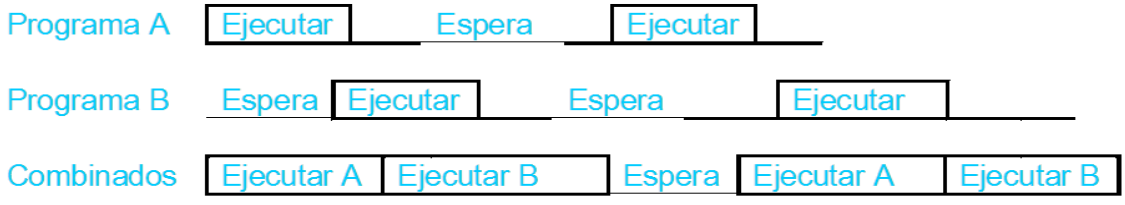
- S.O. interactivos – 1965 a 1980 – Circuitos integrados.

  El S.O. ofrece al usuario la posibilidad de comunicarse con el ordenador. Avances importantes son: Multiprogramación, multiproceso, tiempo compartido y el desarrollo del sistema de ficheros que protege más a los datos. Aparecen los conceptos de interrupción, buffering y spool.

    - Interrupción, es una señal que se envía al procesador mediante la cual la memoria o los dispositivos pueden interrumpir la ejecución del trabajo del procesador. Mediante las interrupciones, el S.O. puede enterarse de eventos tales como: la finalización de lectura/escritura en un periférico, un fallo en el hardware.

    - Multiprogramación y Multiproceso.

  Multiprogramación es la capacidad del procesador de ejecutar varios programas a la vez. Los programas residían en memoria y sólo uno de ellos era ejecutado por la CPU. Cuando dicho programa realizaba alguna operación de E/S el procesador ejecutaba el siguiente programa. De esta manera la CPU no estaba desocupada. Cuando el programa ha terminado de realizar la operación de E/S el procesador continuará ejecutándolo.



Multiproceso son sistemas con varios procesadores en un único ordenador (compartiendo memoria y periféricos).

- Tiempo compartido.

El S.O. utiliza la planificación de la CPU y la multiprogramación para dotar a cada usuario de una parte del ordenador. La CPU es utilizada por cada usuario durante un breve período de tiempo ya que las acciones o comandos son breves. Al cambiar de un usuario a otro rápidamente, éstos tienen la impresión de disponer cada uno de un ordenador.

- Sistemas en tiempo real.

  Los sistemas han de ser capaces de dar respuestas a cualquier petición de forma inmediata. En estos sistemas lo importante es la rapidez de la respuesta.

- Sistemas distribuidos y redes de ordenadores.

  Estos sistemas deben dar soporte a la comunicación y a la sincronización rápida de miles de procesadores. Cada procesador dispone de sus propios dispositivos. En estos sistemas se pueden conectar ordenadores con distintos procesadores y compartir recursos.

  Modelos de S.O según su estructura interna en su diseño:

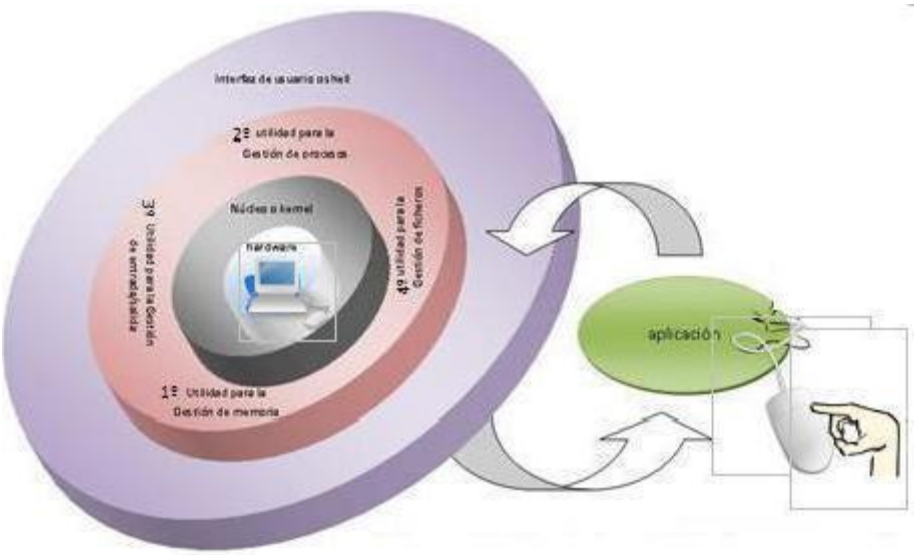
- Diseño **monolítico**: el sistema está constituido por un único programa compuesto de múltiples rutinas o subprogramas que pueden ser llamadas unas a otras ya que cualquier procedimiento puede invocar a otro. Se utilizó en los primeros sistemas operativos. La arquitectura más simple para un S.O. es un núcleo compacto, que contiene todas las rutinas de S.O., ejemplo: Linux
- Diseño en **capas**: está constituido por una serie de capas o anillos que se comunican entre sí atendiendo a las funciones que puede realizar. El sistema operativo consta de una estructura que parte de una capa núcleo que tiene relación con el hardware y se va completando en capas de modo que cada capa suministra servicio a la capa siguiente. Los servicios que brinda cada capa son expuestos en una interface pública y son consumidos solamente por los de la capa de arriba. Diseño más modular y escalable que el monolítico. Ejemplo: OS/2
- **Máquinas virtuales**: permite emular mediante software sistemas operativos, una máquina o una red de computadora. El software emulador traduce las peticiones hechas a la máquina virtual en operaciones sobre la máquina real. Se pueden ejecutar varias máquinas virtuales al mismo tiempo. Estas máquinas virtuales no son máquinas extendidas, sino una réplica de la máquina real, de manera que en cada una de ellas se pueda ejecutar un sistema operativo diferente, que será el que ofrezca la máquina extendida al usuario. Los recursos de hardware se reparten entre las distintas máquinas virtuales por lo que se necesita altas prestaciones de hardware. Ejemplo: Java, VMware, VirtualBox.
- Modelo **cliente/servidor**: según este modelo, el SO se organiza como un conjunto de módulos autónomos, cada uno de los cuales tiene a disposición del resto una serie de servicios. Cada módulo actúa como un servidor de ciertas funcionalidades, que atiende las peticiones de otros módulos y que su vez puede ser cliente de otros módulos. Los procesos o servicios pueden ser tanto servidores como clientes. El sistema operativo es el encargado de mantener la comunicación y organización entre procesos o servicios.

- **Micronúcleos:** se constituye de un núcleo que brinde un manejo mínimo de procesos, memoria y, además, provea de una capa de comunicación entre procesos. La capa de comunicación es la funcionalidad principal del sistema. Los restantes servicios del sistema son construidos como procesos separados al micronúcleo que ejecutan en modo usuario. El acceso los servicios del sistema se realiza a través de pasaje de mensajes. Ejemplo: Windows

2.1.- Componentes de un sistema operativo

Dentro de un sistema operativo podemos destacar los siguientes componentes o niveles:

- **El "Kernel"** o núcleo, es un programa multihebra o multihilo que reside permanentemente en memoria. Se encarga principalmente de controlar la CPU, es decir gestionar el Procesador.
- En los siguientes niveles podemos encontrar los **programas de utilidad**. Podemos realizar la siguiente clasificación por la función que realizan:
    - Utilidades para la gestión de memoria: se encarga de administrar la memoria para los procesos y programas, repartiendo la memoria disponible entre los distintos procesos.
    - Utilidades para la gestión de procesos: controla los procesos en ejecución en tareas como inicio, parada, coordinación, la creación y destrucción de procesos, intercambio, detección y arranque de mensajes.



    - Utilidades para la gestión de E/S a disco: gestiona la comunicación entre dispositivos que se encargan de la E/S de la información y de su almacenamiento en función de los dispositivos existentes.
    - Utilidades para la gestión de ficheros y de la información: cuyo objetivo es el de controlar los archivos para mantener una correcta organización dentro y fuera del sistema, realizando tareas como la asignación de nombres, permisos, atributos, etc. a los ficheros y programas. Gestiona los nombres lógicos y la protección de la información realizando funciones de creación y destrucción de ficheros, lectura y escritura y protección de accesos.

- Programa **interface de usuario o Shell**: encargados de permitir al usuario la comunicación con el sistema por medio de entornos gráficos o de texto mediante una línea de entrada de comandos.

2.1.1.- Núcleo de los sistemas operativos

En informática, el **núcleo** o **kernel** es la parte fundamental de un sistema operativo. La mayoría de los sistemas operativos se construyen en torno al concepto de núcleo. Acceder al hardware directamente puede ser realmente complejo, por lo que los núcleos suelen implementar una serie de abstracciones del hardware. Esto permite esconder la complejidad, y proporciona una interfaz limpia y uniforme al hardware lo que facilita su uso para el usuario.

En informática, el núcleo de un sistema operativo, es el programa informático formado por un conjunto de subrutinas o módulos de programa que permiten algunas de las siguientes **funcionalidades**:

- **La comunicación** entre los programas informáticos y el hardware. Responsable de facilitar a los distintos programas acceso seguro al hardware de la computadora o en forma más básica
- **Gestión de las distintas tareas** o **procesos** de una máquina. Como hay muchos programas y el acceso al hardware es limitado, el núcleo también se encarga de decidir qué programa podrá hacer uso de un dispositivo de hardware y durante cuánto tiempo.
- Gestión del hardware (memoria, procesador, periférico, forma de almacenamiento, etc.). Es el encargado de **gestionar recursos**, a través de servicios de llamada al sistema.
- Los núcleos garantizan la carga y la ejecución de los procesos mediante el módulo llamado cargador responsable de cargar programas en memoria, se carga al iniciar el sistema y permanece en memoria hasta que el sistema se apaga. Los enlazadores dinámicos son otro tipo de cargador que carga y liga librerías dinámicas (archivos con extensión dll o so).

2.1.2.- Programas de utilidad de un sistema operativo. Interprete de comandos o Shell.

El S.O dispone de **módulos o programas útiles** que junto con el núcleo al ser ejecutados gestionan recursos como: el control de las funciones de la CPU, los soportes y dispositivos que llevan a cabo la entrada/salida de información del ordenador, el almacenamiento de información en la memoria central o principal, los procesos o programas que se están ejecutando en un instante dado, etc.

Dentro de todas las **funciones que controla el S.O** podemos destacar las siguientes, como principales:

- La **gestión de procesos** o programas que se ejecutan mediante las tareas de crear, eliminar, detener, reanudar, comunicación y sincronización en el uso de la CPU, memoria y dispositivos de la máquina.
- El **control de las direcciones** de la memoria principal donde se almacenan los procesos y datos en ejecución, controlando los espacios de memoria libre y utilizada, tablas de localización de una información concreta, etc.
- La **gestión del sistema de entrada/salida** de datos y ficheros, realizando tareas como el mantenimiento de datos en almacenamiento secundario o externo con una planificación

de los diferentes volúmenes de discos, la gestión de la memoria de almacenamiento temporal o memoria caché.

- La **gestión del sistema de archivos** permitiendo la organización relacionada del almacenamiento de los datos y ficheros mediante la asignación de unidades y directorios. Existen diferentes tipos de sistemas de archivos como son FAT32, EXT2, NTFS, etc.
- **Otras utilidades** como son: el sistema de auditorías para la protección de programas, un sistema de comunicación basado en red para intercomunicar unos sistemas con otros mediante interfaces de red, soporte para la creación propia de procesos mediante la oferta de lenguajes de programación (compiladores, interpretes, etc.), procesos para la información del estado del sistema, etc.

  Un **intérprete de comandos o Shell** es un programa informático que actúa como interfaz de usuario para comunicar al usuario con el sistema operativo mediante pantalla gráfica o ventana que espera órdenes escritas por el usuario con el teclado, los interpreta y los entrega al sistema operativo para su ejecución.

  La respuesta del sistema operativo se muestra al usuario en la misma ventana o abriendo otros interfaces gráficos en su caso. La parte del sistema operativo que realiza esta tarea de interfaz entre el usuario y el ordenador se denomina programa *Shell* que queda esperando más instrucciones o eventos del usuario.

  El sistema operativo Windows trae una Shell llamada Windows **PowerShell**, que combina características de las tradicionales Shell de Unix con su framework orientado a objetos .NET. Algunos ejemplos de Shell de Unix (ksh, csh, bash, tcsh, Bourne Shell, etc.),

  Por extensión, también se llama *intérprete de comandos* a algunas interfaces de programas específicos que comunican al usuario con el software o al cliente de un servidor como, por ejemplo MySQL, OpenSSL, FTP, etc.

  Los interpretes de comandos suelen incorporar características tales como control de procesos, redirección de entrada/salida, listado y lectura de ficheros, protección, comunicaciones y un lenguaje de órdenes para escribir programas por lotes o scripts o guiones, tuberás, etc.

  Su posibilidad potencial de trabajo es generalmente en modo texto mediante órdenes escritas en una línea de comandos, aunque algunos sistemas presentan la posibilidad de trabajar en una interfaz gráfica que facilita al usuario la operatividad con el ordenador a costa de mayor consumo de recursos computacionales y una mayor vulnerabilidad en la seguridad.

3.- Funciones o servicios de un Sistema Operativo

Los sistemas operativos, en su condición de software están formados por un conjunto de rutinas o módulos que posibilitan y simplifica el manejo de la computadora, desempeñan una serie de funciones básicas esenciales para la gestión del equipo. El SO en su diseño tiene que brindar las siguientes posibilidades:

- **Interfaces del usuario**: es la parte del sistema operativo que permite comunicarse con él, de tal manera que se puedan cargar programas, acceder archivos y realizar otras tareas. Proporciona más comodidad en el uso de un computador. Existen tres tipos básicos de

interfaces: las que se basan en comandos, las que utilizan menús y las interfaces gráficas de usuario.

- Administración de recursos: sirven para administrar los recursos de hardware y de redes de un sistema informático, como la CPU, memoria, dispositivos de almacenamiento secundario y periféricos de entrada y de salida.. Dos de las funciones más importantes que realiza en este apartado son: la administración de periféricos (coordinando y manejando los distintos dispositivos conectados a la máquina) y administración de memoria (asignando y gestionando la memoria del sistema a los distintos procesos)
- Administración de archivos: un sistema de información contiene programas de administración de archivos que controlan la creación, borrado y acceso de archivos de datos y de programas. También implica mantener el registro de la ubicación física de los archivos en los discos magnéticos y en otros dispositivos de almacenamiento secundarios.
- Administración de tareas o control de la ejecución de programas: acepta los trabajos, administra cómo se realizan y les asigna recursos. Los programas de administración de tareas de un sistema operativo administran la realización de las tareas informáticas de los usuarios finales. Las funciones de administración de tareas pueden distribuir una parte específica del tiempo del CPU para una tarea en particular, e interrumpir al CPU en cualquier momento para sustituirla con una tarea de mayor prioridad, es decir, gestiona el llamado control de concurrencia estableciendo prioridades entre los distintos procesos que desean utilizar un mismo recurso
- Servicios de soporte o actualización del sistema: los servicios de soporte de cada sistema operativo dependerán de la implementación particular de éste con la que estemos trabajando. Entre las más conocidas se pueden destacar las implementaciones de Unix, desarrolladas por diferentes empresas de software, los sistemas operativos de Microsoft, y las implementaciones de software libre, como GNU/Linux, etc. Estos servicios de soporte suelen consistir en:
    - Actualización de versiones.
    - Mejoras de seguridad.
    - Inclusión de alguna nueva.
    - Controladores para manejar nuevos periféricos.
    - Corrección de errores de software.
- Control de seguridad. Proporciona seguridad para la información almacenada y los usuarios del sistema realizando una gestión de permisos y de usuarios para evitar conflictos entre los distintos trabajos.

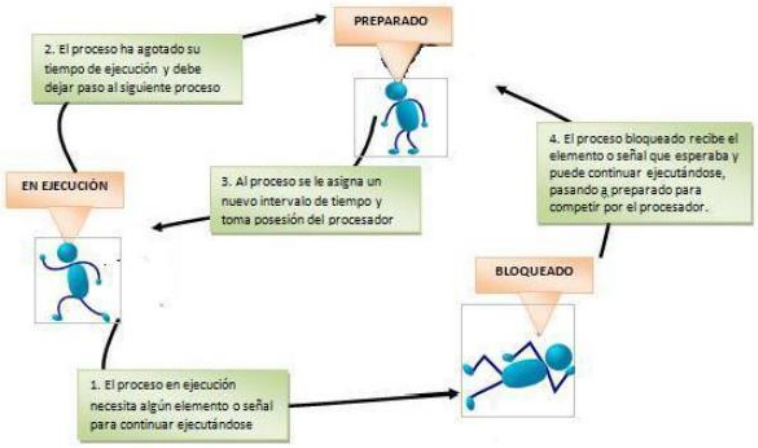
3.1.- Controlar los procesos

Un proceso es un programa o tarea en ejecución al cual el S.O asignará recursos y controlará su ejecución. Cada proceso está formado por órdenes o instrucciones que se cargan en memoria para su ejecución, en su almacenamiento en memoria se crea una estructura de datos que sirve para identificar cada proceso y permite controlar los aspectos de su ejecución denominada Bloque de Control de Proceso (BCD). El módulo del SO denominado **cargador** es el encargado de cargar en memoria virtual el proceso demandado en la llamada cola de procesos con el estado de preparado, creando el bloque de control de proceso representado por un identificador de procesos, seguidamente le asigna una prioridad y los recursos necesarios para su correcto funcionamiento.

La información que nos aporta el sistema en la estructura de Bloque de Control de Proceso generada para cada proceso es:

- **Estado del proceso:** puede presentar los siguientes estados:

    - **Preparado o listo**: se encuentran todas las tareas que están listas para ejecutarse pero que esperan a que el procesador quede libre ya que hay otros procesos más prioritarios en ejecución. Posteriormente al proceso se le asigna un nuevo intervalo de tiempo y tomará posesión del procesador al envió de una señal.
    - **Ejecución o activo**: cuando el proceso recibe alguna señal para continuar ejecutándose. En el caso de sistemas con un único procesador, sólo puede haber un proceso en dicho estado en un instante dado. EL tiempo de uso del microprocesador se reparte entre todos los procesos cargados de manera que el usuario cree que se están ejecutando varias tareas a la vez, sin embargo, en un tiempo dado solamente se ejecuta una.
    - **Bloqueado o suspendido**: sucede cuando el proceso ha agotado su tiempo de ejecución y debe dejar paso al siguiente proceso. Los procesos están a la espera de que se cumplan alguna condición o recibir una señal para reanudar la ejecución.
    - **Muerto**: un proceso está en este estado cuando ha terminado su ejecución de manera correcto o porque se ha producido un error en su ejecución.
    - **Nonato o ignorado**: el proceso existe, pero todavía no es conocido por el S.O



- Código de identificación del proceso o **pid**.
- **Valor de prioridad** a la hora de asignar los recursos del sistema.
- **Direcciones** o zona de memoria asignada
- **El estado hardware** (contador de programa, códigos de condición, punteros de pila, etc.), información para gestionar la memoria (punteros, tablas, registros), información de estado del sistema de E/S (dispositivos de E/S asignados al proceso, lista de archivos abiertos, etc.).

En un instante determinado el sistema tendrá un estado general, indicado por el conjunto de recursos y procesos existentes con sus estados correspondientes dentro del propio sistema; este estado global cambia en el momento que se solicite respuestas a los eventos

generados externa e internamente modificando el estado de los procesos y la asignación de los recursos.

3.1.1.- Planificador de procesos

Todos los S.O. admiten la multiprogramación. Esto supone una planificación de procesos, bien en tiempo compartido, bien con varios procesadores. El tiempo compartido consiste en dividir el tiempo de ejecución del procesador en pequeños intervalos de tiempo, decenas o cientos de milisegundos, e ir asignando un intervalo de ejecución a cada uno de los procesos que se están ejecutando, de este modo se consigue ejecutarlos, poco a poco, de forma paralela.
Cuando diversos procesos están listos para ejecutarse, el S.O debe decidir cuál de ellos ha de utilizar el procesador y cuándo un proceso puede pasar de un estado a otro. El módulo encargado de esta tarea se denomina **planificador de procesos** o **scheduler**. Para ello utiliza un Algoritmo de planificación.

Funciones y objetivos del planificador:

- **Equidad**: al asignar el tiempo de utilización del procesador de la forma más justa posible
- **Eficiencia**: dar servicio al número máximo posible de procesos para conseguir que el procesador esté ocupado el mayor tiempo posible.
- **Tiempo de respuesta bajo**: garantizar buenos tiempos de respuesta a los usuarios mediante la disposición de recursos suficientes cuando son necesarios.
- **Alto rendimiento**: al maximizar el número de procesos que se ejecutan en un periodo de tiempo, activando los procesos que están en el estado preparado

La misión del planificador es:

- Garantizar que cada proceso tiene acceso al recurso en su justa medida.
- Explotar el recurso, manteniéndolo ocupado el máximo tiempo posible.
- Reducir todo lo posible los tiempos de espera de los diferentes procesos.

La dificultad de la planificación radica en que no se puede saber, a priori, qué procesos estarán compitiendo por un recurso ni tampoco el contexto que se generaría en caso de concedérselo. Por ello, para controlar el acceso de los procesos a los recursos, el planificador puede seguir dos tipos de política:

- **Política no Expropiativa**: una vez que el proceso accede al recurso, se permite que haga uso de él hasta que ya no lo necesite.
- **Política Expropiativa**: se proporciona acceso al recurso para un proceso, pero, en un determinado momento, se puede ceder el recurso a otro proceso diferente, para lo cual se le retirará el acceso al primero (pasando a estado suspendido, bloqueado o preparado, según corresponda).

La política apropiativa es mucho más justa cuando existen procesos muy diferentes, ya que permite, por regla general, que ningún proceso acapare el recurso. Por otro lado, es más compleja y consume más recursos (en las expropiaciones), por lo que habrá que sopesar cuál es más adecuada en cada situación concreta.

Los procesos no se ejecutan en cualquier orden, sino que siguen un orden establecido por el sistema operativo. La forma en la 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** (en inglés, scheduler).

3.1.1.1 Niveles de planificación.

La planificación es demasiado compleja como para realizarse en una única fase. Los sistemas operativos, en general, disponen de tres niveles de planificación, cada uno con su planificador:

- **Planificación a largo plazo (nivel alto)**: llamada también "planificación de admisión", ya que es la que determina qué trabajos se admiten para su procesamiento y, por consiguiente, se cargan en memoria.
- **Planificación a medio plazo (nivel medio)**: gestiona el estado de suspensión de los procesos. Esta operación recibe el nombre de intercambio (o swapping). Es muy típica en sistemas Linux, donde incluso existe una partición específica para este fin.
- **Planificación a corto plazo (nivel bajo)**: también llamada despachador (dispatcher). Estipula qué procesos en estado preparado pasarán a ejecución. Debe ser una planificación sencilla y breve, ya que se ejecutará muchas veces.

3.1.1.2.- Algoritmos de planificación.

Los planificadores funcionan aplicando algoritmos de gestión. De entre todos los existentes, destacamos estos:

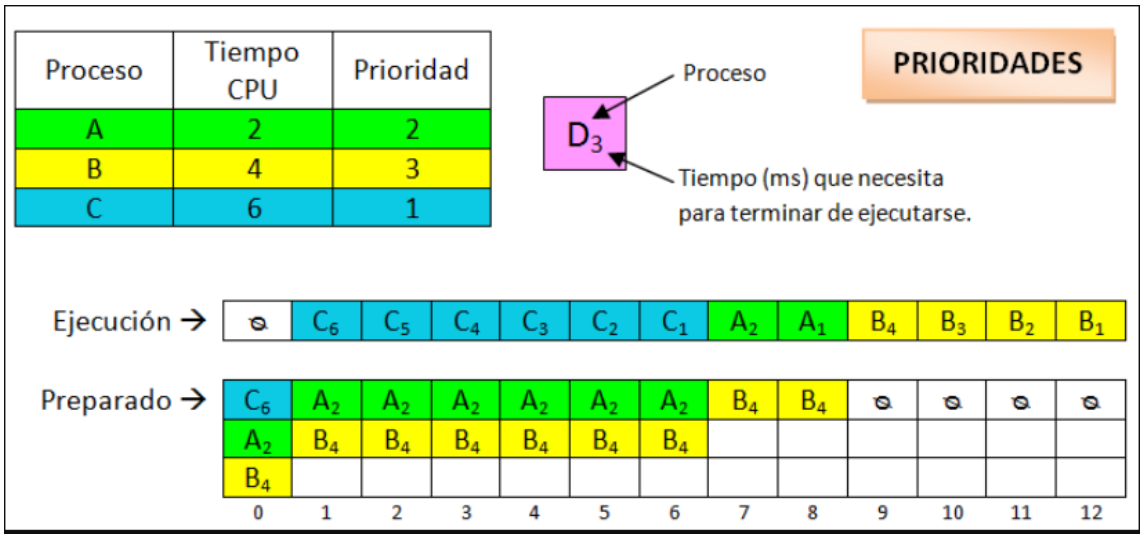
Algoritmo por prioridades.

- A cada proceso se le asigna una prioridad según la cual se ejecutan. Asigna los tiempos de ejecución según una lista de prioridades. Es lógico pensar que hay procesos más prioritarios que otros. Por ejemplo, un proceso del administrador del sistema que sirva para solucionar un problema deberá tener más prioridad que la ejecución de una consulta de un usuario.

  Este algoritmo tiene el inconveniente de que los procesos con prioridad baja pueden relegarse en el tiempo.

  Para reflejar la existencia de varias prioridades en un sistema multiusuario se emplea un algoritmo de planificación con varias listas de procesos en ejecución, una por cada una de las prioridades que se puedan establecer. Generalmente, si se está ejecutando un proceso de prioridad media y entra un proceso de prioridad mayor, se requisa la CPU al primer proceso y se le entrega al proceso de mayor prioridad, este es el llamado algoritmo de prioridad **apropiativa**. Es uno de los más complejos y eficaces. Si por el contrario el proceso

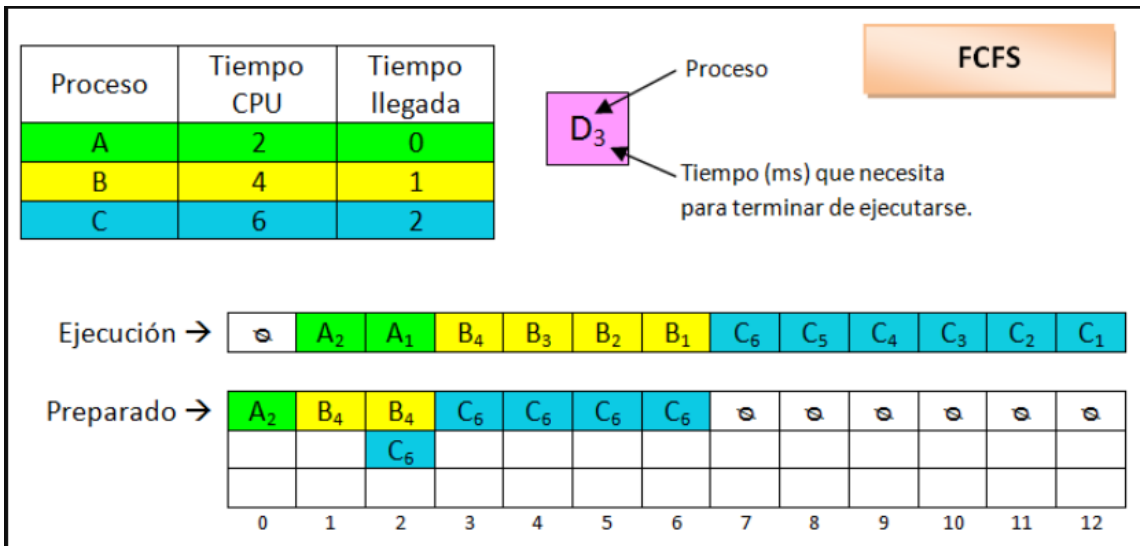
entra en ejecución y este se ejecuta en su totalidad se llama algoritmo de prioridades **apropiativo**.



Algoritmo FCFS (Fisrt Come First Served).

- Utiliza una estructura de cola en la que los procesos se ejecutan según entran en ella. (primero en llegar es el primero en ejecutar). El primero en entrar no libera los recursos hasta que no termina. Es el más sencillo pero el más ineficaz por su menor rendimiento.

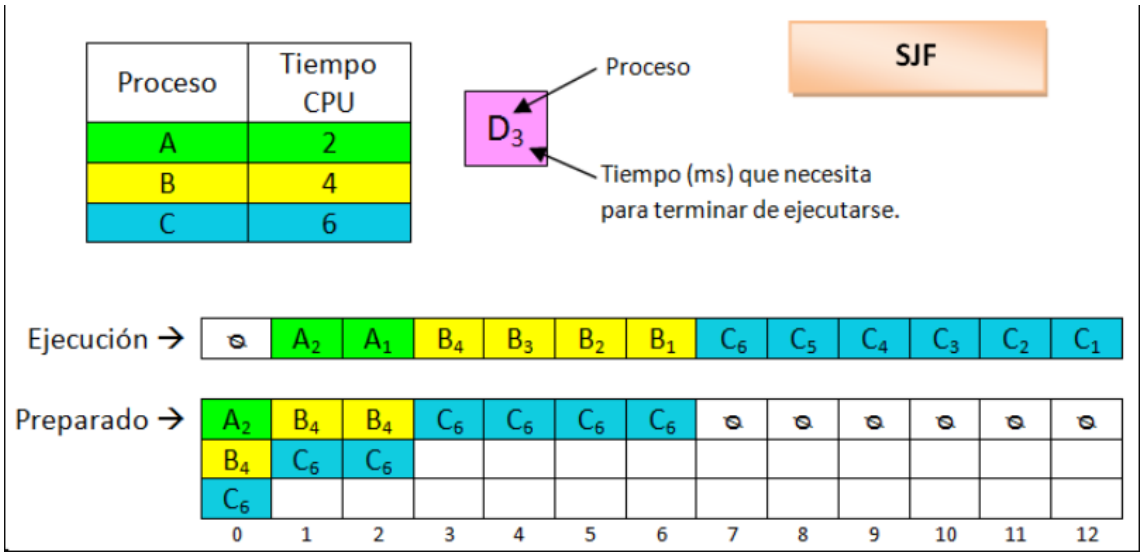
  Aquí, los procesos demasiado largos harían esperar al resto de procesos hasta que termine de ejecutarse.



Algoritmo SJF (Shortest Job First).

(primera tarea más corta). El trabajo más corto se ejecuta primero. Intenta reparar el problema de FCFS pero, en este caso, los procesos largos se ven desfavorecidos y pueden retrasarse en su ejecución continuamente.

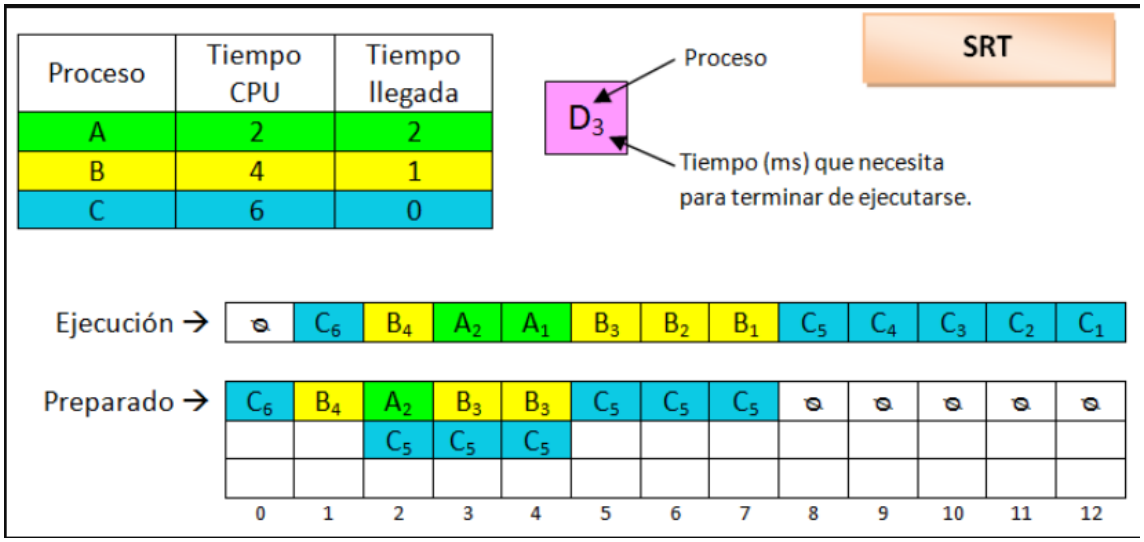
Una vez que el proceso entra en ejecución, se ejecuta por completo, aunque haya en cola procesos más cortos.



Algoritmo SRT (Shortest Remaining Time).

Es una versión ex_propiativa de SJF, donde se tiene en cuenta también los procesos de la cola.

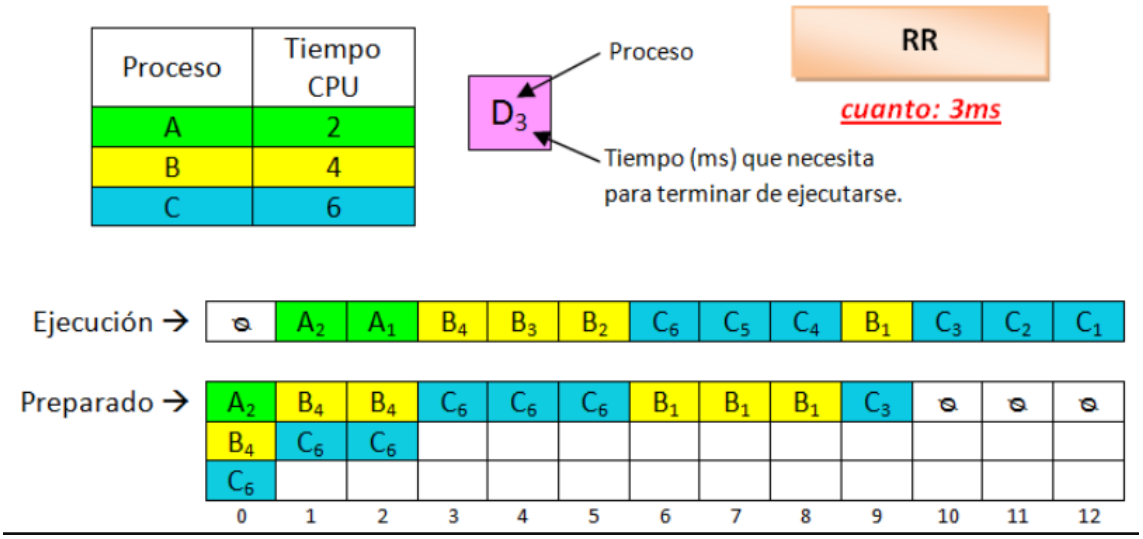
En el caso de que existan procesos en cola más cortos, se expropia el proceso en ejecución y se ejecutaría el de la cola.



Algoritmo RR (Round Robin).

Utiliza una organización en **cola circular**: Los procesos se ejecutan en cola y cuando acaba el último se sigue con el primero. A cada proceso se le asigna un tiempo de uso de CPU denominado cuánto.

El problema de este algoritmo está en la fijación del cuanto, ya que cuantos demasiado largos degeneran en FCFS y cuantos demasiado cortos disminuirían el rendimiento por los continuos cambios de contexto de los procesos.



Algoritmo de **rueda** (Round-Robin) o **RR** (prioridad circular), Asigna secuencialmente el mismo tiempo de ejecución (**quantum** o cuanto) a los diferentes procesos en forma rotatoria. Primero se ejecuta el proceso P1 y cuando agota su intervalo de tiempo pasa a ejecutarse el proceso P2, cuando éste agota su intervalo pasa a P3, si un proceso necesita un tiempo de ejecución mayor de su quantum asociado, una vez transcurrido este y si existen más procesos en espera de ejecución, se colocan al final de la lista del estado preparado y el procesador pasa al proceso que queda en cabeza de la lista es decir se vuelve a ejecutar P1, formado así un ciclo de procesos en ejecución. Cuando un proceso termina de ejecutarse se retira del ciclo y quedan los restantes procesos compitiendo por el procesador. Si llega un proceso nuevo a ejecución, entrará en el ciclo y esperará su turno para comenzar a ejecutarse. Cuando un proceso tiene que realizar una operación de E/S el procesador ejecutará el siguiente proceso. Cuando el proceso en espera haya terminado la operación, volverá a ser ejecutado, pero siempre respetando el turno. Es de los más sencillo, justo y de uso más amplio.

Como se puede observar, cada algoritmo tiene sus ventajas y sus inconvenientes. En la práctica se utiliza un sistema híbrido de varias colas en las que se aplican diferentes algoritmos según las necesidades y el momento concreto con la finalidad de optimizar los recursos y los tiempos de respuesta del sistema.

Medidas o valores para evaluar los algoritmos de planificación son:

- Tiempo o ráfaga de uso de la CPU: se expresa como un porcentaje del tiempo medio de utilización, es decir, el porcentaje de tiempo en el que el procesador está ocupado.
- Productividad (P): el número de procesos o trabajos ejecutados por unidad de tiempo.(P= Nº procesos completados/Segundos)
- Tiempo o ráfaga regreso o finalización (TF): es la suma del tiempo de ejecución real o útil y el tiempo consumido en la espera por los recursos. TF=E+U (tiempo CPU). También se

puede llamar tiempo de servicio ya que es el tiempo que tarda en ejecutarse un proceso desde carga, espera, ejecución. (tiempo_realiza_proceso=tiempo_termina_ejecutar-tiempo_empieza_ejecución)

- Tiempo de espera (E): es el tiempo que el proceso espera hasta que se le concede el procesador, es decir, el tiempo que ha estado en estado de preparado o listo.(espera=tiempo_realiza_proceso-tiempo_ejecutandose)
- Tiempo de servicio: tiempo que tarda en ejecutarse un proceso desde carga, espera, ejecución y en accesos de entrada/salida. (tiempo_realiza_proceso=tiempo_termina_ejecutar-tiempo_empieza_ejecución)

3.2.- Procesos.

Un proceso es un programa o un fragmento de programa en ejecución.

Cada proceso necesita una serie de recursos, siendo el sistema operativo el encargado de proporcionárselos. En función del tipo de proceso, los recursos necesarios pueden ser diferentes, pero, por regla general, todos necesitan un espacio en memoria y un tiempo de uso del microprocesador (CPU).

Todos los procesos, desde su creación, tendrán la capacidad de comunicarse y sincronizarse con otros procesos y con recursos del sistema. Este hecho da lugar a diferentes **tipos de procesos**:

- **Independientes**: no se comunican con otros procesos. Estos tipos de procesos apenas existen.
- **Cooperativos**: se comunican y sincronizan para realizar una actividad común.
- **Competitivos**: necesitan hacer uso del mismo recurso y, por consiguiente, competirán por él.

En cualquier caso, la ejecución de los procesos exigirá concurrencia, cualidad que deberá gestionar el sistema operativo gracias a técnica denominada **multiprogramación**, que permite que dos o más procesos puedan ejecutarse "a la vez".

De todos los procesos que se están ejecutando a la vez sólo uno tiene la "atención del usuario"; Este proceso se dice que está en **primer plano** y del resto se dice que están en **segundo plano**. Es posible pasar un proceso de primer plano a segundo plano y viceversa.

3.2.1.- Estados de un proceso

Un proceso, a lo largo de su vida, cambia de estado. En concreto, un proceso puede estar en uno de los siguientes estados:

- **Nuevo (new):** Momento en el que se está creando el proceso.
- **Preparado (ready):** Está esperando a que se le asigne la CPU (un procesador).
- **En ejecución (running):** Se está ejecutando en la CPU.
- **Bloqueado o en espera (waiting):** Está a la espera de que suceda un evento.
- **Terminado (terminated):** Finaliza su ejecución. Muere.

Además de estos estados, en algunos sistemas también existe un estado denominado **suspendido (suspend),** que se produce cuando un proceso se suspende y es desplazado a memoria secundaria. En esta situación, el proceso puede quedar a la espera de un evento (**waiting suspend**) o suspendido temporalmente pero listo para ejecutarse (**ready suspend**).

El proceso puede cambiar de estado cuantas veces sea necesario, pero nunca puede estar en más de un estado a la vez.

El objetivo final de un proceso es procesarse, es decir, estar en ejecución durante el tiempo que sea necesario para ser procesado. Sin embargo, un procesador sólo puede ejecutar un proceso a la vez. Este hecho es el que marca el paradigma de la gestión de procesos en el sistema operativo, mediante el cual se establecen los mecanismos necesarios para que los diferentes procesos que compiten por la CPU puedan hacerlo de la forma más eficiente y beneficiosa para el sistema.

3.2.2.- Creación de procesos.

La creación de un proceso se hace a través de la llamada al sistema "crear proceso" desde otro proceso denominado proceso padre. El proceso resultante se denomina proceso hijo y es prácticamente una réplica de su padre.

Como se puede deducir, debe existir un proceso inicial con la capacidad de pedir al sistema que se cree un proceso. Este proceso creador es de vital importancia en los sistemas operativos y constituye la raíz del árbol de procesos.

Los procesos se marcan en su creación con un número único llamado identificador de proceso (PID). Salvo el proceso raíz, todos los procesos llevan dos números:

- El PID que lo identifica a él.
- El PPID que identifica a su padre.

Después de que un proceso genera un hijo, ambos continúan ejecutándose desde el punto en el que se hizo la creación.

3.2.3.- Terminación de procesos.

Generalmente un proceso finaliza su ejecución cuando pasa al estado terminado y le pide al sistema que lo elimine utilizando la llamada al sistema "exit".

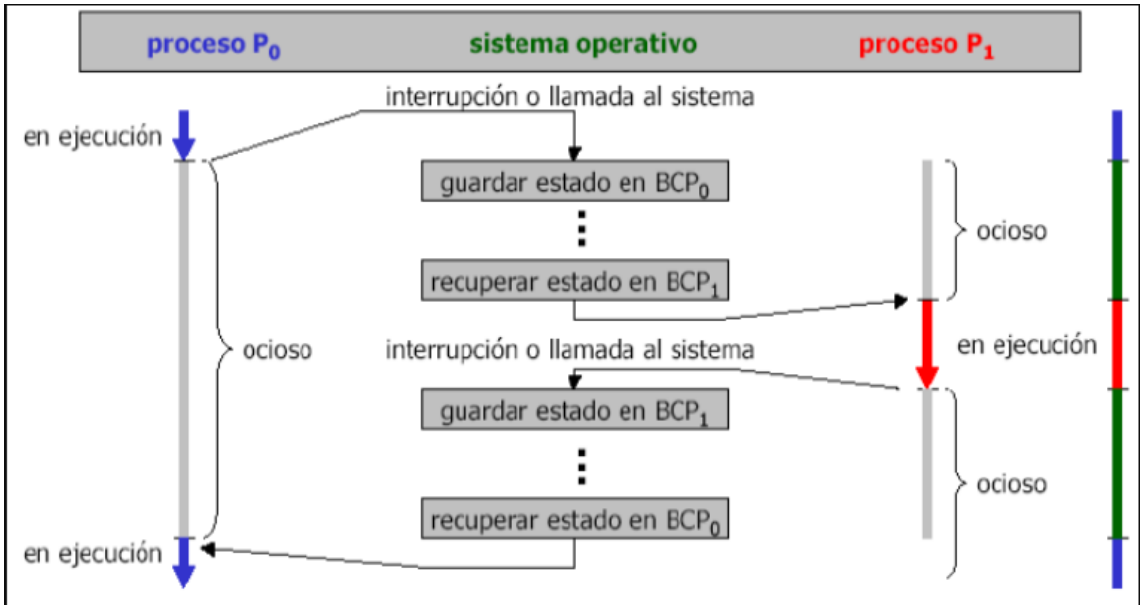
Los procesos hijo pueden ser eliminados cuando el proceso padre quiera. Esto se denomina **matar un proceso** y es habitual cuando:

- El hijo se excede en el uso de los recursos asignados.
- La tarea que se asignó al hijo ya no se necesita.
- El proceso padre quiere terminar y el sistema operativo no permite que los hijos continúen sin el padre.

Hay que prestar especial atención al hecho de que un proceso no debería terminar hasta que todos sus hijos lo hagan. Sin embargo, las operaciones sobre procesos no son fiables al 100% y se pueden dar anomalías en la gestión de los procesos:

- **Procesos huérfanos.** Se denominan así a los procesos que quedan en el sistema cuando su padre ha finalizado. Cuando esto sucede, el PPID del proceso (para sistemas UNIX) pasa a ser el PID del proceso inicial.
- **Procesos zombies.** Son procesos que han finalizado pero su padre los mantiene como vivos. Este tipo de procesos suelen ser fruto de errores de programación o de fallos del sistema. Al contrario que los procesos huérfanos, los zombies no son adoptados por el proceso inicial, sino que tienden a eliminarse para evitar el consumo de recursos.

3.2.4.- Cambio de contexto.



Como hemos visto, un proceso, a lo largo de su vida, puede pasar por diferentes estados. El cambio de un estado a otro no es trivial y tanto la forma como el tiempo para hacerlo marcarán la eficiencia del sistema.

El proceso pasa gran parte de su vida esperando ser ejecutado. Cuando sale de ese estado sin haber finalizado su ejecución (porque es expropiado a otro estado, suspendido o bloqueado) se espera que la próxima vez que alcance el estado de ejecución continúe donde lo había dejado.

Para que esto sea posible se aplica una operación conocida como cambio de contexto.

El cambio de contexto consiste en interrumpir la ejecución de un proceso para comenzar o seguir con otro.

De forma un poco esquemática, en el cambio de contexto:

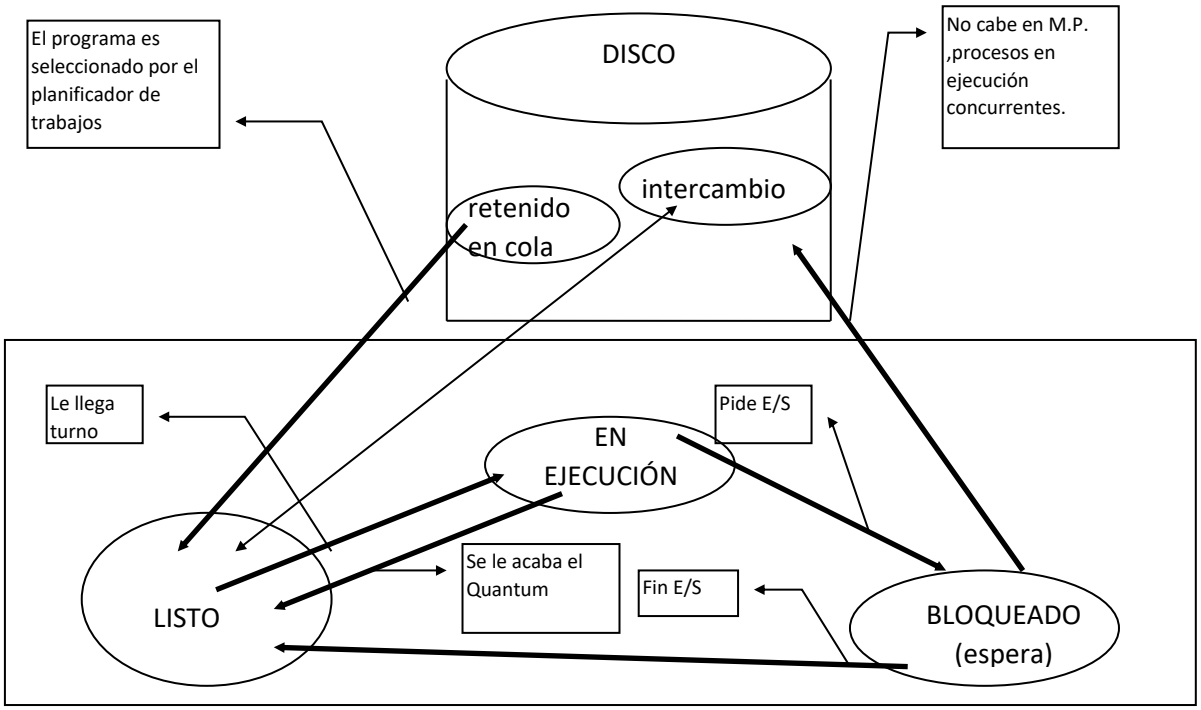
- El proceso saliente guarda todos sus valores asociados en el BCP.

- Se guardan los valores de los registros y direcciones de memoria implicados en el proceso en ese momento.
- Se cambia de estado el proceso.
- Se cambia de estado al proceso entrante.
- Se cargan los valores de los registros y direcciones de memoria asociados al proceso.

Todas estas operaciones son gestionadas y supervisadas por el procesador, lo cual quiere decir que en el tiempo que dura el cambio de contexto no se puede ejecutar ningún otro proceso.

Cuando el cambio de contexto se hace sin salir de memoria principal, el tiempo y los recursos empleados no son excesivos; pero en los cambios de contexto entre memoria principal y secundaria la situación cambia, siendo más lenta y costosa.

La técnica de cambiar de contexto es útil cuando se quiere dar tiempo de ejecución a todos los procesos, pero un uso excesivo del cambio de contexto puede ser contraproducente ya que el procesador estaría más tiempo ocupándose de las acciones asociadas a los cambios de contexto que a la ejecución de los procesos que se intercambian.



3.2.5.- Hilos de ejecución.

El manejo de los procesos, como se ha podido ver, es bastante complejo. Como alternativa a la visión general del proceso como ente básico surgió hace relativamente poco tiempo el concepto de hilo (thread).

Un hilo es la parte de un proceso que puede ser ejecutada de forma independiente.

De esta manera, un proceso estará constituido por al menos, un hilo, existiendo la posibilidad de que tenga varios.

La capacidad (del S.O) de mantener varios hilos de ejecución dentro del mismo proceso se conoce con el nombre de multihilo. Si no existe, entonces se habla de monohilo.

En general, todos los S.O modernos son multihilo.

Trabajar a nivel de hilos tiene grandes *ventajas* sobre hacerlo a nivel de procesos:

- Al tratarse de entidades mucho más ligeras, los tiempos empleados para su manejo (creación, terminación, cambio de estado y de contexto) son mucho menores respecto a los procesos.
- El tiempo para crear un hilo es mucho menor que para crear un proceso.

3.2.6.- Interrupciones y excepciones.

Durante el transcurso ordinario de la ejecución de procesos pueden darse dos situaciones especiales:

- **Interrupción.**

Se produce cuando se quiere que la CPU deje de ejecutar el proceso en curso y pase a realizar otra función de quien hace la interrupción. Cuando la CPU realiza esa función se dice que está atendiendo la interrupción.

Las interrupciones pueden darse a dos niveles:

  - **Nivel de software**: El usuario realiza una llamada al sistema (para hacer uso de un recurso del núcleo).
  - **Nivel de hardware**: Un dispositivo (hardware) requiere la atención de la CPU para ejecutar su driver.

Cuando se produce una interrupción se pasa el control al sistema operativo, quien salva el contexto del proceso que se estaba ejecutando y se analiza la interrupción. Las interrupciones están catalogadas y el sistema operativo dispone de rutinas especiales para manipular cada tipo de interrupción. Una vez se ha atendido la interrupción la CPU continúa con su anterior tarea.

- **Excepción.**

La excepción es un tipo de interrupción provocada por la propia CPU a causa de un error en la ejecución del proceso en activo como puede ser la realización de operaciones no permitidas, códigos de operación mal expresados, direcciones de memoria fuera de rango, etc.

El tratamiento de una excepción es similar al de la interrupción, con la salvedad de que las excepciones, a menudo, no continúan el proceso con fallo, sino que lo abortan.

3.2.7.- Demonios.

Existe un tipo muy particular de proceso que recibe el nombre de **demonio**.

Un demonio es un proceso que se ejecuta en segundo plano sin necesidad alguna de interacción con el usuario para llevar a cabo su tarea.

En sistemas Windows los demonios se les conocen más comúnmente como servicios, pues su finalidad es ofrecer un servicio al usuario.

Los demonios no disponen de interfaz, ni gráfica ni textual, ya que no necesitan comunicarse con el usuario. Tampoco hacen uso de los dispositivos de E/S comunes para notificar resultados o errores. En su lugar, vuelcan toda esta información en un archivo LOG o la comunican a otros demonios encargados de recopilar este tipo de datos.

Los demonios pueden iniciarse, detenerse y reiniciarse:

- Iniciar un demonio: `# <ruta del demonio> start`
- Parar un demonio: `# <ruta del demonio> stop`
- Reiniciar un demonio: `# <ruta del demonio> restart`

Los servicios (demonios Windows) permiten, además, ser pausados y reanudados.

3.2.8.- Inicio del sistema.

La puesta en marcha del S.O es una secuencia de operaciones que se varía de unos S.Operativos a otros. Todos tienen en común la ejecución de una serie de procesos que constituirán la base del sistema. Estos procesos posteriormente residirán en memoria y generarán nuevos subprocesos para ofrecer otras funcionalidades.

De entre todos los sistemas operativos disponibles, vamos a estudiar cómo es el proceso de inicio para sistemas Windows y para sistemas Linux.

Es conveniente resaltar que el proceso de inicio del S.O, incluso tratándose del mismo tipo de sistema operativo, puede variar de unas versiones a otras e incluso es dependiente del tipo de procesador para el que se instala (x86 o x64).

La secuencia de inicio se produce primordialmente en segundo plano, constando de cinco fases:

a) Secuencia previa al inicio.

Comienza cuando se enciende el equipo, antes de que se inicie el sistema operativo, y en ella tiene lugar el POST y el paso a la carga y ejecución del MBR.

b) Secuencia de inicio.

Se ejecuta el gestor de arranque y se seleccionará el S.O (en el caso de que hubiera más de un S.O instalado en el equipo) para iniciar. Posteriormente se hará una detección del hardware instalado.

c) Secuencia de carga del núcleo.

Se invoca al archivo WINLOAD.EXE para que cargue el archivo NTOSKRNL.EXE, que está considerado como el núcleo del S.O. También se carga el archivo HAL.DLL, capa intermedia entre el núcleo de Windows y el hardware.

Tras la carga de HAL se realiza la carga de la clave del registro HKEY_LOCAL_MACHIME\SYSTEM. El contenedor SELECT determina qué controlador debe cargar y el contenedor CONTROLSET proporciona los datos de configuración utilizados para controlar el sistema (controladores, servicios,...).

d) Secuencia de inicio del núcleo.

Coincide con la aparición del logotipo de Windows. Realmente, durante este tiempo se llevan a cabo varias tareas entre las que destacamos la creación de la clave HARDWARE en el registro con los datos recogidos en la fase de detección de hardware y la inicialización de los servicios y de los controladores de dispositivos.

Esta secuencia finaliza con la inicialización de los subsistemas y servicios de orden superior y el lanzamiento del subsistema WoW (en sistemas de 32 bits el subsistema es Win32), encargado de habilitar, entre otros, el interfaz de usuario, que comienza con el proceso WINLOGON.

e) Secuencia de inicio de sesión.

WINLOGON inicia la autoridad de seguridad local (LSASS.EXE), momento en el que aparece la pantalla de identificación de usuario (según esté configurada).

3.3.- Controlar y gestionar la memoria

El administrador o gestor de memoria es el módulo del S.O encargado controlar el espacio en memoria para poder alojar los procesos, también de liberarla cuando hayan finalizado, controla el intercambio de datos entre los dispositivos y de la protección de los datos almacenados. Dispone de cualidades como la capacidad de almacenamiento de (datos y programas), la velocidad de transmisión de datos unida al tiempo que tarda en operaciones de lectura/escritura. Todo proceso necesita espacio de memoria para almacenar el código de instrucciones u órdenes que le forman, los datos que manipula y el espacio o pila para operar y trabajar.

Podemos realizar la siguiente clasificación de los **tipos** de memoria:

- Según su **función**:
  - Memoria **interna**: que podemos clasificar en:

    - Memoria **principal o central**: se encarga de almacenar los programas y los datos que ejecutará el ordenador. Dispone de una gran velocidad de acceso, pero con poca capacidad de almacenamiento.
    - Memoria **caché**: proporciona una gran velocidad de acceso para acelerar el rendimiento del sistema. Hay que tener en cuenta que la velocidad de acceso de la memoria principal es muy inferior a la velocidad de operación del microprocesador, produciendo una ralentización en la ejecución de los procesos ya que el microprocesador tiene que esperar a que le llegue la información a tratar. Para paliar este defecto existe la memoria caché.
    - Memoria **de registros**: pequeñas direcciones de memoria temporales que guardan los datos en el momento en el que son objeto de procesamiento. Son muchos más rápidos que la caché, pero disponen de una mínima capacidad de almacenamiento.
  - Memoria **externa** o **secundaria**: es aquella que se emplea como almacenamiento pasivo en un dispositivo periférico como un disco duro, CD, etc.
- Según su **posibilidad de acceso**:
  - **RAM** (Random Access Memory): memoria de acceso aleatorio. Es volátil cuando se interrumpe la alimentación, la RAM pierde su contenido. Según su funcionamiento se distinguen dos tipos:
    - SRAM o RAM Estática: no pierde su contenido mientras recibe alimentación eléctrica. Esta memoria es muy rápida pero su fabricación es más costosa que las otras. Las memorias caché, de pequeño tamaño y de acceso muy rápido, están formadas por este tipo de RAM.
    - DRAM o RAM Dinámica: que pierde el contenido con el tiempo, aunque no se interrumpa el suministro de energía. Para evitar pérdidas de datos es necesario reescribir su contenido continuamente: es lo que se llama refresco de la memoria. Este tipo de memoria tiene un rendimiento menor que la SRAM pero su precio también es menor. Los módulos principales de memoria que se conectan en los zócalos (slots) de la placa base son de este tipo.
  - **ROM** (Read Only Memory): memoria de sólo lectura ya que podemos leer su contenido, pero no escribirlo. La información que contiene la ROM se escribe en el momento de su fabricación y, a partir de entonces, ya no puede cambiarse.

3.3.1.- Técnicas de administración de la memoria

Existen diferentes técnicas de administración o de gestión de memoria como son:

- **Memoria Virtual.** Para que un programa pueda ejecutarse, tiene que estar cargado en memoria real. La necesidad cada vez mayor de ejecutar programas grandes y la aparición de CPU's más potentes empujaron a los diseñadores de S.O.'s a implantar un mecanismo para ejecutar automáticamente programas más grandes que la memoria real disponible, esto es, de ofrecer memoria virtual. El sistema operativo deja en memoria principal RAM las partes del programa que se están utilizando (instrucciones y datos) y el resto lo almacena en disco mediante una zona de intercambio o archivo de intercambio, es decir, como un programa que se ubica en memoria

  puede ser excesivamente grande para el tamaño físico de ésta, permanece en memoria la parte del programa que se está ejecutando, mientras el resto está en el disco. Supongamos

que un ordenador necesita 80 MB de memoria para ejecutar programas, y sólo tenemos 64 MB. Para ello el sistema memoria virtual emplea el gestor de memoria virtual, que crea un archivo en el disco duro a modo de memoria adicional para suplir a la que falta. A este archivo se le denomina archivo de intercambio (swap file). Cuando el S.O. requiere hacer uso de una página de memoria que no está en memoria principal, la toma del archivo de intercambio. El S.O., en su módulo de administrador de memoria, se encarga de intercambiar programas enteros, segmentos o páginas entre la memoria real y el medio de almacenamiento secundario. Si lo que intercambia son procesos enteros, se habla entonces de **multiprogramación en memoria real**, pero si lo que se intercambian son segmentos o páginas, se puede hablar de m**ultiprogramación con memoria virtual**.

- **Swapping.** Es una técnica similar a la memoria virtual. Cuando varios usuarios están ejecutando procesos en un mismo ordenador, éste los carga en la RAM. Según el estado en el que este el proceso de cada usuario, la memoria se ira liberando o no. Si un usuario interrumpe por un instante la ejecución de un proceso, pasara a la zona de SWAP mediante la técnica llamada swap-out, liberándose la memoria interna para que pueda alojarse otro proceso del mismo u usuario o de otro. Si el usurario vuelve a solicitar su proceso para seguir ejecutándolo, se produce el swap-in, que consiste en pasar el programa de la zona de swap a la memoria interna. Esta zona de swap se suele utilizar en sistemas operativos como UNIX, LINUX y OS . Está formada por un espacio físico del disco en el que tenemos el sistema operativo y las aplicaciones que se van a ejecutar. Los fabricantes recomiendan que esta zona sea del 20% . aproximadamente, del espacio en disco o el doble de la capacidad de memoria RAM del ordenador.

  > La diferencia entre la gestión de memoria virtual y el swapping es que, mediante la primera , puede llegar a ocurrir que el disco esté tan lleno que la gestión sea difícil o imposible, ya que el espacio destinado al intercambio suele ser espacio del disco duro en el que está instalado tanto el sistema operativo como el software de aplicaciones y los datos del usuario, en el swapping no puede ocurrir esto, ya que esta zona siempre estará disponible para el intercambio de programas con la memoria principal.

- **Paginación.** Método que consiste en dividir la memoria física en zonas de tamaño fijo llamadas frames o tramas y los programas o espacio lógico en partes del mismo tamaño llamadas **páginas**. Cuando varios usuarios están ejecutando procesos en un mismo ordenador, éste se ve obligado a cargarlos en RAM, según el estado en el que se encuentre el proceso de cada usuario, la memoria se irá liberando o no. La transformación de las direcciones lógicas en físicas la realiza la unidad de administración de memoria o Management Memory Unite (MMU). El sistema operativo MS-DOS utiliza una técnica parecida a la paginación.
- **Segmentación.** Técnica similar a la paginación pero definiendo los bloques de memoria de tamaño variable. La información lógica del proceso se divide en distintos bloques lógicos denominados segmentos, donde cada segmento tiene información lógica del programa (datos y código) y de pila (stack). La principal ventaja de la segmentación es que, como de cada segmento sabemos su tamaño, podemos controlar mejor los errores.

En muchas ocasiones es necesario conocer las diferentes unidades de **medida de la información o datos en informática**, ya que es un dato que aporta información al sistema. La unidad más pequeña de información en un ordenador corresponde a un dígito binario, es decir, 0 o 1. A este dígito se le denomina **bit**, abreviatura de la palabra inglesa Binary

Digit. Al conjunto de 8 bits se le denomina byte, por lo tanto, cada carácter está representado por un **byte**.

Estas unidades de medida resultan muy pequeñas, por lo que se necesitan algunos múltiplos del byte. Así hablamos de kilobyte, Megabyte, Gigabyte, etc. La proporción entre las distintas magnitudes es 1024 porque esta cantidad es la potencia de base 2 que más se aproxima a la proporción 1000, equivalente en el sistema métrico decimal al prefijo kilo ($2^{10} = 1024$).

Unidades	Equivalencias	Equivalencias en bytes
1 Kilobyte (Kb)	1024 bytes	2^{10} bytes
1 Megabyte (Mb)	1024 Kilobytes	$2^{10}.2^{10}$ bytes = 2^{20} bytes
1 Gigabyte (Gb)	1024 Megabytes	$2^{10}.2^{10}.2^{10}$ bytes = 2^{30} bytes
1 Terabyte (Tb)	1024 Gigabytes	$2^{10}.2^{10}.2^{10}.2^{10}$ bytes = 2^{40} bytes
1 PetaByte (Pb)	1024 *Terabyte*	
1 ExaByte (Eb)	1024 *PetaByte*	
1 ZettaByte (Zb)	1024 *ExaByte*	
1 YottaByte (Yb)	1024 *ZettaByte (Zb)*	

3.4.- Controlar los dispositivos periféricos. Clasificación de periféricos

Los periféricos de entrada/salida son dispositivos hardware que junto con los soportes se encargan almacenar, leer datos y programas que serán procesados por el sistema. Una de las funciones principales de un S.O es el control de estos periféricos enviando órdenes para determinar que dispositivo necesita la atención del procesador con el fin de gestionar la tarea de entrada/salida de la información. Para conectar los dispositivos periféricos al ordenador, se utilizan **conectores** denominados slots y puertos.

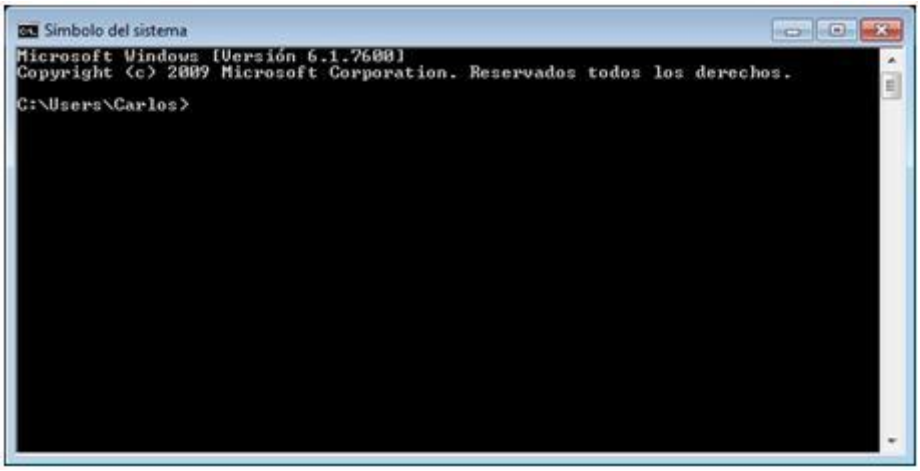
Cuando se realiza el acceso a un dispositivo se hace a través de su parte electrónica llamada **controladora física** de dispositivo y mediante el software denominado **driver** o **controlador lógico** que es el encargado de traducir las órdenes dadas por el S.O al dispositivos, es decir, es el encargado de indicar los comandos que tiene que ejecutar y verificar que se ejecuten de forma adecuada. Estos drivers vienen diseñados para varios S.O; así, el mismo periférico lo podremos utilizar en un S.O Windows o en un sistema UNIX, dependiendo del driver que instalemos.

Los dispositivos físicos son los encargados de manejar los soportes de almacenamiento mediante los interfaces que permiten la comunicación entre el usuario y el S.O. Otros elementos necesarios para la comunicación son los buses (autopistas de la información) o canales encargados de transmitir la información entre los diferentes componentes que

integran el ordenador. Para gestionar los dispositivos se necesitan dos valores que lo identifique denominado **interrupción** y de una dirección de acceso directo a memoria (**DMA**).

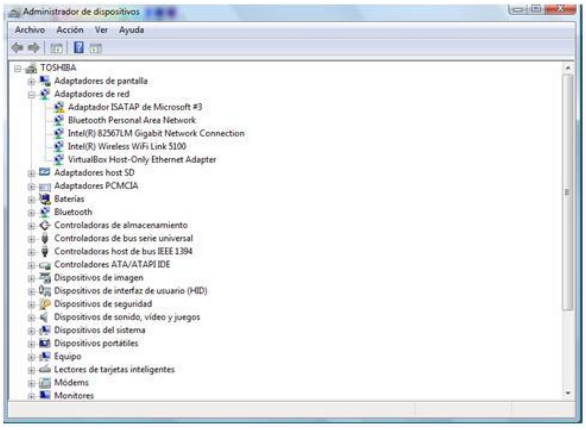
Para facilitar la comunicación entre el usuario y los dispositivos, el S.O aporta los denominados **interfaces de comunicación** que pueden ser:

- **Interfaz tipo texto**. Todas las órdenes que el usuario introduzca y las respuestas que el S.O dé se visualizarán mediante cadenas de caracteres.



Pantalla Windows 7. Elaboración propia

- **Interfaz tipo gráfico**. La información en pantalla se muestra en ventanas, y en ellas aparecen una serie de componentes y objetos que sirven para enviar o recibir información sin tener que teclear nada.



Los periféricos se pueden clasificar según su función de su uso:

- **De entrada.** Son los que sirven para introducir información (datos o programas) en el ordenador. La información va desde ellos hacia la memoria y el resto de componentes internos, para ser procesada. Son periféricos de entrada el teclado, un escáner, la unidad lectora de CD-ROM, el ratón, etcétera.

- **De salida**. Son los que se utilizan para extraer la información (datos en forma de resultados, programas, etc.) desde la memoria y el resto de componentes internos del ordenador y mostrar los datos. La impresora, la pantalla, el plotter, etc., son periféricos de salida.
- **De entrada/salida (E/S).** Son los que se utilizan para introducir o extraer datos desde y hacia el ordenador, como, por ejemplo, los dispositivos de almacenamiento (discos duros). En ellos se puede escribir información (salida) al igual que leerla (entrada). Hay otros muchos periféricos dentro de esta categoría, como los monitores táctiles, módems, routers, tarjetas de red, disqueteras, impresoras multifunción, etcétera.

3.5.- Controlar la organización de ficheros o archivos

Los ficheros son la estructura utilizada para alojar datos o instrucciones que se almacenan en soportes externos para poder ser procesada por el sistema mediante un determinado programa. El S.O. utiliza el sistema de ficheros para manejar, organizar y almacenar los ficheros de forma permanente en soportes externos.

Los sistemas de ficheros manejan dos tipos fundamentales de objetos:

- Los **ficheros regulares** (file): es una unidad lógica de memoria para almacenar datos que se identifica por un **nombre**. Las características de los nombres de los ficheros dependen de los sistemas operativos, por ejemplo la **extensión** indica el tipo de fichero que es, el **atributo** que caracteriza a cada fichero indicando que tipo de operaciones o usuarios pueden interactuar con él, etc.
- Los **directorios** (directory): son contenedores o carpetas que sirve para almacenar archivos u otros directorios. La utilización de directorios permite una mayor organización de los ficheros dentro del disco. En casi todos los sistemas de fichero existe un directorio principal llamado raíz (root) que es el directorio que contiene todos los demás ficheros y directorios. A partir de él se crea una estructura jerárquica en forma de árbol invertido de ficheros y directorios. Los directorios también disponen de atributos indicando que tipo de operaciones o usuarios pueden interactuar con él.

  La estructura de directorios suele ser jerárquica, ramificada o "en árbol". En los sistemas de archivos jerárquicos, usualmente, se declara la ubicación precisa de un archivo con una cadena de texto llamada "**ruta**"o path. La nomenclatura para rutas varía ligeramente de sistema en sistema, pero mantienen por lo general una misma estructura. Una ruta viene dada por una sucesión de nombres de directorios y subdirectorios, ordenados jerárquicamente de izquierda a derecha y separados por algún carácter especial que suele ser una barra ('/') o barra invertida ('\') y puede terminar en el nombre de un archivo presente en la última rama de directorios especificada.

  Así, por ejemplo:

- en un sistema de archivos de Windows se vería como:

  C:\Documents and Settings\carlos\Mis Documentos\foto.png

  Las principales **operaciones que se suelen realizar con los ficheros** en la mayoría de los sistemas son: crear, renombrar, abrir, copiar, buscar, leer, escribir, cerrar y borrar, las cuales van relacionadas con los permisos y derechos que tiene cada usuario para su

uso. Las operaciones con los directorios, crear, borrar, abrir, cerrar, leer, cambiar de nombre.

Para crear un sistema de ficheros es necesario realizar la operación denominada particionar el disco. Una partición de disco es el nombre genérico que recibe cada división presente en una sola unidad física de almacenamiento de datos. Toda partición tiene su propio sistema de archivos o formato. Una sola partición primaria o unidad lógica puede usar sólo un sistema de archivos. Un disco físico puede tener varias particiones y por lo tanto tener instalado varios sistemas operativos

4.- Tipos de Sistemas Operativos

Según la perspectiva con la que se observen los sistemas operativos, pueden realizarse múltiples clasificaciones.

Entre ellas se pueden incluir las siguientes:

4.1. Por Los Servicios Ofrecidos.

En esta clasificación se tiene en cuenta la visión del usuario final y puede ser:

- **Por el número de usuarios:**
  - **Monousuario:** Únicamente soporta un usuario a la vez, sin importar las características de la máquina sobre la que está montado el sistema.
  - **Multiusuario:** Son capaces de dar servicio a más de un usuario a la vez, también independientemente de la plataforma hw sobre la que está montado.
- **Por el número de tareas:**
  - **Monotarea:** Son aquéllos 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 sólo una tarea al mismo tiempo.
  - **Multitarea:** Son aquellos que le permite al usuario estar realizando varios trabajos al mismo tiempo. 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.
- **Por el número de procesadores:**
  - **Monoproceso:** Son los que únicamente permiten utilizar un procesador. Sin embargo, permiten simular la multitarea haciendo que el sistema realice una tarea rotatoria con intercambio muy rápido.
  - **Multiproceso:** Son los que permiten utilizar varios procesadores simultáneamente y, por tanto, son capaces de ejecutar varias tareas al mismo tiempo.

4.2. Por La Forma De Ofrecer Los Servicios.

- **Sistemas Centralizados:** Con este tipo de modelo los computadores ***mainframe*** se encargaban de todo el procesamiento y los **usuarios** manejaban únicamente terminales "tontos" (es decir, no disponían de memoria, ni de procesador). Los principales SS.OO. centralizados en el mercado son z/OS, OS/390, Linux, TPF, VSE y ESA.

- **Sistemas en Red:** Estos sistemas operativos son aquellos sistemas que mantienen a dos o más computadoras unidas a través de algún medio de comunicación (físico o no), con el objetivo primordial de poder compartir los diferentes recursos y la información del sistema. En este entorno, cada computador mantiene su propio sistema operativo y su propio sistema de archivos local.

El primer sistema operativo de red estaba enfocado a equipos con un procesador Motorola 68000, pasando posteriormente a procesadores Intel como Novell NetWare.

Los sistemas operativos de red usados más ampliamente son: Novell NetWare, Personal NetWare, LAN Manager, Windows NT Server, Windows 2000 Server UNIX, LANtastic, etc.

- **Sistemas Distribuidos:** Los sistemas operativos distribuidos son sistemas cuasi independientes que permiten distribuir los trabajos, tareas o procesos entro un conjunto de procesadores. Puede ocurrir que este conjunto de procesadores se encuentre en el mismo equipo o en equipos distintos (siendo, en este último caso, transparente para el usuario). Los sistemas operativos distribuidos más extendidos son los siguientes: *Sprite, Solaris-MC, Mach, Chorus, Spring, -amoeba, Taos,* etc.

- **Sistemas operatorios de escritorio:** Estos sistemas operativos son los que se utilizan en los equipos de sobremesa, estaciones de trabajo o portátiles . También se les puede denominar como sistemas operativos cliente . Entre ellos se encuentran: *Windows XP Professional, Windows Vista* , *Windows* 7 y *Linux.*

4.3. Por Su Disponibilidad.

- **Sistemas operativos propietarios:** son aquellos que son propiedad intelectual de alguna empresa. Esto implica que se necesitan licencias de uso para que el usuario ejecute el software y no se dispone a acceso a su código fuente o, aun teniendo acceso a él, no se tiene derecho a modificarlo ni distribuirlo. En este grupo se encuentra Windows

- **Sistemas operativos libres:** son aquellos que garantizan las cuatro libertades del software (según Richard M. Stallman):

1) La libertad de usar el programa con cualquier propósito.

2) La libertad de estudiar cómo funciona el programa y modificarlo, adaptándolo a las necesidades que tuviera el usuario.

3) La libertad de distribuir copias del programa, con lo que se puede ayudar a otros usuarios.

4) La libertad de mejorar el programa y hacer públicas dichas mejoras a otros usuarios, de modo que toda la comunidad se beneficie de ello.

Las libertades 2 y 4 requieren acceso al **código fuente** para estudiar y modificar dicho *software,* por lo que al final el ***software* libre** es también **software de código abierto**.

- El software libre suele estar disponible gratuitamente o al precio de coste de la distribución a través de otros medios; sin embargo no es obligatorio que sea así, por lo tanto, no hay que asociar software libre a software gratuito, ya que, conservando su carácter de libre, podrá ser distribuido comercialmente (software comercial).

- De la misma manera, el software gratuito puede incluir el código fuente, pero eso no quiere decir que se pueda considerar como libre a no ser que se garanticen los derechos de modificación y redistribución de las versiones modificadas del programa.

- Tampoco debe confundirse software libre con software de dominio público. Este último es aquel que no requiere de licencia pues sus derechos de explotación pertenecen a todos por igual y cualquiera puede hacer uso de él, siempre con fines legales y consignando su autoría original.

4.4. Por Su Estructura Interna.

- **Monolítica.** Es la estructura utilizada en los primeros sistemas operativos en la que todas las funciones se implementaban en el Kernel. Puede decirse que su estructura consiste en que no existe una estructura como tal.

El sistema operativo está constituido por un único programa compuesto de multitud de rutinas interrelacionadas entre sí, de forma que cada una de ellas pueda llamar a cualquier otra.

- **Por capas.** A medida que los sistemas operativos fueron creciendo, fue siendo necesaria una mayor estructuración.

Este diseño se corresponde con una estructura jerárquica que se divide en distintos niveles.

- **Maquina virtual.** Se trata de un tipo de sistemas operativos que presentan una interfaz a cada proceso, mostrando una máquina que parece idéntica a la máquina real subyacente.

4.5. Por Los Modos De Explotación.

Los modos de explotación se corresponden con las distintas maneras en que puede funcionar un sistema operativo.

- **Procesamiento por lotes.** Este modo de explotación se caracteriza por la agrupación en bloques de los trabajos similares. El rasgo más característico de este tipo de sistema operativo es la ausencia de interacción entre el usuario y el proceso mientras éste se ejecuta.

Los sistemas operativos por lotes más extendidos son *SCOPE, DC6600* (orientados al procesamiento científico), *EXECII UNIVAC1707* (orientados al procesamiento académico).

- **Multiprogramación**. En este modo de explotación, el sistema operativo se encarga de distribuir la carga computacional entre los procesadores existentes (monoprocesador o multiprocesador), con el fin de incrementar el poder de procesamiento de la máquina.

Dentro de los sistemas operativos multiprogramados cabe diferenciar:

- **Tiempo compartido.** Son los sistemas operativos más extendidos y a los que se refiere este capítulo. Utilizan las distintas técnicas de planificación de CPU para que se atiendan todos los procesos en espera de ser ejecutados. Este proceso ocurre tan rápidamente que el usuario no lo percibe.

Entre este tipo de sistemas operativos se encuentran: *UNIX, Windows 95, Windows 98, Windows Millenium, Windows XP, Windows NT, Windows 2000, MAC-OS y OS/2.*

- **Tiempo real.** Un sistema en tiempo real es aquél en el cual los resultados son correctos no sólo si la computación es correcta, sino que también ha de serlo el tiempo en el cual se producen los resultados.

Los sistemas operativos en tiempo real son sistemas muy complejos que suelen diseñarse a medida para ciertas aplicaciones, después de mucho tiempo de estudio de todas las opciones y problemas que pudieran surgir.

Entre ellos se encuentran: *Solaris, Spectra* y *VxWorks.*

- **Híbrido.** Estos sistemas operativos intentan ser una mezcla de los dos anteriores, buscando combinar las ventajas de los sistemas en tiempo compartido y en tiempo real. No se han obtenido aún sistemas realmente eficientes.

5.- Aplicaciones informáticas

La informática es el "conjunto de conocimientos científicos y técnicas que hacen posible el tratamiento automático de la información por medio de ordenadores ", es decir, se encarga del tratamiento de la información mediante el estudio de métodos, procesos, técnicas y desarrollos utilizando computadoras o ordenadores para conseguir almacenar, procesar y transmitir información y datos en formato digital. Para realizar esta tarea es necesario elaborar programas informáticos que contienen instrucciones u órdenes para que una computadora realice las tareas deseadas.

Las computadoras necesitan de los programas para funcionar, y un programa no hace nada a menos que sus instrucciones sean ejecutadas por el procesador del ordenador, es decir, cuando su código fuente es transformado en un ejecutable, cuando es compilado. De esta manera podemos decir que, en informática, una aplicación es un tipo de programa informático diseñado para facilitar al usuario la realización de un determinado tipo de trabajo. Al conjunto de programas o aplicaciones informáticas se le llama software informático o soporte lógico.

De acuerdo a sus funciones, los programas pueden ser clasificados:

Software de sistema o software base (como pueden ser cargadores de programas, sistemas operativos de estaciones de trabajo o de servidores, controladores de hardware, utilidades) encargado de proporcionar al usuario el control del sistema informático de una forma desatendida con herramientas interactivas para su correcto mantenimiento. Podemos incluir como caso especial el software de programación (como son compiladores, ensambladores, enlazadores, utilidades, etc.) que permiten desarrollar programas y aplicaciones informáticas utilizando diferentes herramientas y los lenguajes de programación.

- **Software de aplicación** o programas diseñados para facilitar al usuario la realización de un determinado tipo de trabajo. Algunos ejemplos de programas de aplicación son los llamados de **propósito general** en los que destacan los paquetes ofimáticos que integran o relacionan los procesadores de textos, hojas de cálculo, y base de datos o los de **propósito específico** destinados a resolver una determinada tarea en el mundo de la gestión como son los ERPs o Sistemas Integrados de Gestión (para la facturación, nóminas, control de almacén, contabilidad), los CRMs o Gestión Integral de Relación con los Clientes, Los Workflows o Sistemas de Gestión de Trabajo (encargados de la automatización de los procesos de una actividad de trabajo).

5.1.- Modelo de aplicación cliente-servidor: aplicaciones distribuidas

Debido al desarrollo de los sistemas informáticos hacia la forma de trabajo en modelos de **red distribuida** (topologia de red) basados en diferentes plataformas capaces de conectar ordenadores entre sí en los que el usuario accede a recursos remotos de la misma manera en que accede a recursos locales, y por el impulso en mejorar los procesos en la red de comunicación de área extensa (Wan) o Internet hacia tecnologías basadas en la llamada Web 3.0 hay que destacar la implantación del software orientado a la llamada **Aplicación distribuida** como un programa o conjunto de programas instalados en diferentes computadoras conectadas en red los cuales están relacionados o integrados entre sí para realizar una tarea o gestionar un proceso entre un ordenador cliente y uno servidor.

Los componentes que aparecen en estos entornos de trabajo con aplicaciones distribuidas son:

- El ordenador **cliente** inicia la comunicación (normalmente mediante un interfaz gráfico) con el servidor por medio de un protocolo de acceso para demandarle datos o para que realice tareas determinadas.
- El ordenador **servidor** dispone de las herramientas adecuadas para procesar las peticiones, incluso de varios clientes a la vez y enviar la respuesta adecuada.
- El middleware será el interfaz que provee la conectividad entre aplicaciones mediante una capa de software que protege a los desarrolladores del software de tener que manejar detalles de bajo nivel de diferentes protocolos de comunicación, sistemas operativos y otras arquitecturas como las de bases de datos.

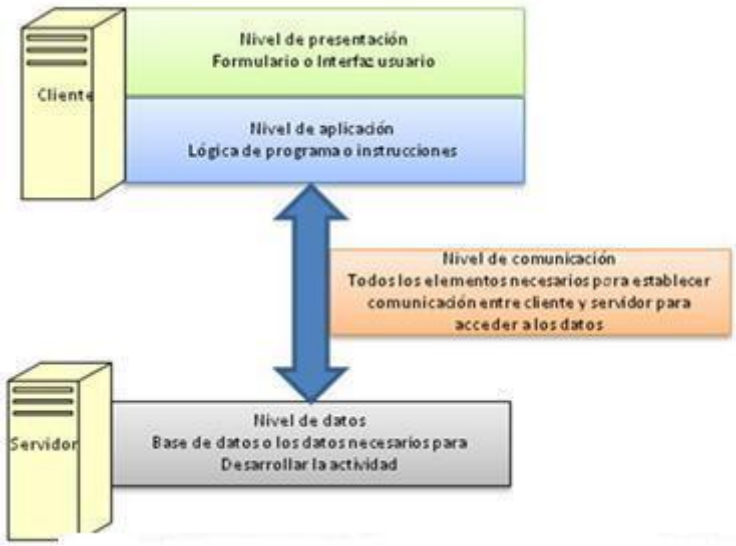
Podemos encontrar diferentes clasificaciónes en el modelo cliente-servidor:

- En función de la carga del proceso entre el cliente y el servidor:

- **Cliente pesado-servidor ligero** (fat client-thin server): el grueso de la aplicación se ejecuta en el cliente.
- **Servidor pesado-cliente ligero** (fat server-thin client): la mayor parte de la aplicación se ejecuta por el lado del servidor.

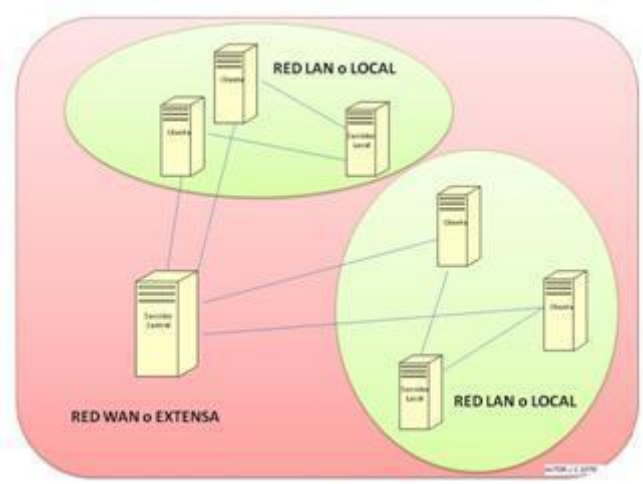
  - Por las funciones asignadas de las prestaciones (agravadas en interfaz de usuario, lógica de negociado y datos compartidas) que ofrece la aplicación:

- **De dos niveles:** son aplicaciones que permiten a ordenadores denominadas estaciones de trabajo solicitar servicios a otras computadoras llamadas servidores que contienen los datos permitiendo al cliente presentar el resultado del proceso que se realiza en el ordenador del cliente o del servidor o inclusive en ambos.



- **De tres niveles:** permite conectar múltiples aplicaciones para crear una aplicación más grande ofreciendo un conjunto de servicios que permite el funcionamiento de aplicaciones sobre plataformas heterogéneas. Es el caso típico en el que se dispone de varios servidores los cuales se encargan de realizar diferentes servicios para gestionar la aplicación, los resultados se presentan en el cliente después de acceder al servidor que ejecuta la aplicación el cual accede a los datos que se encuentran en otro servidor.

- **Multinivel**: el procesamiento se puede dividir en un sistema multicapa permitiendo dividir las tareas complejas de la aplicación en tareas más sencillas entre varios servidores.



Elaboración propia utilizando la galería *openclipart-0.18-full*. Procedencia

- Por el servicio ofrecido por los servidores:

- **Servidores de bases de datos**: servidores que gestiona peticiones realizadas por clientes mediante el lenguaje de consulta (SQL).
- **Servidores de transacciones**: el proceso cliente llama a funciones que residen en el servidor de manera que el intercambio a través de la red se realiza en un único acceso de solicitud y respuesta independiente mente de la aplicación.
- **Servidor web**: peticiones realizadas mediante el protocolo de comunicación HTTP.
- **Servidores de archivos**: permite el acceso remoto a archivos almacenados en un ordenador servidor. Los protocolos que suele utilizar son SMB, NFS.

6.- Licencias y tipos de licencias

Una **licencia de software** es una autorización mediante contrato (aceptación de condiciones legales normalmente en el proceso de instalación) para poder utilizar aplicaciones informáticas de una forma determinada; es decir, la licencia es un documento que expresa la voluntad del autor sobre los límites y alcances del uso que pueden hacer las personas respecto a la copia, reproducción, modificación, traducción y adaptación.

Cuando se realiza una aplicación y se quiere adjuntar un contrato de licencia para su descripción debemos de tener en cuenta aspectos como si se desea ofrecer el código fuente, si se permite su modificación, si se puede redistribuir o no, las instalaciones que se permiten, etc. Seguidamente podemos registrar los **Derechos de Autor** o **Copyright**, así como la patente.

Según los criterios en que se formulen en el **contrato de uso** (es conveniente leer antes de aceptar sus condiciones), podemos encontrar diferentes **modos de clasificación**, y una de ellas podría ser la que utilizan muchos portales de Internet que distribuyen software que lo suelen identificar con alguno de estos tipos según su manera uso y diseño:

- **Software Libre:** puede ser utilizado, copiado, distribuirlo y modificado (cuando el código fuente disponible) para mejorar el programa o adaptarlo a las necesidades. Normalmente llevan clausulas en el contrato para que su uso no sea con fines comerciales. También puede haber programas libres que no pueden ser modificados ni redistribuidos pero si instalados para uso exclusivo. Una variante destacable es el llamado software con

licencia **Open Source iniciative** que detalla claramente la libertad a los usuarios para leer, modificar y redistribuir el código fuente de un programa; los usuarios lo adaptan a sus necesidades, corrigen sus errores a una velocidad impresionante, mayor a la aplicada en el desarrollo de software convencional o cerrado, dando como resultado la producción de un mejor software.

Logotipo de Open Source.

- **Software propietario o privado:** es aquel que sin permiso del propietario queda prohibida la copia, redistribución o modificación. Para poder usar se suele pedir permiso a la organización que lo desarrollo. Generalmente para su disponibilidad hay que pagar bajo unos derechos de autor (**un Copyryght**). En conclusión, los propietarios son los que establecen los derechos de uso, distribución, redistribución, copia, modificación, cesión y en general cualquier otra consideración que se estime necesaria. Los fabricantes de programas sometidos a este tipo de licencias por lo general ofrecen servicios de soporte técnico y actualizaciones durante el tiempo de vida del producto, también regulan el número de copias que pueden ser instaladas e incluso los fines concretos para los cuales puede ser utilizado.
- **Software comercial:** para su disponibilidad hay que realizar un pago. Puede existir software libre y propietario de este tipo.
- **Software de dominio público.** El Software con dominio público es software **sin copyright**. Se permite uso, copia, modificación o redistribución con o sin fines de lucro.
- **Freeware:** programas que permiten la redistribución, pero no la modificación, y que a veces incluyen su código fuente. Estos programas no son softwares completamente libres de uso.
- **Shareware:** es el software disponible con permiso para ser redistribuido, pero su uso está limitado en tiempo o en funciones (no contienen todos los procesos). Para tener una disponibilidad completa hay que realizar un pago. Generalmente, el código fuente no se encuentra disponible.
- **GPL:** se la puede considerar como Licencia de software libre con protección heredada. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios, impidiendo que este software sea integrado en software propietario. Es la licencia que acompaña una gran variedad de software que incluye el núcleo del sistema operativo Linux. Una de las más destacada es *Licencia Pública General de GNU* (GNU GPL) en la que autor conserva los derechos de autor (copyright), y permite la redistribución y modificación bajo términos diseñados para asegurarse de que todas las versiones modificadas del software permanecen bajo los términos más restrictivos de la propia licencia.
- **Con Copyleft:** es aquel software que dispone de un tipo de copyrihgt creado para el software libre que no permite agregar normas de uso de las aparecen en la licencia determinada por el autor original y en las que detalla las condiciones bajo las cuales garantiza las libertades de uso (no disponibles en el contrato original del copyright proporcionado por las leyes vigentes de un país).

  Hay dos tipos de licencias copyleft:

  - **Completa**: permite cualquier modificación de la obra inicial a excepción de la licencia.
  - **Parcial**: limita las partes de la obra que son modificables.

    - **DFSG:** es parte del contrato realizado entre Debian (http://www.debian.org/social_contract.es.html) y la comunidad de usuarios de software libre. La licencia de Open Source Initiative deriva de Debian.

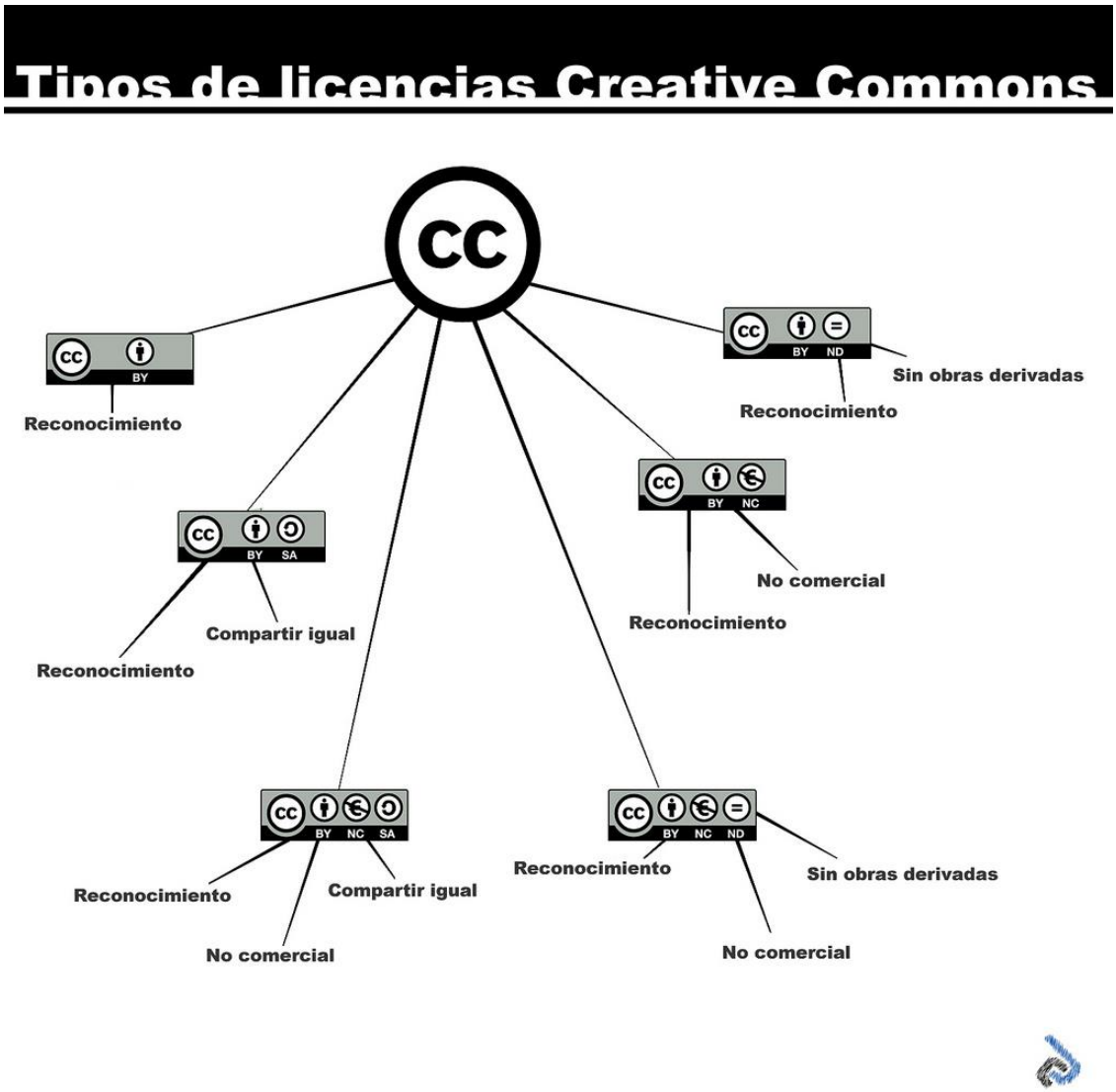
- **BSD. Licencia de software libre sin protección heredada.** Se puede crear una obra derivada sin que ésta tenga obligación de protección alguna. Puede argumentarse que esta licencia asegura "verdadero" software libre, en el sentido que el usuario tiene libertad ilimitada con respecto al software, y que puede decidir incluso redistribuirlo como no libre (ser vendido) y no hay obligaciones de incluir el código fuente. Esta licencia garantiza el crédito a los autores del software pero no intenta garantizar que las modificaciones futuras permanezcan siendo software libre.
- **Licencias estilo MPL:** es Software Libre y promueve eficazmente la colaboración evitando el efecto "viral" de la GPL (si usas código licenciado GPL, tu desarrollo final tiene que estar licenciado GPL).

- Las licencias **Creative Commons** están basadas en la GPL, y, aunque no suelen ser empleadas para licenciar software, si suelen ser utilizadas para **distribuir y usar contenidos**. Suelen responder al lema "algunos derechos reservados" por lo que un autor que decida publicar su trabajo mediante una licencia CC debe escoger aquellos derechos que está dispuesto a ceder, de entre los que posee.

Es importante señalar que en todos los casos de **licencias CC** se presupone que el autor otorga los derechos de copia y distribución de la obra. Partiendo de esa base, se toman en consideración los siguientes derechos que el autor puede elegir reservarse o no:

- **Reconocimiento (Attribution)**: se permite la copia, distribución y presentación pública de la obra y trabajos derivados de la misma siempre que se reconozca y cite adecuadamente al autor original.
- **No comercial (Non-Commercial)**: se permite la copia, distribución y presentación de la obra y trabajos derivados de la misma siempre que se realice con fines no comerciales.
- **Prohibición de obras derivadas (No Derivatives)**: se permite la copia, distribución y presentación de la obra en su versión original, pero se prohíbe la realización de trabajos derivados de la misma.
- **Redistribución bajo la misma licencia (Share Alike)**: se permite la distribución de trabajos derivados de la obra siempre que se realice bajo una licencia idéntica a la que ampara la obra original.

Debemos ser conscientes y cuidadosos con las implicaciones legales (jurídicas) que se derivan del software que empleamos y de las consecuencias que conlleva no respetar las restricciones que marca esta. En el mejor de los casos solo nos veremos obligados a desinstalar completamente el software o tener que abonar la correspondiente licencia para seguir haciendo uso del mismo, pero se dan casos de multas por incumplimientos de licencia.



7.- Gestores de arranque

La gestión de arranque en ordenadores consiste en la manera de encendido y puesta en marcha de los Sistemas Operativos (S.O.) dependiendo del soporte donde se encuentran instalados: memorias USB, los Live CD, discos duros, etc. Se llama **encendido del ordenador** a los pasos seguidos por el computador hasta llegar al punto de carga del S.O. y pueden ser los siguientes:

1. Cuando se enciende el ordenador (botón power) y llega corriente a los componentes de la placa base el microprocesador resetea e inicia todos sus contadores y registros. Busca una dirección de la **ROM-BIOS** del sistema y ejecuta la **BIOS** (Basic Input/Output System).
2. Seguidamente comienza el proceso conocido como **POST** (Power On Self Test), en el que se comprueba el correcto funcionamiento de los componentes instalados (normalmente en caso afirmativo emite un pitido, en caso de avería de algún componente importante emitirán más), además, la BIOS está formada por un conjunto de programas que se encarga de la configuración de la **CMOS** la cual controla y supervisa los dispositivos conectados al ordenador (integrados o no a la placa base) y otras preferencias mediante valores otorgados a unos parámetros, estos programas se encuentran grabados una

memoria de tipo **flash ROM** que permite que las rutinas grabadas puedan ser actualizadas para mejorar la adaptación de los componentes conectados al PC .

3. La BIOS asignará direcciones de acceso directo (**DMA**) y de de interrupción (IRQ) a los dispositivos, activará los dispositivos Plug & Play, inicia la BIOS de la tarjeta de vídeo (es en ese momento aparecen los mensajes en la pantalla en los que se ven el resultado del testeo y la cantidad de la memoria RAM), habilita el teclado comprobando su correcto funcionamiento posibilitando mediante una combinación de teclas la entrada a configurar parte de los parámetros de la BIOS (conocido como **Setup**) como son: fecha, hora, secuencia de arranque, etc. AL final la BIOS comprueba la secuencia de arranque de los dispositivos que almacenan el o los Sistemas Operativos en el sistema; localiza el **MBR** (Master Boot Record los primeros 512 bytes del disco duro), del disco a arrancar y comienza con el proceso denominado **bootstrap** o carga del Sistema.
4. El MBR es el primer sector del disco duro que contiene la tabla de particiones y de un programa llamado Master Boot que se encarga de leer la tabla de particiones (divisiones de un disco que pueden ser como máximo tres primarias y una extendida que a su vez se puede dividir en lógicas) y de ceder el control al sector de arranque de la partición que está marcada como activa (que almacena el sistema operativo con el que arrancará el ordenador). Si se dispone del llamado bootstrap loader en los primeros 446 bytes del MBR podremos seleccionar el S.O. con el que deseamos arrancar (en caso de haber más de uno en diferentes particiones) o de arrancar el S.O. alojado en la partición que está marcada como activa, en ambos casos cederá el control al sector inicial de dicha parición y se cargará el sistema.

   En resumen y **conclusión** podemos considerar que cuando encendemos el ordenador, la corriente eléctrica da vida a los componentes de la placa base. Inmediatamente que el microprocesador envía una orden al chip de la memoria ROM del BIOS (Basic Input/Output System - Sistema básico de entrada/salida), donde se encuentran grabadas las rutinas del POST (Power-On Self-Test - Autocomprobación diagnóstica de encendido) o programa de arranque. Una vez que el BIOS recibe la orden del microprocesador, el POST comienza a ejecutar una secuencia de pruebas pasando el control al MBR se dirigirá al Master boot Record (sector de arranque del disco duro) para proseguir con el arranque del ordenador. Si hay cargador de arranque (boot loader) se ejecuta ofreciendo un menú de selección de arranque de sistema, en caso contrario el MBR analiza la tabla de particiones y se cargar en memoria el sector de arranque de la partición activa (en el que existirá un cargador encargado de ejecutar el S.O. instalado en la misma o de de mostrar un menú de selección (un boot loader).

7.1.- Conceptos relacionados con el arranque de sistemas operativos

Algunas consideraciones y **conceptos importantes relacionados con el arranque de sistemas** son.

- **La BIOS:** el Sistema Básico de Entrada/Salida o ***BIOS*** (*Basic Input-Output System)* es un código de software que localiza y reconoce todos los dispositivos necesarios para cargar el sistema operativo en la RAM; es un software muy básico instalado en la placa base que permite que ésta cumpla su cometido. Su función primordial es la de encontrar el S.O y cargarlo en memoria RAM. EL programa que controla la BIOS reside en la memoria EPROM (Ver Memoria BIOS no-volátil). Es un programa tipo firmware (se puede actualizar) que permite la configuración de aspectos importantísimos de la máquina.

- **El BOOTLOADER:** un **bootloader** (cargador de arranque) es un programa sencillo que no tiene la totalidad de las funcionalidades de un sistema operativo, diseñado exclusivamente para preparar todo lo que necesita el S.O para funcionar.
- **El BOOTSTRAP:** la palabra inglesa **bootstrapping** es generalmente un término utilizado para describir el arranque, o proceso de inicio de cualquier ordenador. Suele referirse al programa que arranca un S.O como por ejemplo GRUB, Lilo o NTLDR. Se ejecuta tras el proceso POST de la BIOS. También es llamado "Bootstrap Loader" (*cargador de inicialización)*. En países de habla hispana se utiliza comúnmente como ***Bootear***.

Cuando se instala un gestor de arranque debemos de tener presente que el de Windows no es capaz de detectar las particiones en las que está instalado Linux advirtiendo que es una partición desconocida no permitiendo el arranque del sistema, sin embargo el gestor de arranque de Linux si es capaz de detectar las particiones de Windows permitiendo arrancar dicho sistema desde el menú de arranque. Por esta razón cuando se realiza una instalación de varios sistemas en el mismo equipo se recomienda instalar Linux el último para que su gestor de arranque pueda detectar todos los S.O instalados en la máquina.

7.2.- Gestores de arranque de Windows

El sistema de arranque se llama **BCD store.** La herramienta en línea que usa para modificar los parámetros del arranque es **bcdedit.exe**. Por seguridad BCD se encuentra oculto en codificación binaria. El fichero BCD se encuentra en el directorio **boot**.

El fichero NTLDR (encargado de cambiar el modo de trabajo del procesador de real a protegido y de leer el boot.ini) se llama *Bootmgr*, y será el cargador de arranque o boot loader del sistema Windows.

Para poder modificar el fichero BCD será necesario ser usuario administrador. Además de poder modificar el BCD con el comando bcdedit podemos usar otras alternativas como:

- Desde el cuadro de diálogo *Inicio y recuperación* permite seleccionar el S.O de arranque por defecto y cambiar el valor de tiempo de espera para seleccionar una opción del menú (se encuentra en la pestaña *Opciones avanzadas* del cuadro de diálogo *Propiedades del sistema*).
- Ejecutando *Msconfig.exe* desde la barra de inicio, aparecerá una ventana con pestañas para configurar el sistema en apartados como General, Arranque, Servicios, Inicio de Windows, etc.

Para obtener ayuda detallada sobre el formato de los comandos y opciones del bcdedit, se escribe en una ventana de consola de línea de comandos la orden ***bcdedit.exe /?***.

Ejemplos: salir al símbolo del sistema con Inicio-Buscar o Ejecutar, escribir **cmd** y escribir las siguientes ordenes:

bcdedit /default ID	Para cambiar la entrada del sistema operativo predeterminado El ID especifica el GUID que se debe usar cuando expira el tiempo de espera y es un número

	hexadecimal que identifica al sistema operativo que hay en cada entrada de menú
bcdedit /default {cb8888bf-b7b8-48ff-951a-fa04564f5d7a}	El siguiente comando establece como predeterminado la entrada de sistema operativo identificado con el GUID: ***{cb8888bf-b7b8-48ff-951a-fa04564f5d7a}*** es el GUID predefinido
bcdedit /bootsequence {ID} {ID} {ID} …	Para modificar la secuencia de arranque en el siguiente reinicio
bcdedit /bootsequnce {803bb32-0gg4-11da-bs33-a12376eba25f} {cb8888bf-b7b8-48ff-951a-fa04564f5d7a}	El siguiente comando configura dos entradas del sistema operativo en la secuencia de arranque de una vez del administrador de arranque.