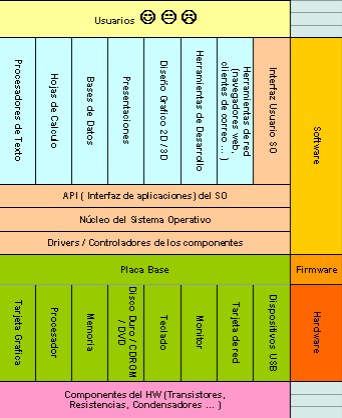
**TEMA 1**

**Esquema General de un Sistema Informático:**



Los usuarios también forman parte del sistema informático. Hay 4 tipos:

**Comunes** (Usuarios finales). **Operadores / Técnicos** (Usuarios Expertos).

**Administradores**. (Usuarios informáticos). **Desarrolladores** (Creadores del Sw)

**Introducción a SI.**

El hardware y el software de los sistemas informáticos han evolucionado de forma paralela y conjunta en las últimas décadas. EI SO debe actualizarse, ya que cada SO está adaptado para operar cada tipo de ordenador.

**1ª GENERACIÓN (1945-1955):** construidos con tubos de vacío, no existían sistemas operativos, se programaba directamente sobre el hardware, programas hechos en código máquina control de las funciones mediante paneles enchufables. Posteriormente aparecen las tarjetas perforadas, que permitían codificar instrucciones en una cartulina con puntos un programa cargador automatizaba la carga de programas ejecutables (SO rudimentario).

**2ª GENERACIÓN (1955-1965):** aparición de los transistores ordenadores más pequeños y potentes, la programación se realizaba en lenguaje ensamblador. Procesamiento por lotes SO ejecuta un proceso que dispone de todos los recursos.

**3ª GENERACIÓN** (1965-1980) aparición de los circuitos integrados, menor tamaño y relación precio/rendimiento. En los SO se desarrolla la multiprogramación (se cargan varios programas en memoria simultáneamente) y los sistemas de tiempo compartido (procesador se comparte entre varios programas).

**4ª GENERACIÓN (1980-HASTA HOY)** Grandes avances en la industria hardware, creación de los circuitos LSI (integrados a gran escala), aparecen los ordenadores personales. Desarrollado por Microsoft para MacOs primera interfaz gráfica basada en ventanas, iconos, menús y ratón.

Los SO evolucionan a interfaz cada vez más amigable al usuario. evolucionan Mac, Windows y Linux (aparece en los 90 a partir del núcleo desarrollado por Linus Torvalds)

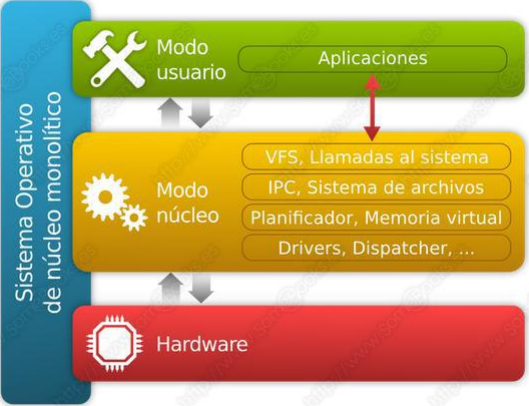
Avance importante fue el desarrollo de redes de ordenadores, que ejecutan sistemas operativos en red y sistemas operativos distribuidos.

**Concepto y objetivos de los SO.**

El SO es un conjunto de programas que se encarga de gestionar los recursos del HW y SW. Actúa como interfaz entre programas de aplicación del usuario y el hardware puro. Los SO administran recursos para controlar las funciones básicas del PC. Los recursos principales que administra son: El procesador, la memoria, dispositivos E/S y el sistema de archivos.

**Principales objetivos de los SO:**

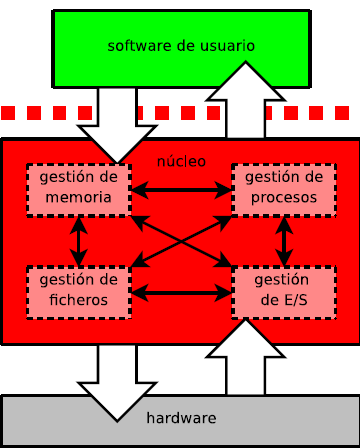
* Abstraer al usuario de la complejidad del hardware, que sea más fácil de utilizar, sin que el usuario tenga que entender cómo funciona el hardware.
* Eficiencia: recursos del ordenador eficientes. Ej: optimizar los accesos a disco para acelerar las operaciones E/S.
* Permitir la ejecución de programas: Usuario quiere ejecutar un programa, el SO realiza todas las tareas necesarias para ello como cargar instrucciones e iniciar dispositivos E/S.
* Acceder a los dispositivos entrada/salida: interfaz homogénea para los dispositivos para usuario usar de forma sencilla.
* Proporcionar una estructura y conjunto de operaciones para el sistema de archivos.
* Controlar el acceso al sistema y los recursos: en sistemas compartidos protección frente a usuarios no autorizados.
* Detección y respuesta ante errores: el SO debe prever todas las posibles situaciones críticas y resolverlas.
* Capacidad de adaptación: SO pueda evolucionar si surgen actualizaciones de SW y HW.
* Gestionar las comunicaciones en red: permitir al usuario usar con facilidad todo lo referente a la instalación y uso de las redes de ordenadores.
* Permitir a los usuarios compartir recursos y datos: relacionado con el anterior, da al SO papel de gestor de recursos de una red.



**MODO USUARIO/NÚCLEO o kernel.**

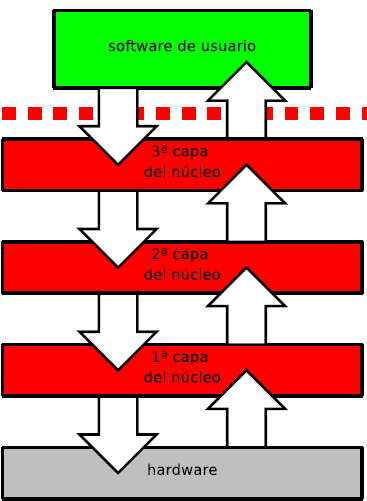
* Modo usuario: menos privilegiado, no se permite el acceso directo al HW. Usan el API del sistema para requerir los servicios del SO y usan su propio espacio de direcciones de memoria. Este es el modo de ejecución de los programas de aplicación.
* Modo Núcleo o Kernel: modo privilegiado, acceso directo a toda la memoria (incluido espacios de procesos en ejecución). Puede acceder a todo el HW disponible. En este modo solo se ejecutan algunas partes de SO.

**SO en base a su estructura, por sus servicios y por su forma:**

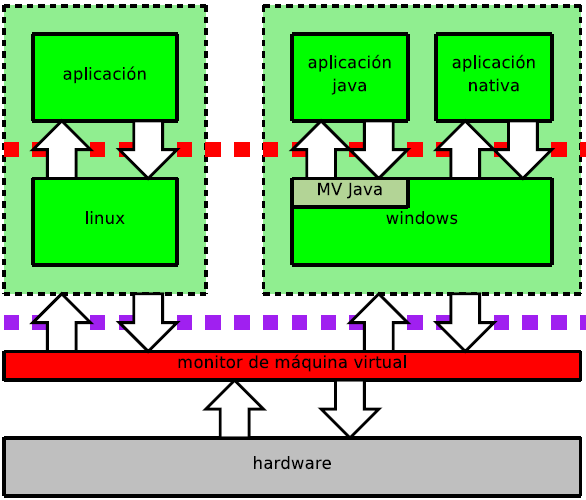


**SO por estructura:**

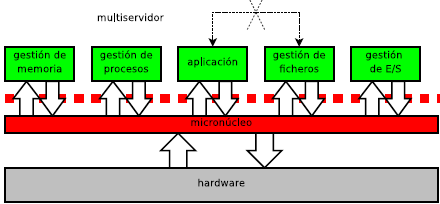
* Monolíticos: Es la estructura de los primeros sistemas operativos, un sólo programa desarrollado con rutinas entrelazadas. SO hechos a medida pero difíciles de mantener.



* Jerárquicos: Según necesidades usuarios crecieron, hizo necesaria una mayor organización del software del SO, dividiéndose en subsistemas, diferenciados por funciones, jerarquizados entre ellos y con una interfaz clara para interoperar con los demás elementos. Ejemplo: MULTICS



* Máquina Virtual:El objetivo es integrar distintos SO en una misma máquina virtual dando la sensación de ser varios ordenadores diferentes. Son réplicas del original, pero no son reales. Ejemplo:VirtualBox



* Microkernel o Cliente-Servidor: El modelo del núcleo de estos SO distribuye las tareas en porciones de código modulares y sencillas. Aisla del sistema, su núcleo, operaciones E/S, gestión de memoria, sistema de archivos, etc. Incrementa la tolerancia a fallos, la seguridad y la portabilidad entre plataformas de HW. Ejemplo: Mac OS X

**SO por sus servicios:**

* Monousuario: SO que soportan a un usuario a la vez, sin importar el número de procesos pueda ejecutar a la vez.
* Multiusuario: SO que da servicio a más de un usuario a la vez. Puede ser por varias terminales conectadas al ordenador o red de comunicaciones. No importa si mono/multiprocesador ni mono/multitarea.
* Monotarea: SO que sólo permite una tarea a la vez por usuario. Puede ser multiusuario y monotarea. Cada usuario 1 tarea.
* Multitarea: varias tareas al mismo tiempo por usuario.
* Monoprocesador: SO que maneja sólo un procesador. Si el ordenador tiene más, son inútiles.
* Multiprocesador: SO capaz de soportar ordenadores con más de 1 procesador y utilizarlos todos para distribuir su carga de trabajo. Existen 2 formas de que funcionen:

1. Simétricos: los procesos son enviados indistintamente a cualquiera de los procesadores disponibles.
2. Asimétricos: uno de los procesadores actúa como maestro o servidor y distribuye la carga de procesos a los demás (esclavos).

**SO por su forma.**

* SO en red: Interactúan con SO de otros ordenadores a través de la red. Usuario debe conocer la ubicación de los recursos en red
* SO distribuidos: Abarcan los servicios de red, las funciones se distribuyen entre diferentes ordenadores, logrando integrar recursos en una sola máquina virtual que es a la que el usuario accede. Usuario no necesita saber la ubicación de los recursos.

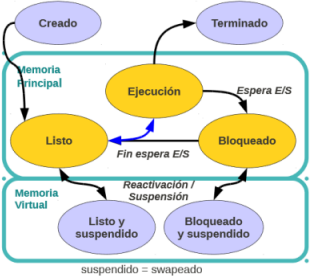
**El núcleo del SO.**

El núcleo supone la parte principal del código de un SO y se encarga de controlar y administrar los servicios y peticiones de recursos. El núcleo reside por lo general en la memoria principal mientras otras partes de SO son cargadas en la memoria principal sólo cuando son utilizadas.

Para administrar y controlar se divide en:

**Gestión de procesos.**

Un proceso en un programa en ejecución. Un proceso simple tiene un hilo. Un hilo es una tarea que puede ser ejecutada en paralelo con otra tarea, por lo que los hilos de ejecución permiten a un programa realizar varias tareas a la vez.

Los SO modernos pueden diferenciar los procesos en diferentes estados:

* Creado: Se le están creando su tabla y cargando su código
* Listo: son los que pueden pasar a estado de ejecución si el planificador del SO los selecciona, según orden de llegada o prioridad.
* En ejecución: está siendo ejecutado en el procesador en un momento dado. Tiene el control del procesador en este momento.
* Bloqueado: están esperando la respuesta de algún recurso no disponible para poder continuar con su ejecución, por ejemplo, una operación de entrada/salida.
* Listo y suspendido: El proceso ha sido suspendido por cualquier causa pero no presenta motivos para ser bloqueado.
* Bloqueado y suspendido: El proceso ha sido suspendido a la espera de un evento, sin que hayan desaparecido las causas de su bloqueo. Cuando haya desaparecido la causa del bloqueo pasará al estado listo y suspendido.

**Planificación del procesador.**

En la planificación del procesador se decide cuánto tiempo de ejecución se le asigna a cada proceso del sistema, en qué momento y el estado en que estará el proceso en cada momento. Si el sistema es monousuario y monotarea no habrá que decidir, pero sí en el resto de los casos.

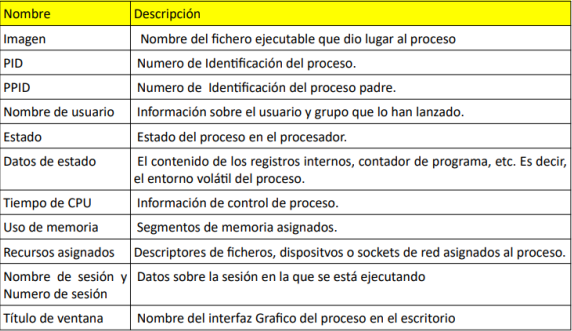
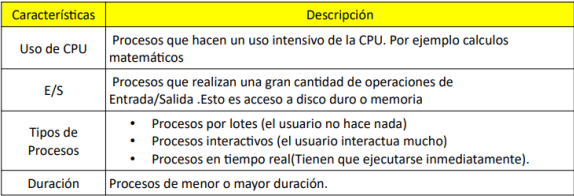


Tabla de control del proceso:

El SO almacena en esta tabla la información relativa a cada proceso que se está ejecutando en el procesador.

Estrategia de planificación: es para buscar que los procesos obtengan sus turnos de ejecución de forma apropiada junto con un buen rendimiento y minimización de la sobrecarga del planificador. Se buscan 4 objetivos:

* *Todos los procesos en algún momento obtienen su turno de ejecución o intervalos de tiempo de ejecución hasta su terminación con éxito.*
* *El sistema debe finalizar el mayor número de procesos por unidad tiempo.*
* *El usuario no percibirá tiempos de espera demasiado largos.*
* *Evitar el aplazamiento indefinido, los procesos deben terminar en un plazo finito de tiempo.*



Tipos de procesos: La carga de trabajo de un sistema informático a otro puede variar, depende de las características de los procesos.

Existen distintas planificaciones en las cuales hay algoritmos para decidir el orden de ejecución y prioridad:

Planificación no apropiativa: Cuando a un proceso haciendo uso de la CPU (ejecutándose) no se le puede arrebatar. No se puede ejecutar otro proceso hasta que este finalice. Problema, si el proceso contiene ciclos infinitos los demás procesos quedan aplazados infinitamente.

Planificación apropiativa: el SO puede arrebatar el uso de la CPU a un proceso en ejecución. Existe un reloj que lanza interrupciones periódicas para que el planificador tome el control y decida si se sigue ejecutando el mismo proceso o le da su turno a otro proceso.

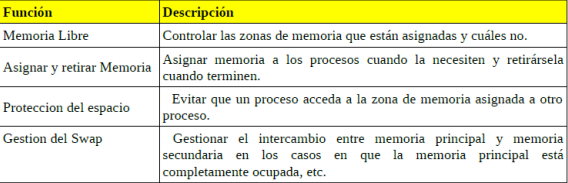
Algoritmos usados:

* *FIFO. First In First Out: primero en entrar, primero en salir. Es un algoritmo que establece que el primer proceso que llega, se ejecuta y una vez terminado, se ejecuta el siguiente . Es muy sencillo y simple por lo que sobrecarga poco el sistema, pero es el que menos rendimiento ofrece.*
* *Algoritmo basado en prioridades. A cada proceso en el sistema se le asigna un número basándose en algún criterio (número bajo es alta prioridad). Se favorece a los procesos de alta prioridad al momento de asignar la CPU. No apunta a reducir el tiempo de espera promedio general. Los procesos importantes serán atendidos sin caer en retardos innecesarios. Las prioridades pueden ser estáticas o dinámicas (sistema por envejecimiento). Produce Muerte por inanición (si siempre hay procesos con mayor prioridad, nunca será ejecutado).*

*Ejemplos:*

1. *Shortest Job First: Se ejecutan los procesos en orden de menor a mayor según el tiempo de ejecución de cada uno. En caso de empate FCFS (first come first served).*
2. *Round Robin: A cada proceso se le asigna un intervalo de tiempo (quantum), si el proceso continúa su ejecución al finalizar su intervalo de tiempo o quantum, es adelantado y la CPU es asignada a otro proceso. Si termina antes, es asignado a otro proceso en ese mismo instante.*

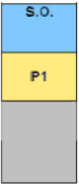
**Gestión de memoria de un SO.**



**Funciones**: para que un proceso se pueda ejecutar requiere estar cargado en memoria principal. Ningún proceso se puede activar antes de que se le asigne el espacio de memoria. El SO tendrá que gestionar la memoria que es otro recurso clave y la parte encargada de ello se denomina gestor de memoria.

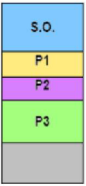
**Requisitos de la gestión de memoria**:

* Reubicación: En sistema multitarea, el gestor de memoria decide qué zonas de memoria asigna a cada proceso y que zonas descarga.
* Protección: El gestor de memoria debe evitar que los procesos cargados en memoria interfieran unos con otros accediendo a zonas de memoria que no les corresponden.
* Control de memoria: El gestor de memoria, tiene que controlar las zonas de memoria libres, cuales están asignadas y saber las zonas de memoria de cada proceso.
* Controlar y evitar en lo posible casos de fragmentación de la memoria: Fragmentación interna: ocurre cuando se guarda un proceso más pequeño que la partición generada. Fragmentación externa: ocurre cuando las particiones se dividen cada vez más y van quedando huecos pequeños y dispersos difícilmente reutilizables.
* Organización lógica y física: El gestor de memoria se encarga de gestionar la transferencia de información entre la memoria principal y la secundaria. A veces por falta de memoria no se puede almacenar el proceso/procesos que se pueden ejecutar. Entonces estos pueden ser intercambiados a memoria secundaria y más tarde, si es necesario, vueltos a cargar en memoria.



**Gestión de memoria en SO monotarea.**

Los procesos se ejecutan secuencialmente a medida que van terminando los anteriores. Se trata del esquema más sencillo, en cada momento la memoria alberga un solo proceso y reserva otra zona de la memoria para el sistema operativo. Necesita un mecanismo de protección para evitar accesos a la parte del sistema operativo de los procesos de usuario.



**Gestión de memoria en SO multitarea.**

Actualmente mayoría de SO son multitarea. Los procesos deberán estar simultáneamente en memoria, debe haber mecanismos de gestión para distribuir la memoria principal entre todos estos procesos que quieren ejecutarse.

**Intercambio o Swapping.**

El gestor de memoria sacará de la memoria algunos procesos (bloqueados, suspendidos, que esperan final de alguna operación) y los llevará a un área de la memoria secundaria, conocida como área de intercambio o de swap. A este proceso se le llama intercambio o swapping, los procesos permanecen ahí hasta que existan huecos libres en la memoria principal.

**Memoria Real y Virtual.**

**Memoria real** (es memoria física) en las técnicas de mem real, se cargan los procesos enteros y físicamente en memoria. Problema con los procesos grandes, puede no caber en la partición de la memoria y no ser ejecutados.

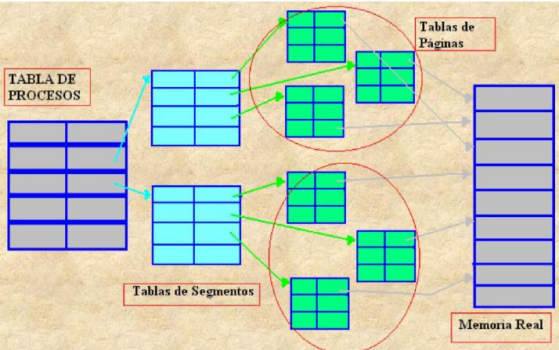
**Memoria Virtual**

Divide los procesos en varias partes y carga sólo algunas de ellas en memoria.son tan usuales que están hasta integradas en la CPU.

Técnicas de la memoria virtual:

* *Paginación pura:* divide en marcos de páginas la memoria principal, son particiones de igual tamaño. Cada proceso es dividido en páginas del mismo tamaño que el marco. Las páginas no es necesario que estén contiguas en memoria. reduce fragm externa(muchas particiones pequeñas difíciles de completar) pero aparece cierta fragm interna(sobrante de espacio por proceso más pequeño que partición).
* *Segmentación pura*: Cada proceso se divide en segmentos, estos no es necesario que sean del mismo tamaño hasta un límite. El proceso carga sus segmentos en particiones dinámicas en memoria y no tienen que estar contiguas. Reduce fragm interna (sobrante de espacio por proceso más pequeño que partición). Cuando no todos los segmentos estén cargados en memoria producirá error de segmento o página, y el gestor traerá dicha parte del proceso desde el disco a la memoria.

La utilización de estas técnicas por la mem virtual se conocen como:

1. *Memoria virtual paginada:* usan paginación simple, pero no es necesario cargar todas las páginas del proceso, las que no se encuentren y se necesiten serán traídas de forma automática de disco a memoria.
2. *Memoria virtual segmentada:* usan segmentación simple, pero no es necesario cargar todos los segmentos de un proceso, los segmentos no asignados en memoria se traerán cuando sean referenciados y requeridos.
3. *Combinación de segmentación y paginación*: Cada proceso se divide en segmentos que a su vez se dividen en páginas. La memoria real es dividida en marcos. Ofrece lo mejor de cada tipo.

**GESTIÓN DE E/S.**

Es la parte del sistema operativo que se encarga de este proceso es la gestión de los dispositivos periféricos. Estos realizan la obtención y muestra de información (E/S). Existen de 3 tipos:

* De entrada: reciben información y la transmiten al ordenador. Ej: teclado o ratón.
* De salida: muestran la información procesada por el ordenador. Ej: el monitor, la impresora
* De entrada y salida: realizan ambas funciones. Ej: monitor táctil, el disco duro,etc.

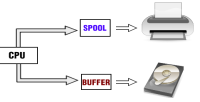
El SO actúa como intermediario haciendo que las funciones se realicen de forma adecuada eficiente y de forma estandarizada gracias a:

**Controladores de dispositivos.**

Se encarga de la comunicación entre la CPU y el dispositivo. Actúan como interfaz entre los programas y el hardware. Son suministrados por el fabricante del dispositivo o por el desarrollador del sistema operativo.

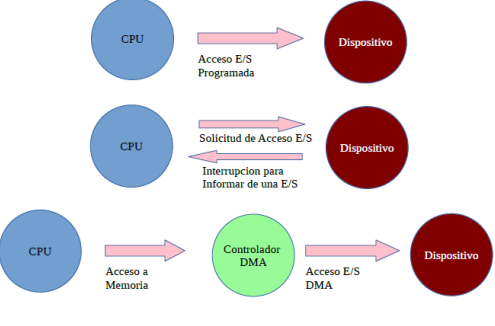
**Estructura de datos de la E/S del SO.**

El SO cuenta con una estructura para manejar la información y comunicación entre dispositivos y entre estos y la CPU. los más usado son:

* Spools: Se utiliza en dispositivos que no admiten intercalación, se almacenan de forma temporal en una cola situada en un dispositivo de almacenamiento masivo (spool). Como la impresora (no puede iniciar proceso hasta que el anterior haya terminado).
* Buffers: El buffer es un retenedor de información. Se usa para dispositivos que pueden atender peticiones de distintos orígenes, en este caso los datos no tienen que enviarse completos, el buffer actúa reteniendo las porciones temporalmente. También se usa para acoplar velocidades entre distintos dispositivos.

**Técnicas de la E/S.**

Existen:

* E/S programada: La CPU lleva a cabo la transferencia de los datos. Repercute en la velocidad del ordenador ya que la CPU deja lo que está haciendo para ocuparse del proceso de E/S.
* E/S por interrupciones: La CPU ejecuta la transferencia pero el inicio es pedido por el periférico que indica así su disponibilidad.
* E/S por acceso directo a memoria (DMA): La transferencia es realizada por un controlador especializado. Acelera enormemente el proceso de E/S y libera a la CPU del trabajo. La CPU inicia el proceso, pero luego este continúa sin necesitar la CPU.

**Interrupciones del sistema.**

Las interrupciones del sistema son llamadas por programas o procesos que se están ejecutando para romper el flujo de tareas normales del procesador y hacer algo más urgente y luego regresar a lo que estaba haciendo anteriormente.