

Arquitectura del sistema informático

1) ¿Qué es un sistema operativo?

¿Que es?: Un sistema operativo (SO o OS) o software base es el conjunto de programas que efectúan la gestión de los recursos de un sistema informático, y permite su normal funcionamiento.

El sistema operativo es fundamental para un sistema informático. Sin él, el hardware sólo sería una entidad física sin utilidad, ya que no podríamos asignarle ningún tipo de tarea.

Su **funciones** principales son:

- Proporcionar una **interfaz** entre los usuarios y el hardware.
- **Gestionar los recursos** del sistema informático para la óptima ejecución de las tareas:
 - CPU → **Gestión de los procesos**
 - Memoria principal → **Gestión de la memoria**
 - Memoria secundaria → **Gestión de archivos**
 - Periféricos → **Gestión de los dispositivos de E/S**
 - Red → **Gestión de los recursos de red**
- Implementar **sistemas de protección**.
- Proporcionar **conectividad** a otros sistemas.
- **Tratamiento de errores y seguridad**.

Nota: los SO también incluyen sistemas de seguridad específicos para que los usuarios no puedan dañar directamente el hardware, también se le permite cerrar errores muy graves.

2) Evolución de los sistemas operativos

Generación 0 (1943-1955): Aparece el primer ordenador *Colossus*. El automatismo es muy reducido. El SO es el ser humano (automatismo). La interacción con el sistema se realiza mediante una consola. Los programadores introducen los programas mediante cableado. El único periférico de salida es la impresora que muestra los resultados

Este modo de funcionamiento es lo que se llama **monoproceso**:

- Los programas había que cargarlos de uno en uno y manualmente.
- La ejecución también se efectúa de uno en un proceso (monoprogramado).

La pérdida de eficacia era muy grande debido a que no existía un sistema operativo que planificarse ni automatizar la ejecución de los programas.

Nota: DIFERENCIA ENTRE PROCESO Y PROGRAMA. Programar es hacer un programa, este

compilado o no en sí es un programa, que es un conjunto de procesos que se ejecutan, un proceso es un conjunto de sentencias que se leen de arriba abajo a no ser que haya bucles o condicionales. Se pueden ejecutar dos procesos al mismo tiempo..

Generación 1 (1955-1959): Aparece el SO Monitor. Tienen un sistema operativo muy básico llamado Monitor. Los programas se introducían con una unidad de carga. Los programadores introducen mediante tarjetas perforadas. El único periférico de salida sigue siendo la impresora.

El modo de funcionamiento de estos sistemas era de **procesamiento por lotes**:

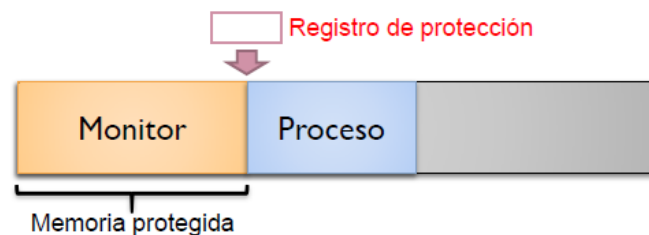
- Los programas están estructurados como secuencias lineales de tareas u órdenes.
- El Monitor se limita a cargar los programas en memoria para ser ejecutados de principio a fin (si no ocurre ningún error o se agota un tiempo de espera).
- El control del hardware se alterna entre el programa en uso y el Monitor (no hay interrupciones).

El sistema Monitor es lo que llamamos un **sistema operativo de núcleo monolítico**. Es decir un conjunto de rutinas enlazadas entre sí formando un único cuerpo.

El Monitor es un sistema operativo que trabaja en modo **Monoprogramado**.

Se implementa un primer sistema de protección de la memoria:

- Las direcciones de memoria donde reside el Monitor no se pueden modificar por una aplicación.
- Si ocurriese el procesador debería detectarlo y devolver el control al Monitor.



Nota: núcleo monolítico es un tipo de núcleo o kernel de un sistema operativo que comprende un núcleo grande y complejo donde se concentran todas las funcionalidades posibles (planificación, sistema de archivos, redes, controladores de dispositivos, gestión de memoria, etc).

Este núcleo está programado de forma no modular y puede tener un tamaño considerable. A su vez, cada vez que se añada una nueva funcionalidad, deberá ser re-compilado en su totalidad y luego reiniciado. Todos los componentes funcionales del núcleo tienen acceso a todas sus estructuras de datos internas y a sus rutinas.

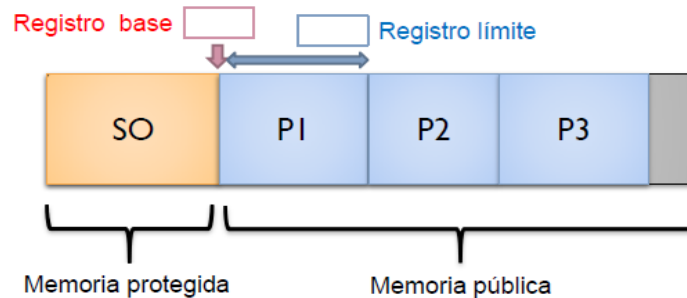
MS2 y UNIX comenzaron usando núcleos monolíticos pero con el paso del tiempo debido a la popularidad y reformas llegaron a ser sistemas muy pesados por lo que tuvieron que cambiar de sistema.

Generación 2 (1959-1964): Aparece el SO Multiproceso. Aumenta la complejidad del sistema operativo. Empieza a utilizarse el soporte magnético para almacenar programas. Se sigue utilizando las

tarjetas perforadas como soporte auxiliar. El único periférico de salida sigue siendo la impresora

Aparece el concepto de **multiprogramación** o multitarea.

- En un SO Multiprogramado aparentemente se pueden realizar varias tareas simultáneamente.
- Pero en realidad se asigna un tiempo finito y muy corto (quantum) a cada tarea, alternándose rápidamente y creando la ilusión de simultaneidad.
- Para implementar la multiprogramación en un sistema operativo se necesita necesariamente cargar todos los procesos que se están ejecutando al mismo tiempo en memoria principal.
- No se pueden ejecutar más procesos de los que permite esta memoria.
- Aparece el concepto de Interrupción.



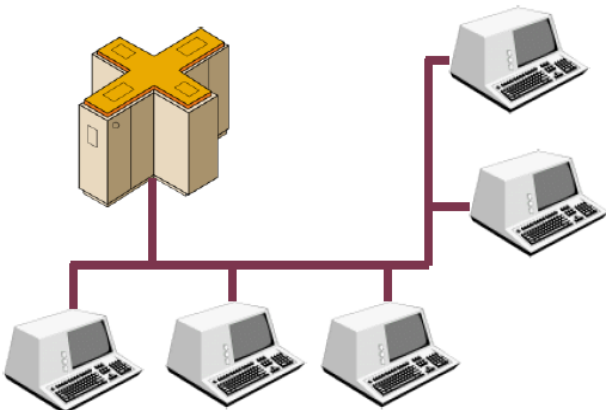
Nota: En los antiguos sistemas monoprogramados, cuando un proceso en ejecución requería hacer uso de un dispositivo de E/S, el procesador quedaba ocioso mientras el proceso permaneciese en espera y no retomara su ejecución. En cambio, en un sistema multiprogramado, cuando un proceso P_x concluye o se bloquea (en espera de una operación de E/S), el núcleo del sistema operativo toma el control de la CPU para efectuar lo que se denomina un «cambio de contexto», a fin de dar turno a otro proceso P_y para que se ejecute. Nótese que cuando el proceso P_x recupere su turno, reanudará su ejecución justo en el punto exacto en que se interrumpió. Este mecanismo supone un mejor reparto de la carga de trabajo y un mayor aprovechamiento de la CPU, lo que redunda en un mayor rendimiento del sistema.

Generación 3 (1964-1977): Aparece el SO multiusuario. Aumenta la complejidad del sistema operativo. Prolifera el soporte magnético como soporte de memoria. Se empieza a utilizar el monitor como periférico de salida. Aparecen los primeros protocolos de red

Es la etapa de los **Mainframes**, ordenadores centrales que controlan un número variable de terminales “tontos”.

- Aparece el concepto de modo **multiusuario**: Los usuarios tienen la ilusión de que utilizan los recursos simultáneamente.
- Aparece el concepto de modo **tiempo compartido**: Los usuarios tienen la ilusión de que utilizan los mismos recursos a la vez.

La automatización informática llega a todos los estratos



de la vida cotidiana (tráfico, aeropuertos, fábricas...).

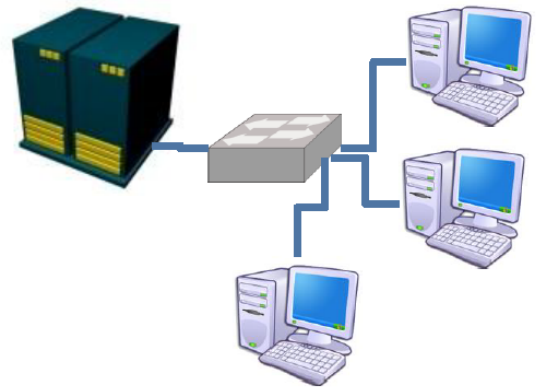
- Aparece el concepto de modo **tiempo real**: Se procura que los tiempos de respuesta sean mínimos. Este modo es incompatible con el modo tiempo compartido.

Los ordenadores de esta etapa son de propósito general (multimodo), capaces de trabajar en lotes, en modo multiprogramado, en tiempo real o en tiempo compartido.

Generación 4 (1977- Futuro): Aparece el primer PC. Los sistemas operativos sufren una gran evolución. Aparecen los sistemas gráficos. Las redes toman fuerza. Aparecen nuevos protocolos más seguros. Aparece el concepto de máquina virtual.

Ordenador personal:

- Los terminales son ahora independientes y computadoras en si mismas.
- Tienen menos potencia que los mainframes.
- Son más accesibles al usuario.
- Su interfaz es más cómoda y amena → Entorno gráfico.



Se idean sistemas operativos capaces de trabajar en una organización en red cliente-servidor. La multiprogramación se hace esencial en los sistemas. Aparecen redes distribuidas. Se fija como objetivo introducir un ordenador en cada casa y conectarlo a la red mundial (Internet). Se multiplica la funcionalidad y versatilidad. Aparece el concepto de multimedia.

Nota: aparecen los sistemas operativos distribuidos, un sistema operativo distribuido es una colección de computadoras independientes que aparecen ante el usuario del sistema como una única computadora.

Un sistema operativo distribuido es aquel común en varios equipos que mantiene comunicación a través de la red (sucursales de un banco).

Sistemas operativos de 4ª generación de Microsoft			
Año	Núcleo MS-DOS	Núcleo Windows NT	Núcleo Windows CE
1985	Windows 1.x (16 bits)		
1987	Windows 2.x (16 bits)		
1990	Windows 3.x (16 bits)		
1993		Windows NT (32 bits)	
1995	Windows 95 (16-32 bits)		
1996			Windows CE 1.x (32 bits/ARM)
1997			Windows CE 2.x (32 bits/ARM)
1998	Windows 98 (16-32 bits)		
2000	Windows Me (16-32 bits)	Windows 2000 (32 bits)	Windows CE 3.x (32 bits/ARM)
2001		Windows XP (32/64 bits)	
2002			Windows CE 4.x (32 bits/ARM)
2003		Windows 2003 Server	
2004			Windows CE 5.x (32 bits/ARM)
2006			Windows CE 6.x (32 bits/ARM)
2007		Windows Vista (32/64 bits)	
2008		Windows 2008 Server	
2009		Windows 7 (32/64 bits)	
2010			
2011			Windows CE 7.x (32 bits/ARM)
2012		Windows 8 (32/64 bits/ARM)	

Sistemas operativos de 4ª generación de Apple			
Año	Familia Lisa	Familia Macintosh	Familia Macintosh basado en BSD
1982	Lisa OS		
1984	Lisa OS II	Mac System 1.x	
1985		Mac System 2.x	
1986		Mac System 3.x	
1987		Mac System 4.x Mac System 5.x	
1988		Mac System 6.x	
1991		Mac System 7.x	
1997		Mac OS 8	
1999		Mac OS 9	Mac OS X Server 1.0 <i>Rhapsody</i>
2000			Mac OS X Beta <i>Kodiak</i> /Server
2001			Mac OS X 10.0 <i>Cheetah</i> /Server Mac OS X 10.1 <i>Puma</i> /Server
2002			Mac OS X 10.2 <i>Jaguar</i> /Server
2003			Mac OS X 10.3 <i>Panther</i> /Server
2005			Mac OS X 10.4 <i>Tiger</i> /Server
2007			Mac OS X 10.5 <i>Leopard</i> /Server
2009			Mac OS X 10.6 <i>Snow Leopard</i> /Server
2011			Mac OS X 10.7 <i>Lion</i> /Server
2012			Mac OS X 10.8 <i>Mountain Lion</i> /Server
2013			Mac OS X 10.9 <i>Mavericks</i> /Server

Sistemas operativos de 4ª generación en dispositivos móviles				
Año	Windows CE	Windows NT	Apple iOS	Nokia/Accenture
1989				EPOC16
1997				EPOC32 1.x
1998				EPOC32 4.x
1999				EPOC32 5.x
2000	PocketPC 2000			
2001				Symbian OS 6.x
2002	Pocket PC 2002			
2003	Windows Mobile 2003			Symbian OS 7.x
2004				Symbian OS 8.x
2005	Windows Mobile 5.x			
2006				Symbian OS 9.x
2007	Windows Mobile 6.x		iOS 1.x	
2008			iOS 2.x	
2009			iOS 3.x	
2010	Windows Phone 7.x		iOS 4.x	
2011			iOS 5.x	Symbian OS 10.x
2012		Windows Phone 8.x Windows RT	iOS 6.x	
2013			iOS 7.x	

Sistemas operativos de 4ª generación en dispositivos móviles				
Año	Google (Linux)	Nokia (Linux)	Intel (Linux)	Samsung (Linux)
2005		Maemo 1.x		
2006		Maemo 2.x		
2007		Maemo 3.x	Moblin 1.x	LiMo 1.x
2008	Android 1.0 Apple Pie	Maemo 4.x		
2009	Android 1.1 Banana Bread Android 1.5 Cupcake Android 1.6 Donut Android 2.0 Eclair	Maemo 5.x	Moblin 2.x	LiMo 2.x
2010	Android 2.2 Froyo Android 2.3 Gingerbread	MeeGo 1.x		
2011	Android 3.x Honeycomb Android 4.0 Ice Cream Sandwich			LiMo 4.x
2012	Android 4.1 Jelly Bean Android 4.2 Gummy Bear Android 4.3 Gominola		Tizen 1.x	
2013	Android 4.4 KitKat Android 5.0 Key Lime Pie		Tizen 2.x Tizen 3.x	

¿Quinta generación??

Japón anunció en 1979 la intención de desarrollar sistemas operativos de quinta generación. A este esfuerzo se le sumaron EEUU y otros países europeos.

La idea era crear inteligencia artificial (AI) que emulase el pensamiento humano. El proyecto fue un

rotundo fracaso y después de muchas decepciones, se abandonó en 1995. Algunos de sus logros están hoy en día en los ordenadores de cuarta generación.

- Integración de las redes inalámbricas.
- Reconocimiento de voz.
- Reconocimiento de imágenes

Dispositivos móviles y la 3ª generación

Hoy, poco a poco, principalmente en dispositivos móviles, vuelven a las filosofías adoptadas en la tercera generación. Los nuevos sistemas operativos tienen a ser más ligeros y eficientes, optimizados para un hardware de baja prestaciones. Los sistemas cada vez más delegan sus recursos de cómputo o de almacenaje a servidores externos llamadas **nubes de computación**. Esto produce un abaratamiento de los sistemas informáticos hasta tal punto de volver a ser meros terminales. Cobra muchísima importancia la estabilidad y el ancho de banda de la conexión a Internet.

Modos de explotación actuales

- **Modos según el número de procesos simultáneos:**

- **Modo Monoprogramado:** El sistema ejecuta los procesos de uno en uno.

Nota: Es muy eficiente se usa para cálculos intensivos.

- **Modo Multiprogramado:** El sistema crea la ilusión de que varios procesos se ejecutan simultáneamente.

Nota: Es menos eficiente, necesita de un sistema gráfico y no ejecuta más de una función a la vez, solo lo aparenta.

SO Monoprogramados	SO Multiprogramados
Monitor Ms-DOS	Sistemas Windows Sistemas UNIX Distribuciones Linux Sistema Lisa Sistemas Mac X/iOS Sistemas Symbian Sistemas Blackberry Sistemas Middleware QNX Sistemas Middleware Linux BeOS

Nota: BeOS: es un sistema operativo que se diseñó para que funcionase a tiempo real. No está basado ni en Windows ni en Unix.

- **Modos según el número de usuarios:**

- **Modo Monousuario:** Un único usuario utiliza los recursos del sistema en un mismo instante.
- **Modo Multiusuario:** Varios usuarios pueden utilizar los recursos del sistema simultáneamente.

SO Monousuarios	SO Multiusuarios
Windows XP Windows 7 Windows 8 Familia Windows CE Sistemas iOS Sistemas Symbian Sistemas Blackberry Sistemas Middleware QNX Sistemas Middleware Linux BeOS	Windows 2003 Server Windows 2008 Server Sistema UNIX Distribuciones Linux Sistemas Mac X

Nota: Los sistemas multiusuario puede funcionar en monousuario pero los monousuarios no pueden funcionar como multiusuarios.

- **Modos según el tiempo de respuesta:**

- **Tiempo compartido:** Cuando lo prioritario es que cada usuario disponga de los recursos de forma equitativa. Propio de los ordenadores centrales (Mainframes).
- **Tiempo real:** Cuando lo prioritario es que la respuesta sea lo más rápida posible, antes de sobrepasar un umbral de tiempo.

- **Modos según el número de procesadores:**

- **SO Monoprocador:** Sólo hay un procesador. Y por tanto sólo se puede ejecutar un proceso al mismo tiempo.
- **SO Multiprocador:** Varios procesadores ejecutan simultáneamente cauces de proceso diferentes.
 - **Arquitectura NUMA** (Multiprocesamiento Asimétrico): Los procesadores disponen de regiones de memoria aisladas con buses de datos diferentes. Normalmente hay un procesador maestro (master) que delega funciones y asigna recursos a los demás esclavos (slaves).
 -
 - **Arquitectura UMA** (Multiprocesamiento Simétrico): Los procesadores comparten un único espacio de memoria y un bus de datos único.
 - - Multiprocesamiento en chips diferentes (SMP): Los procesadores solo comparten la memoria y el bus de datos.
 - Multiprocesamiento a nivel de chip (CMP): Los procesadores comparten memoria, buses, caché y otros recursos.

Nota: No es lo mismo un sistema mono / multiprogramado que mono procesador. Un sistema monoprocesador puede ser multiprogramado. Pero no tiene sentido que un sistema multiprocesado sea

monoprogramado, porque se desaprovechan los recursos.

Propósito de los sistemas operativos

- **SO para procesamiento por lotes:** Diseñados para el procesamiento continuo de tareas.
- **SO para servidores:** diseñados para dar servicios a otros ordenadores. Gestionan un amplio ancho de transmisión.
- **SO para ordenadores:** pensados para usuarios de PC con pocos conocimientos informáticos.
- **SO para dispositivos móviles:** pensados para tabletas y teléfonos móviles.
- **SO embebidos:** se encuentran en el firmware de dispositivos electrónicos menores, son muy sencillos y no permiten la reconfiguración.
- **SO para tarjetas inteligentes:** se encuentran embebidos en chips de tarjetas TCI.

Servidores	Clientes	Moviles	Embebidos	TCI
Windows 2003 Server	Windows XP	Windows Mobile	Windows CE	Java Card
Windows 2008 Server	Windows 7	Windows Phone	QNX	
Windows 2012 Server	Windows 8	Sistemas iOS		
Sistema UNIX	Distribuciones Linux	Sistemas Symbian		
Sistemas Mac X	Sistemas Mac X	Sistemas Blackberry		
Distribuciones Linux	BeOS	Middleware QNX		
		Middleware Linux		

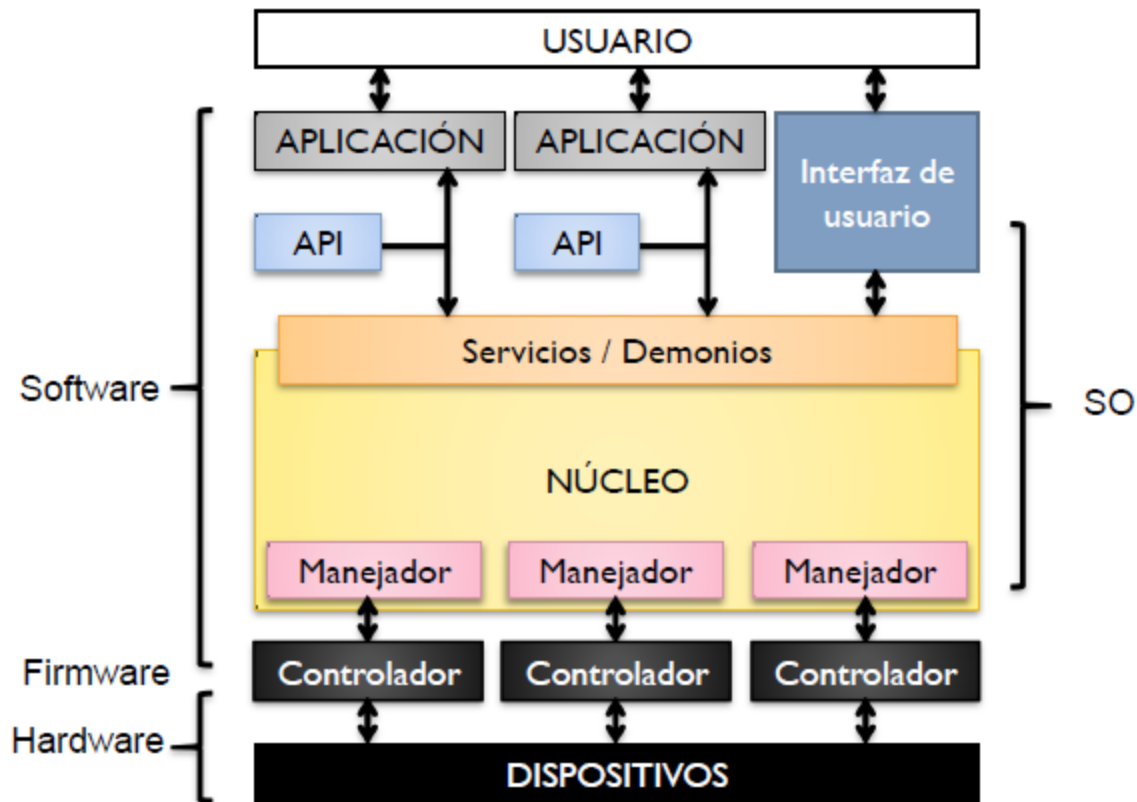
3) Estructura de un sistema operativo

¿Qué es?: Es la forma de organizar un SO para hacer posible la interacción entre el hardware y el usuario.

Se establece un **sistema por capas:**

1. **Interfaz de usuario:** Permiten al usuario comunicarse con el sistema operativo.
 - Línea de comandos (CLI) – Modo Texto / Simbolo de Sistema
 - Entorno gráfico (GUI) – Modo Gráfico
2. **Interfaz de aplicación (API):** Permite a la aplicaciones solicitar recursos al sistema operativo (llamadas al sistema).
3. **Servicios/Demonios:** Procesos no controlados por el usuario. Gestionan el hardware ejecutándose infinitamente.

4. **Núcleo:** Controla el funcionamiento de todo el sistema operativo, es quién realiza la gestión de los recursos.
5. **Manejadores** (Drivers): Cada uno permite la comunicación del sistema operativo con un elemento del hardware concreto.



Modo protegido

En un sistema informático es muy importante limitar el acceso al usuario de ciertos recursos al ser estos muy delicados.

En modo normal o **modo usuario**, lo normal es que no se puedan realizar todas las tareas. Este modo es el que utilizan las llamadas a sistema de las aplicaciones y las solicitudes del usuario.

En modo protegido o **modo kernel**, se tiene prioridad absoluta y se puede realizar cualquier tarea. Sólo el núcleo del sistema operativo puede ejecutar instrucciones en modo kernel. Este modo tiene acceso directo al hardware.

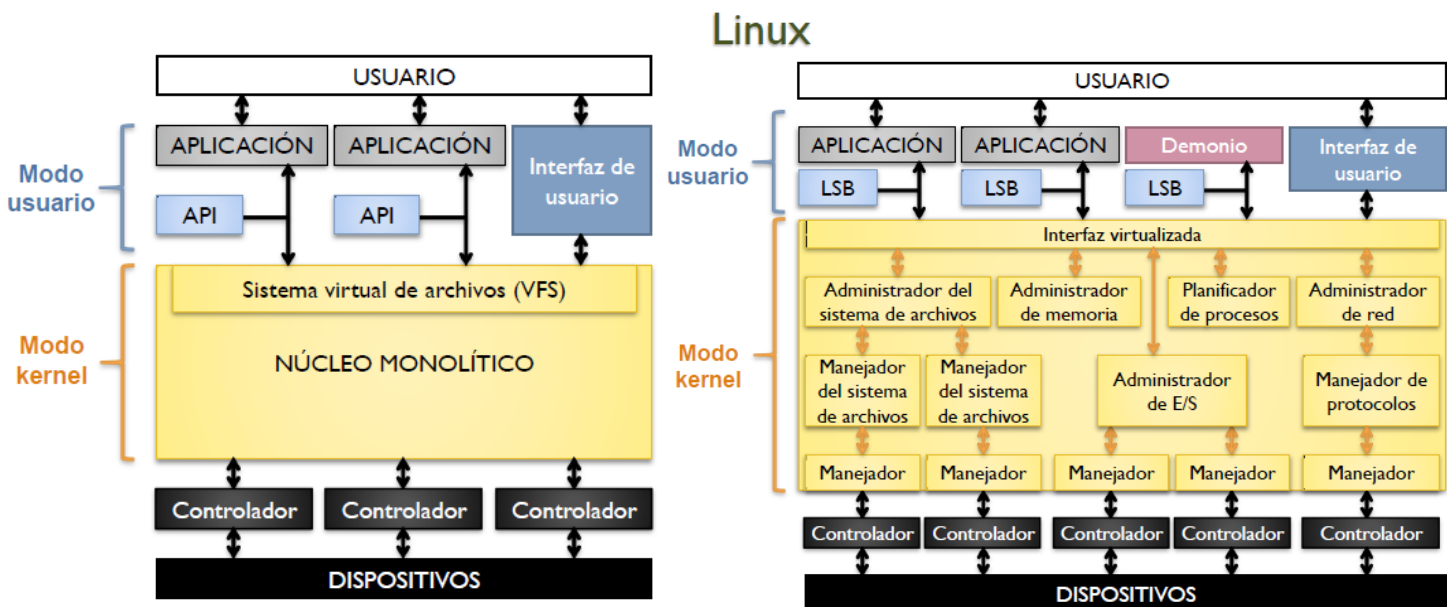
Nota: Al principio no se protegía la memoria, luego se empezó a proteger al igual que otros periféricos. Siempre que queremos hacer una escritura a un registro, vamos a tener que solicitárselo al sistema, y si se puede nos dará permiso. Igual que para hacerlo tiene que ser administrador o tener esos privilegios (por

ejemplo la hora, solo lo puede cambiar un usuario administrador, pero el control de la energía no lo podrá hacer nadie.) Modo kernel o núcleo. Los únicos que van a trabajar en este modo sólo podrán los controladores y el núcleo. Un usuario nunca podrá usar el modo kernel.

Arquitecturas del núcleo

• Núcleo monolítico

- Está compuesto de un único módulo muy complejo que integra todas las funciones del sistema operativo.
- La comunicación entre sus rutinas internas es total lo que hace más fluido su ejecución.
- Por otra parte un error puede propagarse a todas las funciones.
- Estos núcleos ofrecen a las aplicaciones de usuario una interfaz que virtualiza la mayoría de los aspectos (como el de un sistema de archivos virtual), esto otorga al sistema una cierta estabilidad.
- Cualquier módulo adicional del sistema operativo se ejecuta en el espacio de memoria del núcleo, y por tanto en **modo kernel**.
- Cada vez que el núcleo se actualice o se modifique es necesario recompilarlo entero. Esto hace muy complicado el diseño e instalación de nuevos manejadores (drivers).

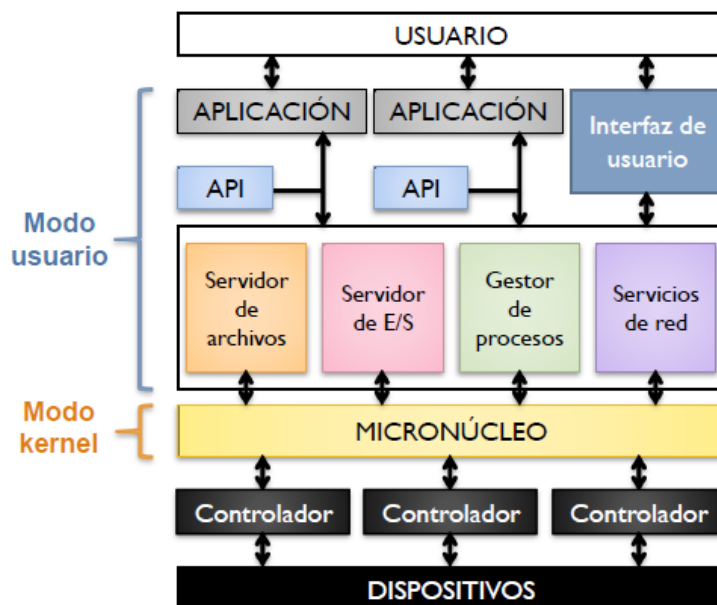


Nota: LSB: librería o api Linux. Para abrir el archivo llamada: open. La escritura se hace mediante una llamada: write. El posicionamiento es: seek. Y para leerlo con una llamada: read. Para finalizar la acción: close. Es muy eficiente, al ser un mismo proceso todo se ejecuta en el mismo hilo de ejecución. Tiene menos invulnerabilidad, por el hecho de haber muchos programadores trabajando en el código. Al haber menos clientes no renta crear tantos virus. También es más seguro porque su núcleo es mucho más antiguo que Windows. También afecta que Linux es mucho menos permisivo.

- **Micronúcleo o microkernel**

- Está compuesto de un conjunto de rutinas muy básicas con las funciones fundamentales del sistema operativo.
- El resto de funciones se realizan mediante procesos servidores en **modo usuario**.
- La modularización del sistema operativo lo hace muy resistente a errores al quedarse estos aislados en la parte afectada.
- .La comunicación entre las partes es dificultosa lo que hace de estos sistemas menos eficientes que los monolíticos.
- La actualización del sistema es sencilla y los nuevos manejadores fáciles de diseñar e instalar.

Nota: En vez de estar compuesto por muchas cosas, en el micronúcleo tendremos muy pocas cosas. Los servicios críticos los llamaremos servidores, mientras al resto simplemente los llamaremos servicios. Al trabajar en servicios, si alguno falla, los demás seguirán trabajando normalmente. O al menos eso pasa en teoría. En modo kernel puede comunicarse con cualquier otra parte de la memoria o sistema operativo, en cambio en este modo tendremos que pedirle permiso a la única parte que funciona en modo kernel (el núcleo) y que nos administre permisos. La actualización es muy sencilla, porque se puede hacer mientras el resto trabaja, después se vuelva a cargar y todo listo.



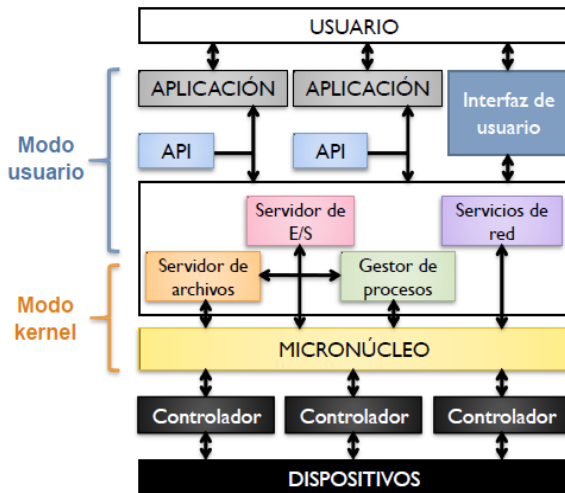
- **Núcleo híbrido**

- Fundamentalmente es muy parecido a un micronúcleo puro, aunque en este caso integra más funcionalidades que éste.
- Vamos a tener pues algunos módulos que se van a ejecutar en **modo kernel** y otros en **modo usuario**.
- Combina los aspectos de los núcleos monolíticos con los de los micronúcleos.
 - La estabilidad del sistema es como la de un micronúcleo, al estar

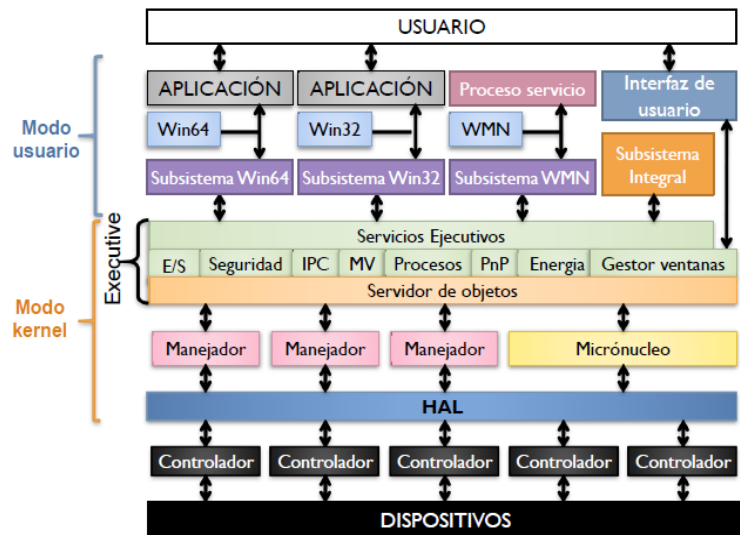
modularizado.

- La comunicación entre las partes es dificultosa sólo en **modo usuario**.
- La actualización del sistema es sencilla y los nuevos manejadores fáciles de diseñar e instalar.

Nota: o Es mucho más utilizado los núcleos híbridos, porque quitamos la mayoría de las desventajas del micrónúcleo y seguimos disfrutando de todas sus ventajas. Si quiero rediseñar algo solo tengo que preocuparme del núcleo individual y de las comunicaciones, pero no hay que preocuparse del resto de núcleos.



Windows NT



Middleware

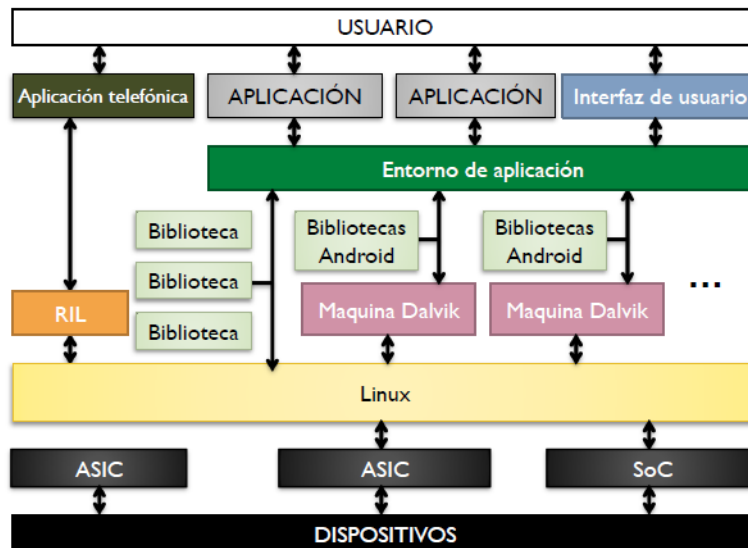
¿Qué es?: Se utiliza una plataforma intermedia entre el núcleo y las aplicaciones, constituyendo una capa de abstracción intermedia. El núcleo de estos sistemas suele ser un núcleo monolítico reducido.

Las aplicaciones se ejecutarán total o en su mayoría mediante un entorno de ejecución (framework). El nuevo nivel de abstracción proporciona una mayor estabilidad. La implementación de aplicaciones se realiza según las especificaciones del entorno de ejecución.

La actualización de las aplicaciones es sencilla. El uso de una máquina virtual y de librerías nativas posibilita la ejecución multiplataforma.

Nota: Ejemplo de framework puede ser Java. Al ser más virtual es menos eficiente, van a aprovechar mal el hardware. Gasta muchos recursos en la virtualización en vez de la tarea que queremos realizar. Por ejemplo Java gasta muchos recursos en el framework.

Android



Nota: Un núcleo de Linux simplificado. El framework va a virtualizar cada pequeña parte de ejecución/aplicación. Por lo que no usa el procesador original, sino el virtual.

Nota: Android no se encarga de gestionar todos los recursos de todas las aplicaciones, or ejemplo los juegos que usamos en un sistema android son ejecutados en una maquina virtual distinta ya que ejecutal algo 3D en android es un despropósito.

En la siguiente tabla encontramos ejemplos de todos los tipos de núcleos que hemos estudiado en este apartado.

Núcleo monolítico	Micronúcleo	Núcleo Híbrido	Middleware
Sistemas UNIX	QNX	Windows XP	Blackberry I0
Distribuciones Linux	BeOS	Windows 7	Samsung Bada
Windows CE	Blackberry OS	Windows 8	LG WebOS
	Symbian	Windows 2003 Server	Android
		Windows 2008 Server	Maemo
		Windows 2012 Server	Moblin
		Windows RT	LiMo
		Windows Mobile	MeeGo
		Windows Phone	Tizen
		Mac OS X	
		iOS	

4) Máquinas virtuales

Máquinas de ejecución nativa

¿Qué son?: Son aplicaciones que emulan el funcionamiento de un ordenador utilizando el hardware y software de un ordenador anfitrión.

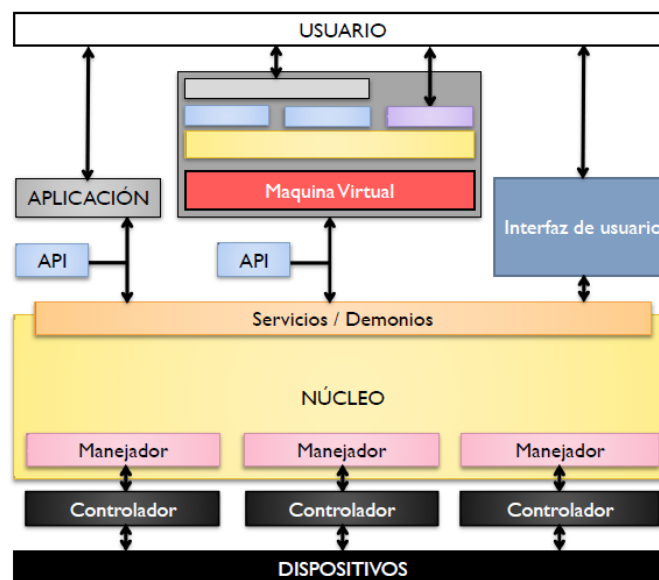
La máquina virtual simula todo el hardware al detalle, aunque éste en realidad no existe. Sobre este hardware simulado se puede instalar cualquier sistema operativo compatible, exactamente de la misma forma que haríamos con una máquina real.

El sistema operativo que da soporte a la máquina virtual se llama igualmente **SO Anfitrión**, mientras que el sistema operativo emulado se le llama **SO Invitado**. El Anfitrión provee de todos los recursos que solicita el Invitado mediante llamadas al sistema a través de la máquina virtual. La máquina virtual jamás puede tener más potencia que la máquina real.

Ejemplos:

- Virtualbox
- VMware
- Virtual PC

Nota: Emula el funcionamiento de un ordenador, con prestaciones virtuales (valga la redundancia). Si el sistema es de 32bit (en teoría) podría instalar un sistema operativo 32bit.



Máquinas virtuales de proceso

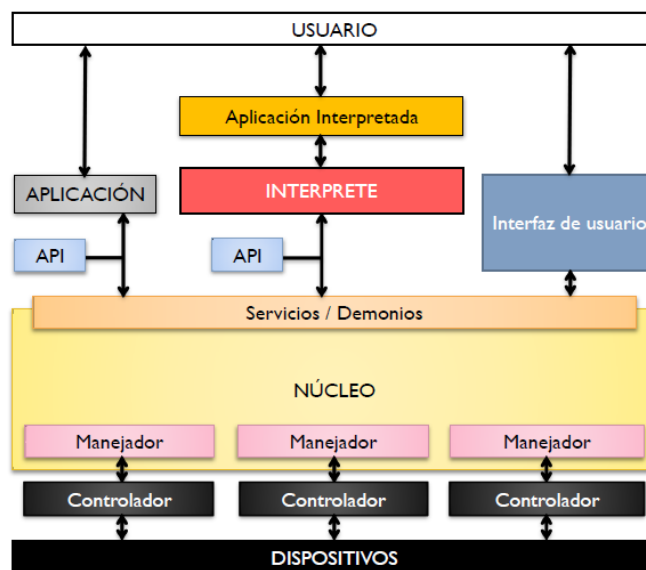
¿Qué?: Son aplicaciones que crean un entorno específico tanto en hardware como en software para un único proceso. La máquina virtual se ejecuta como una aplicación más. También se emula todas las

señales de un ficticio sistema operativo. El hardware emulado es totalmente ficticio.

Tipos:

- **Interpretes** → Trabajar con el código fuente de los programas.
 - Java Virtual Machine (JVM) → Java
 - Common Language Runtime (CLR) → C# , Visual Basic .NET
 - Macromedia Flash Player → SWF
 - Scumm VM → Juegos de MsDOS
-
-
- **Emuladores** → Trabajan directamente con archivos binarios.
 - DOSBox → Ejecutables de MsDOS
-
-
- Las llamadas **capas de compatibilidad** no son maquinas virtuales sino API no nativas.
 - WINE → Ejecutables de MsDOS, Win16, Win32 y Win64
 - Cygwin → Ejecutables POSIX y LSB

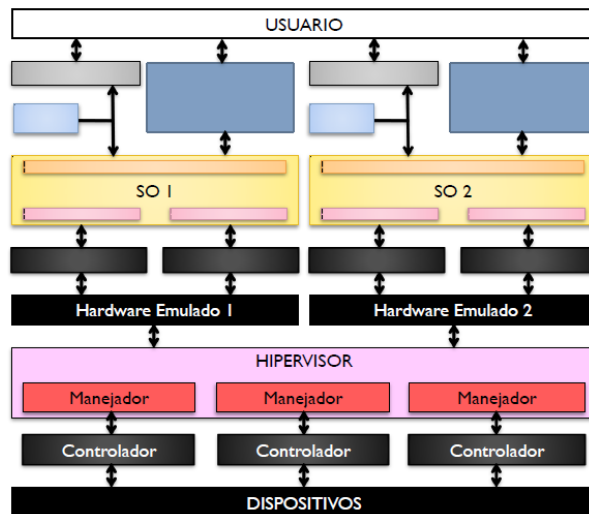
Nota: A diferencia del otro solo crea un proceso para todo. El intérprete de Java trabaja en Risc. Por lo cual eso nos dice que realmente Java es un lenguaje de bajo nivel. Puede emularse en cualquier hardware.
¡Examen! En teoría, es más eficiente una aplicación corriendo en la capa de compatibilidad, pero es más estable corriendo con en el intérprete.



Máquinas virtuales a nivel de sistema operativo (Hipervisores)

¿Qué son?: Son máquinas virtuales que se montan directamente sobre el hardware. Permite instalar varios sistemas operativos de forma paralela. Se establecen así diferentes compartimentos que actúan independientemente.

Este tipo de virtualización se utiliza para SO Multiusuarios (servidores), cuando tienen que proveer servicios a sistemas diferentes. El usuario percibe que en realidad se están ejecutando varios sistemas operativos en sus respectivas máquinas. Un ejemplo máquina virtual Hipervisor: VMware ESX.

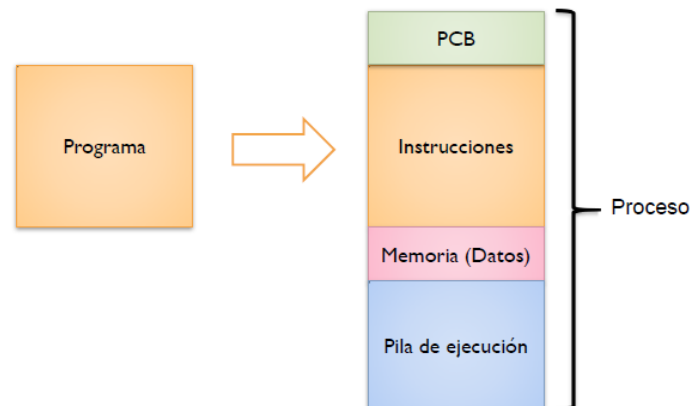


5) Gestión de procesos

¿Qué son?: Los procesos son la instancia en ejecución de un programa.

Los procesos están **compuestos por**:

- La lista ordenada de instrucciones (programa).
- Los recursos de memoria asignados para su ejecución.
 - Memoria de datos (Memoria Estática) → Variables.
 - Pila de ejecución (Memoria Dinámica)
- **PCB** (Bloque de Control de Proceso): Guarda el contexto del proceso en ejecución:
 - **Identificador del proceso (PID)**
 - Contenido de los registros del procesador.
 - Ubicación en memoria.
 - Archivos y otros recursos en uso.
 - Prioridad de ejecución.
 - Directorio de trabajo.
 - Privilegios.



Contexto de un proceso

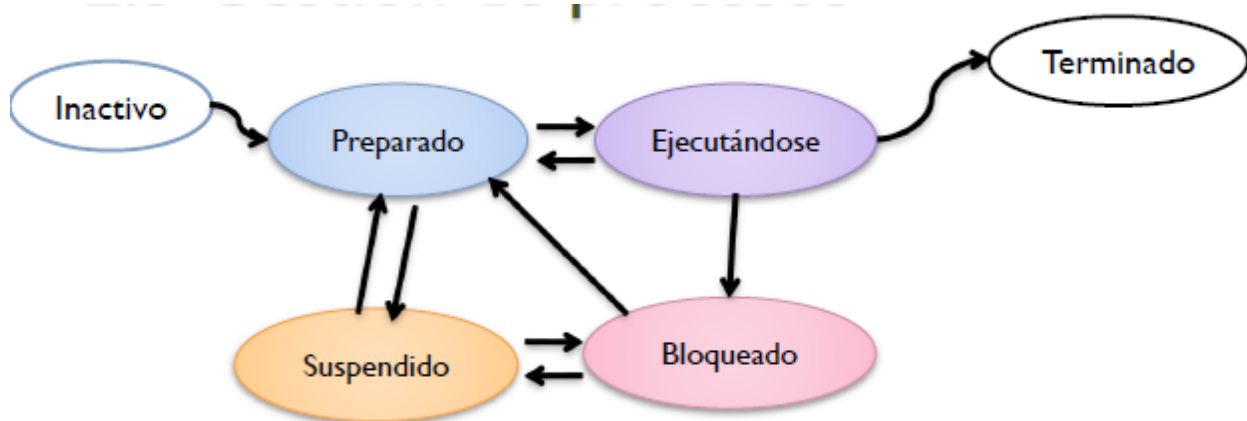
¿Qué es?: Es la situación de los registros de la CPU de un proceso tomando en consideración un momento en concreto.

Establecer y controlar el contexto de un proceso es fundamental para un sistema multiprogramado:

- Cuando se interrumpe un proceso para ejecutar otro, se cede el control al SO, salvando su estado en el PCB.
- Se pasa de modo usuario a modo kernel.
- Se introduce el estado del nuevo proceso en la CPU y se reanuda su ejecución.
- Se vuelve al modo usuario.

Este procedimiento se denomina cambio de contexto.

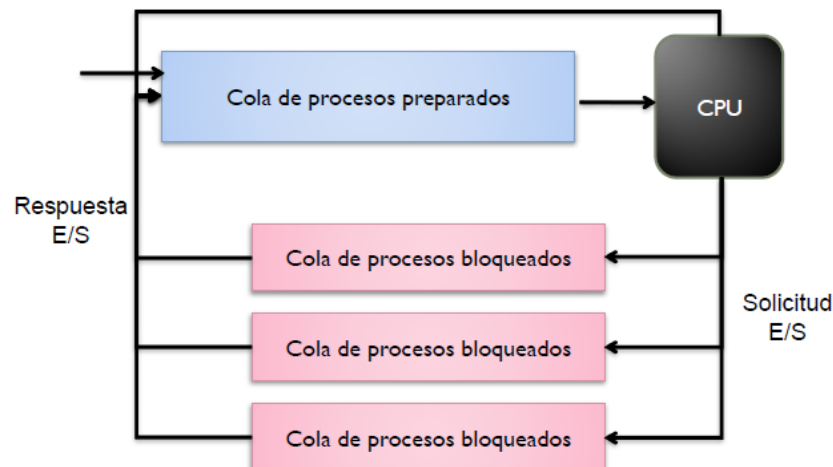
Estado de un proceso



- **Estado inactivo:** El proceso se acaba de crear, tiene los recursos de memoria asignados y se ha registrado en la tabla de procesos. Aún no se le ha asignado ninguna prioridad.
- **Estado preparado:** El proceso se ha puesto en cola y está listo para ejecutarse, esperará su turno en la **cola de procesos preparados**.
- **Estado ejecutándose:** El proceso toma el control de la CPU, se ha cargado desde la **cola de preparados**. El contexto almacenado en el PCB se carga en los registros. Cuando la ejecución cesa, se carga un nuevo proceso desde la cola de preparados.
- **Estado Terminado:** El proceso ha finalizado no se ejecutará más, se liberan todos los recursos asignados y se borra su PCB.
 - Formas de terminar un proceso:
 - Por solicitud interna o del usuario.
 - Cuando ocurre un error en el proceso.
 - Cuando se le mata.
- **Estado Bloqueado:** Un proceso que necesita un recurso o solicita una acción que no está disponible en ese momento. El proceso pasa a una de **cola de procesos bloqueados**. Cuando sí se ha obtenido una respuesta, el proceso vuelve a la **cola de procesos preparados**.
- **Estado Suspendido:** En este estado se hallan los procesos que no pueden competir por la CPU, por lo menos de momento. Este proceso normalmente pasa a la **memoria virtual**.

Nota:

- Entre carga y carga debe ejecutarse un poco el SO. Toma el control de la CPU.
- Cuando decimos que matamos un proceso lo que hacemos es decirle al sistema operativo que cierra ya ese proceso, sin importar los datos que tenía ni si después se necesitaba para algo. Para poder hacer esto tienes que ser administrador.
- Normalmente van a estar bloqueados hasta que el sistema o el usuario lo solicite, por ejemplo los driver, el driver está bloqueado todo el tiempo hasta que el usuario solicita su acción (el driver del ratón está bloqueado mientras no se mueve, y cuando realizamos un movimiento pasa a la cola de preparados. O una consola de comando mientras espera a que nosotros introduzcamos un comando se encuentra bloqueado) *Pregunta de examen: ¿Cuáles de los procesos en un sistema multiprogramado van a estar más tiempo en la cola de preparado? Aquellos que tienen una ejecución continua, por ejemplo un demonio o un antivirus.* Siempre un solo proceso por cada procesador al mismo tiempo (siempre teniendo en cuenta lo del grado de multi o mono programación).
- Se creó cuando la memoria era insuficiente para guardar todos los procesos bloqueados; Lo que hacemos es que cuando están en estado suspendido lo almacenamos en el disco duro. Estos archivos pasan a estar en la memoria virtual del disco duro.

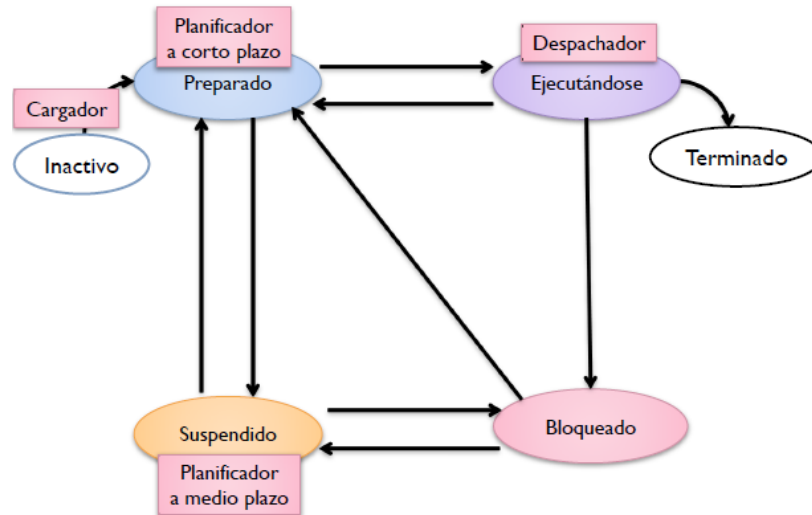


Nota: Cuando se ejecuta un proceso va primero a la cola de procesos preparados, después se decide a dónde se dirige de nuevo, dependiendo si tiene quantum para seguir en preparado o si debe ir a bloqueados o a suspendido.

Módulos Implicados

- **Cargador:** Se encarga de preparar el programa para ser lanzado.
 - Crea el PCB y se le asigna una prioridad.
 - Inserta el proceso en la tabla de procesos.
- **Planificador a corto plazo:** Se encarga de decidir el orden de ejecución de los procesos en la cola de preparados. Se ejecuta unas 50 veces por segundo.
- **Planificador a medio plazo:** Se encarga de decidir que procesos pueden competir por la CPU.
- **Despachador:** Realiza la sustitución del proceso en ejecución por el primero en la cola de preparados o sea. Se ejecuta en tres momentos:

1. Cuando el proceso en ejecución agota su tiempo de CPU.
2. Cuando el proceso en ejecución es interrumpido por una solicitud de un recurso o acción.
3. Cuando el proceso en ejecución realiza una llamada al sistema
4. Cuando el proceso en ejecución termina.



Nota: Intercambiador, lleva los procesos del estado suspendido hasta e preparado

Interrupciones

¿Qué son?: Son señales recibidas por el procesador durante la ejecución de un proceso. Detiene la ejecución actual y devuelve el control al sistema operativo, el cual después de atender la interrupción devolverá la CPU a los procesos. Las interrupciones hacen posible la interactividad en un ordenador. Éstas se gestionan a través del manejador de interrupciones y el SCB.

Tipos:

- **Interrupciones Hardware**
 - **Interrupción de la CPU:** Interrupción síncrona que indica que el tiempo de CPU se ha agotado para el proceso en ejecución. Hace posible la multiprogramación.
 - **Interrupción de respuesta E/S:** Interrupción asíncrona producida por un periférico cuando da una respuesta a una solicitud realizada por un proceso en un momento anterior.
- **Interrupciones Software (Traps)**
 - La produce una llamada al sistema de una aplicación o usuario.
 - Se produce de forma síncrona con la ejecución, producen el estado bloqueado en el proceso.

Nota: *Pregunta examen: ¿Sin un proceso se está ejecutando y se detiene cual es el siguiente proceso en ejecutarse? El sistema operativo.*

El manejador de interrupciones está casi siempre ejecutándose. Los drives se activan cuando se lo manda el manejador.

Excepciones

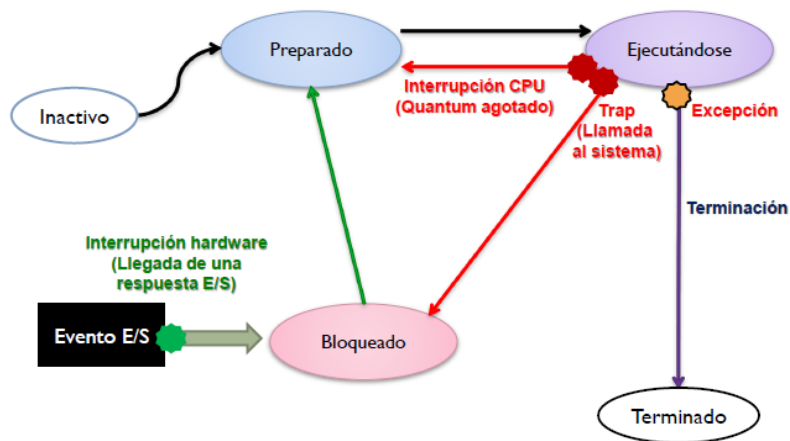
¿Qué es?: La produce una situación anormal en la CPU producida por una instrucción errónea (división por cero, fallo de dirección...). La interrupción es generada por la propia CPU. Se produce de forma **síncrona** con la ejecución. En este caso se llama al gestor de excepciones que junto al SCB tratan de resolver el problema.

Pueden pasar tres cosas (**tipos de error**):

Que el error sea **fatal**: afectan al sistema y este puede incluso bloquearse.

Que el error sea **crítico**: no afectan al SO, sólo al proceso el cual no puede continuar. El SO normalmente decide matar al proceso.

Que el error sea **recuperable**: realizando algunos ajustes, el proceso puede continuar, aunque quizás no lo haga correctamente.



Nota: Hay dos tipos de interrupción

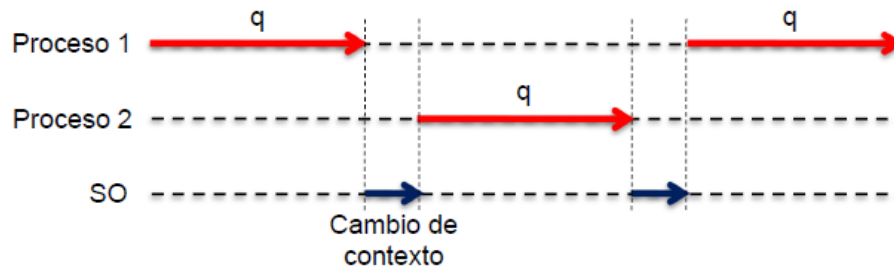
- De software en el procesador llamada al sistema \neq solicita al SO un recurso. El proceso deja de ejecutarse y toma el mando el SO.
- De hardware. En la CPU. Cuando se agota el Quantum (número de ciclos de ejecución que se asigna a un proceso).

Si el quantum de tiempo dura 10 milisegundos, el quantum de proceso es la inversa.

$qt = 10\text{ms (milisegundos)} \parallel q_{\text{ciclo}} = qt \cdot v \parallel qt = t_2 \cdot v = 1\text{GHz} \cdot 10$

El Despachador

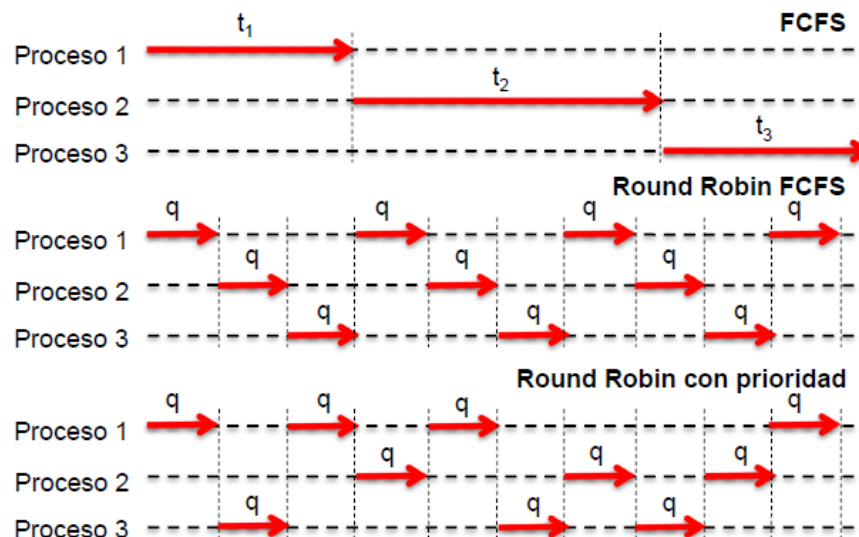
¿Qué es?: Cuando un proceso está ejecutándose, las variables que en ese momento se están utilizando, están cargadas en los registros de la CPU. Cuando el **quantum** de tiempo se agota o se produce una **trap** (Interrupción de software). El despachador se activa y detiene proceso, salva los valores actuales de estos registros en el PCB de ese proceso. El despachador también se encarga de cargar los datos del nuevo proceso en los registros de la CPU. Los sistemas de multiprogramados realizan de 100 a 1000 cambios de contexto por segundo.



El Planificador a Corto Plazo

¿Qué es?: Cuando se activa (lo hace varias veces por segundo) reorganiza el orden de los procesos en la **cola de preparados**. Dependiendo de la complejidad del SO, los algoritmos ordenación que utiliza en planificador a corto plazo **se pueden clasificar en:**

- **No apropiativas:** Una vez que se le asigna la CPU a un proceso, no se le retira hasta que éste finalice o se bloquee. Esto es común en el proceso por lotes.
 - FCFS (First Come First Served). Primero en entrar primero en salir. No hay planificador.
 - SJF (Shortest Job First). El trabajo más corto es el primero.
- **Apropiativas:** Cuando el sistema operativo puede apropiarse de la CPU cuando se considere oportuno. Necesario en la **Multiprogramación**.
 - Round-Robin FCFS. Se van turnando. El primero que entra es el primero en salir. No hay reordenamiento.
 - **Round-Robin con prioridad.** Se usa actualmente. Hay un planificador que reordena según la prioridad base, que depende del tipo de proceso. Prioridad efectiva: antigüedad, peticiones.



Sistema de prioridades

En el momento en el que es creado un proceso, se le asigna un número que representa su prioridad base. Muchas veces este número irá cambiando dinámicamente debido dependiendo del entorno y del tiempo que ha estado en ejecución el proceso.

En los sistemas **Windows NT** vamos a tener prioridades del 0 al 31:

- 0 → proceso suspendido
- 1-15 → aplicaciones de uso compartido (dinámicas)
- 4 → "Debajo de los normal"
- 7 → "Normal"
- 9 → "Mas de lo normal"
- 13 → "Alta"
- 16-31 → aplicaciones de tiempo real (fijas)
- 24 → "Tiempo real"

En los sistemas tipo **Unix** vamos a tener prioridades del 0 al 159:

- 0 - 59 → aplicaciones de uso compartido (dinámicas)
- 60-99 → aplicaciones kernel (procesos bloqueados)
- 100-159 → aplicaciones de tiempo real (fijas)

En estos sistemas, el valor nice le indica al SO entre que valores puede cambiar la prioridad dinámicamente. Por defecto este valor es cero, pero se puede cambiar entre -20 y 19 puntos de prioridad.

Grado de multiprogramación

¿Qué es?: Es el número de procesos que se están ejecutando simultáneamente. Por lo general, el uso de la CPU aumenta con el grado de multiprogramación. La multiprogramación por lo general implica un potencial desaprovechamiento de los recursos de sistema.

Un sistema con un GM bajo se comporta como un sistema monoprogramado. Pero el aprovechamiento de los recursos del sistema es mayor. Para aumentar el GM debemos reducir la duración el quantum de tiempo. Si en quantum es demasiado bajo, el SO se ejecutará en total más tiempo que los procesos de usuario.

Nota: Procesos que están en la cola de preparados. Siempre va a ver al menos un proceso ejecutándose, por lo que siempre hay un "proceso inactivo de sistema", para ir intercambiándose, para no tener que parar el grado de multiprogramación, cuando solo hay un proceso. Dependiendo del número de procesos que se están ejecutando en ese momento y de la duración del quantum.

- Si aumentas la potencia del microprocesador disminuye el grado de multiprogramación los procesos se ejecutan más rápido.
- Si aumentamos el quantum, disminuye el grado de multiprogramación, por lo que se aprovecha mejor la CPU.
- Si disminuye el quantum, aumenta el grado de multiprogramación, por lo que se desaprovecha mejor la CPU.

Técnicas de implementación de procesos

Bifurcación

¿Qué es?: Cuando un proceso lo requiera, este puede efectuar una llamada al sistema para solicitar

una instancia de si mismo. Los dos procesos pasan a tener una relación de dependencia **Padre-Hijo**. Esto se efectúa para realizar una tarea en paralelo que puede estar relacionada o no con la del padre. El hijo se crea como una copia idéntica de su padre, sólo se diferencia en su PID. Durante la ejecución el padre y el hijo divergirán, pero conservaran su dependencia.

En los SO que se utiliza esta técnica todos los procesos son descendientes de un único proceso.

Sistemas Windows NT → system.exe (PID 4)

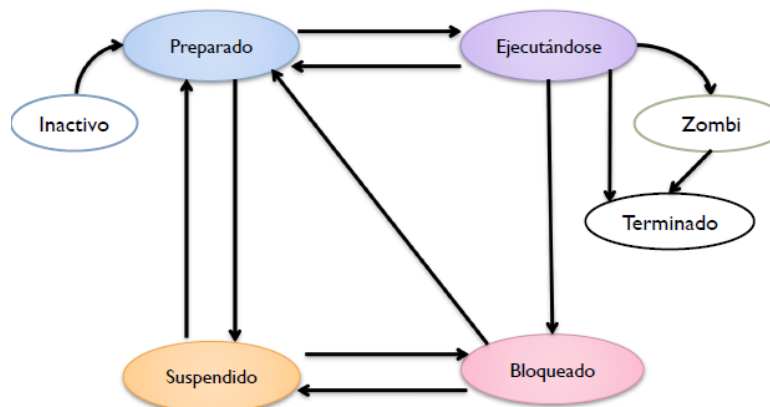
Sistemas basados Unix → init (PID 1)

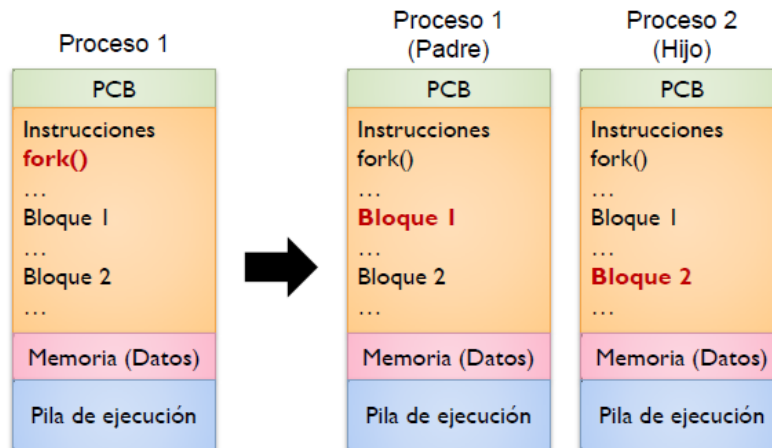
Normalmente el proceso padre se bifurca para obtener algún servicio del hijo. Si hijo termina y tiene datos relevantes a la ejecución de su padre, lo normal es que hasta que este ultimo termine, el hijo espere en estado **Zombi**.

Si padre es el que termina antes que su hijo, para no convertirse en Zombi otro proceso pasara a ser su padre.

- Servicio de sistema Windows NT → services.exe
- Proceso de usuario Windows NT → explorer.exe
- Proceso tipo Unix → init

Si un padre es matado o se cierra anormalmente, todos sus hijos mueren con él.





En LSB fork devuelve el PID del proceso hijo en caso de tratarse del padre y en caso de tratarse del hijo, devuelve cero.

```
switch (fork())
{
case -1:

    Código en caso de error

break;
case 0:

    Código del proceso hijo

break;
default:

    Código del proceso padre

}
```

Hilo de ejecución

¿Qué es?: El proceso principal posee una serie de recursos asignados y una serie de **hilos o hebras** (thread) que lo forman. Cada hilo constituye ahora una unidad de ejecución que se ejecuta concurrentemente con los demás hilos del proceso madre.

Durante la ejecución del proceso, cada hilo realiza una tarea diferente pero orientada a un mismo fin.

Todos los hilos comparten los recursos de memoria. El uso de hilos mejora el rendimiento ya que:

- La creación, manejo y terminación de un hilo es más rápido que la de un proceso tradicional.
- Los hilos se comunican entre más fácilmente que los procesos tradicionales (entre hilos no hay protección de memoria).

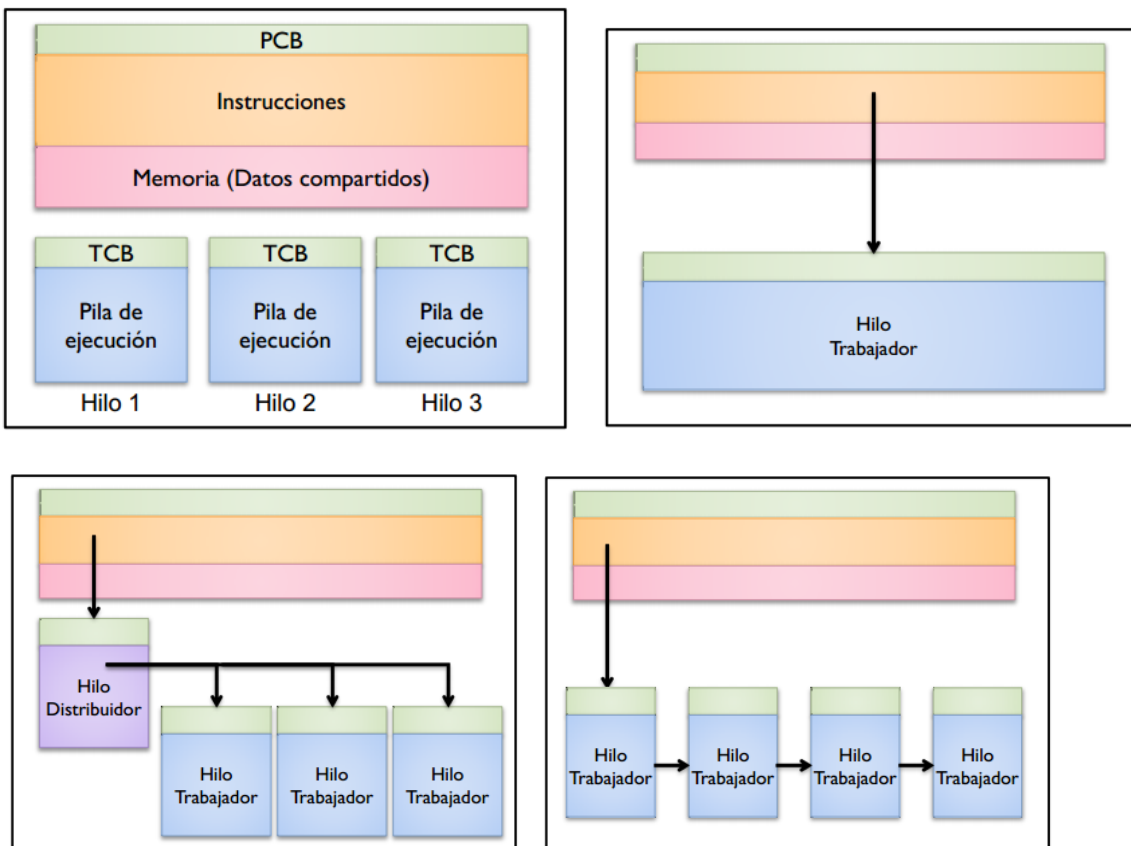
- La ejecución paralela evita retrasos debido a operaciones de E/S.

Nota: Todos sus hijos tienen que estar bloqueados, para eliminar a la madre.

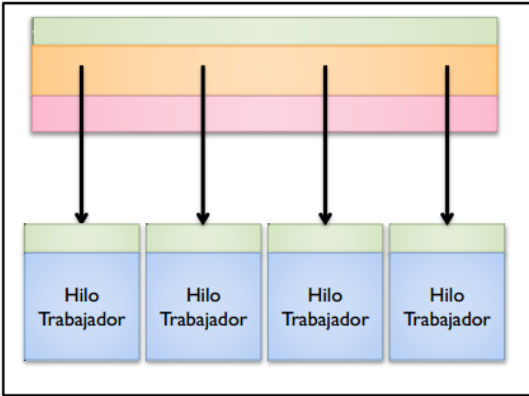
Estado del proceso madre:

- Si algún hilo está en ejecución, el proceso madre estará en ejecución.
- Si ningún hilo de un proceso está en ejecución, pero hay alguno en estado preparado, el proceso madre está en estado preparado.
- Si todos los hilos se bloquean, el proceso madre pasa al estado bloqueado.
- Si el SO decide suspender un proceso, suspende todos sus hilos con él.
- Un proceso se termina cuando terminan todos sus hilos.

Proceso Madre:



Nota: Cada hebra se ocupa de una función. y el distribuidor recibe los datos.



En lenguaje C, la biblioteca con las funciones de creación de hilos están en **pthread.h**

En Java, podemos manipular hilos con la clase **java.lang.Thread**

Crear hilos -->	public void run()
Asignar prioridad -->	public void setPriority(int P)
Iniciar el hilo -->	public void start() (llama a run())
Detener el hilo -->	public void suspend()
Reanudar hilos -->	public void resume()
Consultar si vive -->	public boolean isAlive()
Obtener etiqueta -->	public String getName()
Asignar etiqueta -->	public void setName(String etiq)

```

class HiloEjem extends Thread {
    public HiloEjem(String str) {
        super(str);
    }
    public void run(){
        for (int i = 0; i < 10; i++) {
            System.out.println(i + " " + getName());
            try {
                sleep((int)(Math.random() * 1000));
            } catch (InterruptedException e) {}
        }
        System.out.println("Final: " + getName());
    }
}

class Prueba {
    public static void main(String[] args) {
        new HiloEjem("Marco").start();
        new HiloEjem("Polo").start();
    }
}

```

0 Marco

0 Polo
1 Polo
1 Marco
2 Marco
2 Polo
3 Polo
3 Marco
4 Marco
4 Polo
5 Marco
5 Polo
6 Polo
6 Marco
7 Marco
7 Polo
8 Polo
9 Polo
8 Marco
Final: Polo
9 Marco
Final: Marco

6) Gestión de memoria

¿Qué es? Esta gestión la realiza el **administrador de memoria**.

Este módulo del núcleo tiene los siguientes cometidos:

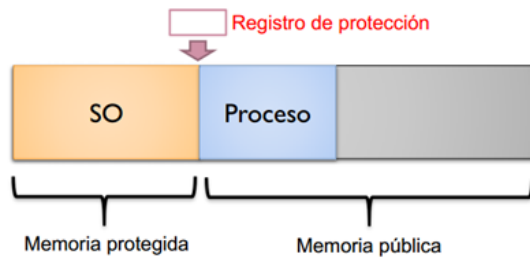
- Guarda las direcciones de memoria de las regiones que están en uso y de las que no.
- Asigna espacio de memoria de los nuevos procesos.
- Libera la memoria de los procesos terminados.
- Protección de los espacios de memoria.

Gestión en los sistemas **monoprogramados**

La memoria se dividía en dos regiones.

- Una región protegida para el sistema operativo.
- Una región pública para ubicar un único proceso.

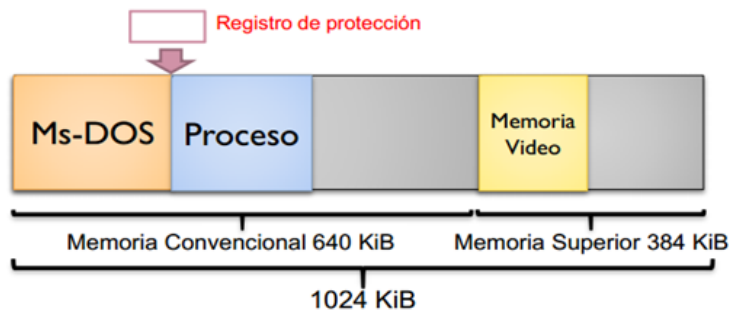
Se utiliza un registro para guardar la dirección de memoria que divide las dos regiones.



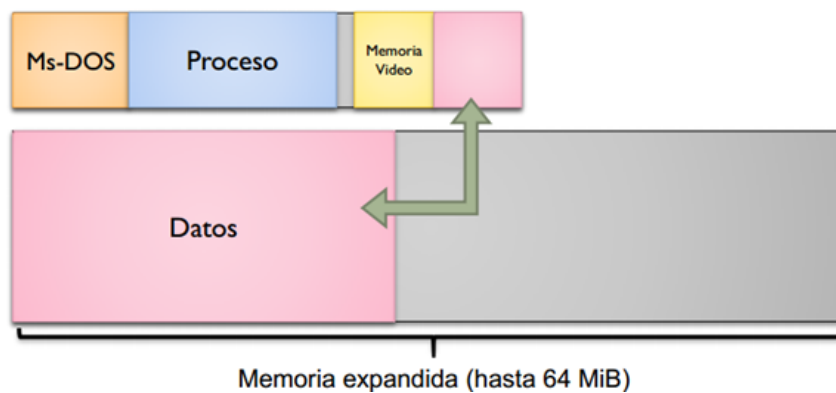
Gestión de memoria en Ms-DOS

Ms-DOS solo puede direccionar 220 palabras de 8 bits (1 MiB):

- Memoria convencional (640 KiB)
 - Núcleo del Ms-DOS, drivers y entorno de texto (comman.com)
 - Aplicaciones.
- Memoria superior UMB (384 KiB)
 - Memoria de Video (128 KiB)
 - Espacio no utilizado (256 KiB)



- Memoria expandida EMS (hasta 64 MiB)
 - Necesita un driver adicional para poder utilizarla (EMM384.EXE)
 - No se pueden ejecutar instrucciones contenidas en la EMS, antes se deben de trasvasar a la memoria de UMB.



- Memoria extendida XMS (hasta 4 GiB)
 - La evolución de los procesadores permitieron eventualmente el aprovechamiento de

la memoria más allá de los 1024 KiB

- Necesita un conjunto de drivers de memoria especiales (RAMDRIVE.SYS)
- La memoria XMS permite al Ms-DOS funcionar en modo multiprogramado →

Primeras versiones de Windows.



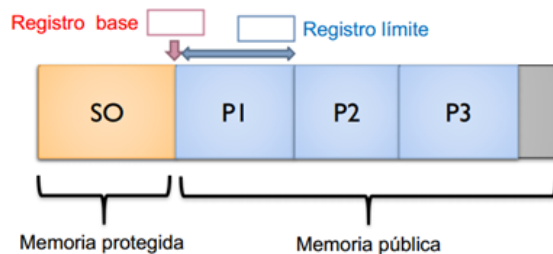
Gestión en los sistemas multiprogramados:

La memoria puede contener varios procesos.

Se utilizan dos registros de protección para limitar el acceso a memoria.

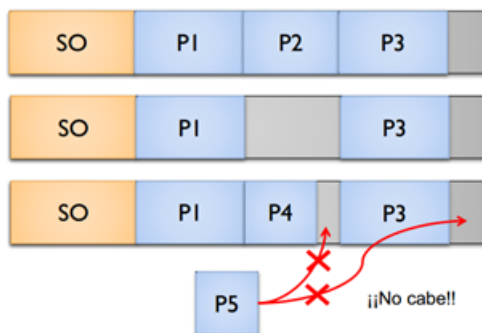
- Registro base: marca el inicio de una partición de memoria.
- Registro límite: indica el tamaño de la partición.

Los registros de protección se guardan en el **PCB** en un cambio de contexto.



Aparición de huecos (Fragmentación externa):

Ocurre cuando se libera memoria de los procesos y los huecos resultantes no son aprovechables.



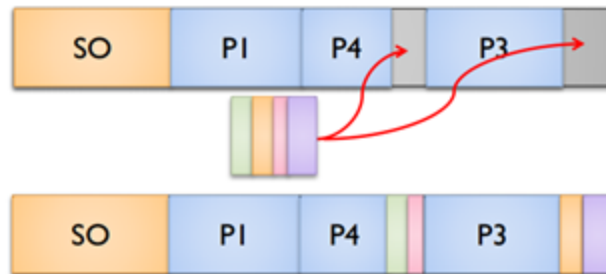
Soluciones a la fragmentación externa:

Segmentación:

Dividir los procesos en segmentos de sus partes fundamentales:

- Hebras.
- Instrucciones, memoria de datos, pila...
- Subrutinas, subespecies de memoria...

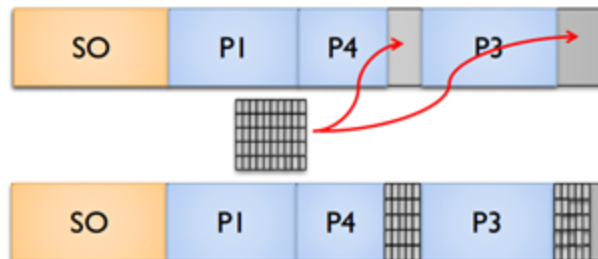
No lo soluciona del todo, los segmentos originan una fragmentación más pequeña. La solución es la paginación



Paginación:

¿Cómo funciona? Se dividen los procesos en porciones iguales de minúsculo tamaño llamadas páginas. Las páginas corrigen completamente el problema de la fragmentación externa porque trabajamos con el mismo tamaño siempre, aquí solo es colocar por lo que no tendremos problemas de tamaño.

Nota: la memoria se divide en marco de página y los procesos en páginas. las páginas van en los marcos de páginas. la ultima pagina del proceso es probable que no esté completa. supongamos que libros de agua son paginas y tengo botellas de un litro y yo divido esa cantidad de libros en litros, supongamos que tengo 3 litros y medio y voy a llenar tres botellas pero tengo que llenar una cuarta hasta la mitad, entonces estoy desaprovechando media botella, este espacio se pierde posteriormente no será ocupado por nada, esto es la fragmentación interna, difícil de solucionar.



En muchos casos, las paginas no se utilizan completamente (la última páginas del proceso) parece entonces un nuevo problema: la **fragmentación interna**.

nota: los datos cuando se relacionan entre sí siempre es bueno que esten juntos, además las lecturas siempre se van a hacer de pagina en pagina y siempre va a ser mejor, sin embargo con un sistema de paginación no se garantiza que siempre estén contiguos a la hora de hacer procesos será muy borroso y por eso surge la segmentacion aginada.

Segmentación paginada:

¿En qué consiste? Combina las técnicas de segmentación y paginación.

Esta técnica es la que se utiliza en los sistemas operativos actuales.

La **compactación o desfragmentación de memoria** es una técnica adicional que alivia la fragmentación externa. Es muy costosa por lo que no se puede realizar con frecuencia.

Nota: es la técnica que usamos actualmente, esto nunca lo vamos a ver de forma directa. consiste en redistribuir la memoria para eliminar los huecos. se puede combinar la segmentación pero no es muy útil, es como un engaño.

Soluciones a la **fragmentación interna**:

Segmentación:

Disminuir el tamaño de las páginas:

- Podemos reducir la fragmentación interna a niveles prácticamente nulos.
- Cuanto más reducimos el tamaño de página, menor es la eficiencia de la gestión de memoria.

Intercambio (swapping):

¿En qué consiste? Consiste en trasladar temporalmente los procesos suspendidos a una unidad de almacenamiento secundario. (cuando entra en suspensión un proceso es lanzado a la unidad de almacenamiento)

La unidad de almacenamiento debe ser un dispositivo rápido con espacio para albergar las imágenes de memoria de los procesos (un disco duro es lo idóneo)

Es el planificador a medio plazo quien se encarga de decidir qué procesos se van a suspender e intercambiar

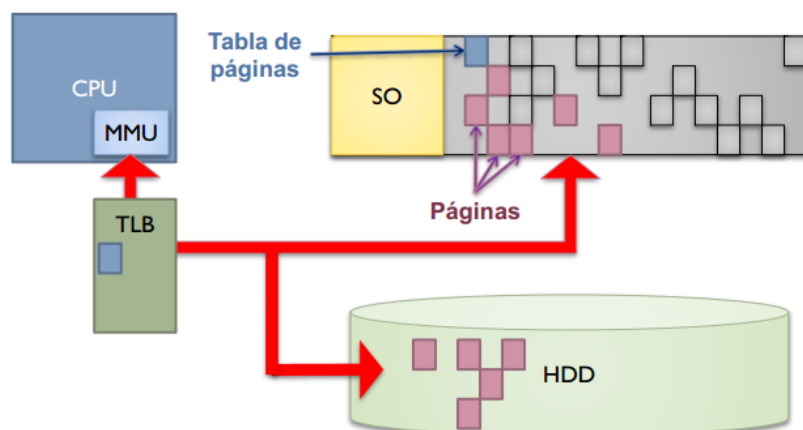
El **intercambiador** va a ser el módulo del SO encargado de gestionar el espacio de intercambio.

Implementación de la memoria virtual en un sistema de memoria paginada:

Cada proceso dispondrá de una **tabla de páginas**. Con ella podremos llevar el control de las páginas que están intercambiadas y su localización exacta.

Se necesita hardware especial que realice las conversiones de las direcciones de memoria:

- Unidad de Gestión de Memoria (MMU)
- Buffer de Traducción Adelantada (TLB)



Nota: una pagina va a estar dedicada a la tabla de procesos (pag azul) y va a a apuntar donde debe ir todas las páginas, vamos a ir moviendo las paginas que menos usos tiene aunq van a tener parámetros de vida, se mueven al disco duro para ello no se usa la cpu se usa la mmu.

Examen: ¿cual son las paginas a las que se accede con más frecuencia? las tablas de páginas porque es lo primero que hay que leer primero y por eso el buffer va a estar lleno de tablas de páginas.

Intercambio Estático (Linux)

- Se crea un espacio de memoria fijo para ubicar los procesos intercambiados por el sistema.
- De su tamaño dependerá el número de procesos intercambiados.
- Los accesos son rápidos y se suele ahorrar memoria.
- Linux utiliza una partición (swap) especial para este cometido.

Intercambio Dinámico (Windows)

- Se crea un espacio de memoria de tamaño variable para cubrir las necesidades de intercambio de todos los procesos.
- No impone límites al número de procesos intercambiados.
- Los accesos son más lentos.
- Se desperdicia más memoria.
- Windows utiliza un archivo de paginación (pagefile.sys)

Hiperpaginación (thrashing) :

Se origina cuando se lanzan demasiados procesos y el sistema no tiene suficiente memoria RAM para manejarlos.

El sistema operativo trata de solucionar el problema intercambiando sus páginas a la memoria virtual.

Se llega un punto en el que las páginas intercambiadas son de uso frecuente.

El sistema operativo tiene que volver a cargarlas casi inmediatamente.

El sistema termina gastando casi todo los recursos en el intercambio... y se colapsa

Política Unused RAM is wasted RAM

- En un sistema tradicional multiprogramado sin esta política, (como Windows XP), el sistema intenta agilizar liberando al máximo la memoria principal, o sea, va a liberar lo más rápido posible los procesos terminados, e intercambiará aquellos que no están previstos utilizarlos durante algún tiempo.
- En sistemas basados en Linux y en Windows 7, el gestor de memoria va a ocupar la RAM al casi al 100% con procesos que no están en uso.
- Con ello se asegura la respuesta de procesos que se sospecha que el usuario utilizará próximamente.

Técnica de Recolección de Basura

- En los sistemas Mac OS X y iOS, un módulo específico del sistema operativo se encarga de liberar aquellos procesos que no se usan demasiado.
- La liberación de los recursos utilizados por estos procesos es total. Pero se guarda su estado de ejecución en memoria secundaria.
- Con esta técnica se gana fluidez, pero se pierde rapidez a la hora de solicitar una respuesta a tiempo real.

7) Gestión de archivo

Unidades de asignación :

Sectores físicos (CHS)

- Un sector es la unidad mínima con la que trabaja el disco duro (no es compatible con otras unidades).
- Su tamaño depende de la geometría interna del disco duro y no se puede modificar.
- A cada sector se le asigna tres números: cilindro, cabezal y sector.
- Su tamaño más usual es de 512 B
-

Sectores lógicos (LBA)

- Se utilizan para direccionar sectores en unidades de más de 8 GiB, sin importar el tipo.
- A cada sector se le asigna un número único.
- Números contiguos podrían no corresponder a sectores físicos contiguos.

Bloques o clusters

- Son la unidad mínima de memoria con la que trabaja el sistema operativo.
- Su tamaño por defecto es de 4 KiB aunque se puede modificar en el momento del formateo

- de una unidad de almacenamiento.
- Al igual que con las páginas de la memoria principal, los bloques lógicos experimentan fragmentación interna.
- Se toman bloques lógicos grandes por razones de eficiencia

Archivos

Archivo o Fichero (File)

- Es la colección independiente de datos binarios, agrupados para facilitar su manipulación y almacenaje.
- Dependiendo del tipo de datos almacenados tendremos diferentes tipos de archivos.
- El sistema operativo gestiona los archivos mediante un **sistema de archivos**.

Sistema de Archivos (Filesystem)

- Es un conjunto de normas y procedimientos para manipular los archivos en las unidades de almacenamiento.
- Cada sistema operativo tiene su propio sistema de archivos nativo.
- A veces un sistema operativo puede leer un sistema de archivos de otro sistema operativo.
- Cada uno de los formatos de almacenamiento óptico tiene su propio sistema de archivos, que suele ser compatible con todos los sistemas operativos.

Tipos de archivos:

Archivos regulares: contiene información útil para su aprovechamiento por el usuario.

- Archivos ejecutables: contienen programas, o sea, el código de los procesos asociados.
- Archivos de datos: contienen información aprovechable por los usuarios..

Directorios / Carpetas (folders): contienen información de otros archivos.

Vínculos (links): contienen la ubicación de otros archivos.

Archivos especiales de E/S (Linux): Permiten la comunicación con dispositivos:

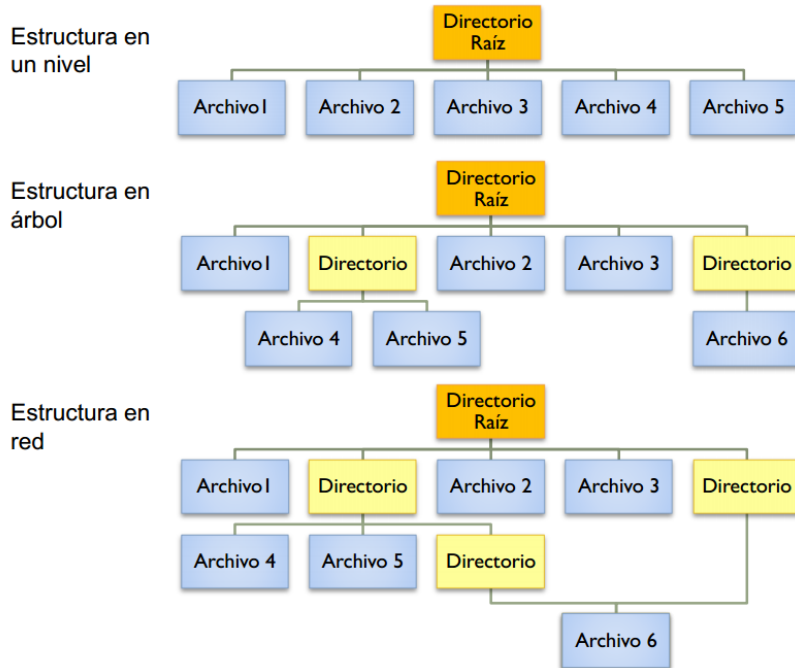
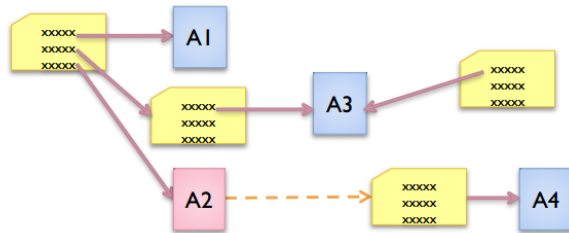
- Archivos especiales de cadenas de caracteres --> Periféricos E/S
- Archivos especiales de bloques --> Unidades de almacenamiento

Estructura de directorios

Directorio: es un archivo con información relativa a la ubicación de otros archivos.

- Se busca una organización de los archivos para facilitar la manipulación desde el punto de vista del usuario. Éste percibe que los archivos están contenidos dentro de los directorios.
- En la realidad los archivos están dispersos en la unidad de almacenamiento.
- La manera a la que se referencia un archivo se le llama enlaces:

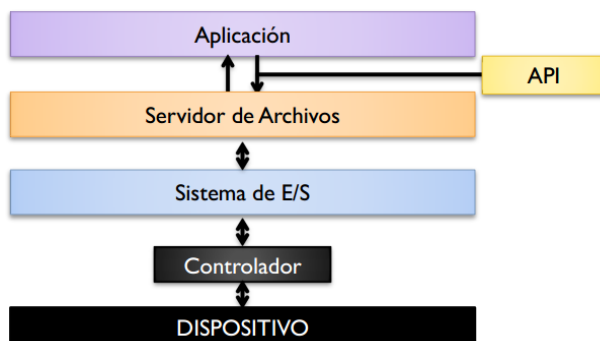
- Enlaces duros, físicos o reales
- Enlaces blandos, lógicos o simbólicos



Servidor de Archivos

Es la parte del sistema operativo que se encarga de la gestión de todos los archivos accesibles.

La principal misión es presentar como una estructura ordenada toda la información contenida en las unidades de almacenamiento.



Estructura de directorios

Operaciones

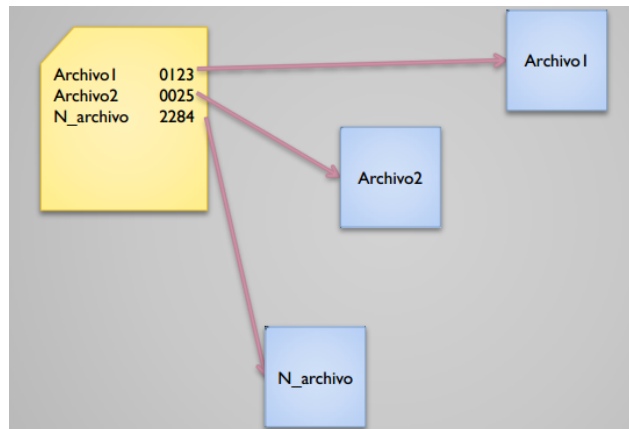
Gestión

- **Crear** un archivo o directorio
- **Borrar** un archivo o directorio
- **Renombrar** un archivo o directorio
- **Mover** (cortar-pegar)
- **Copiar** (copiar-pegar)
- Obtener/ cambiar atributos

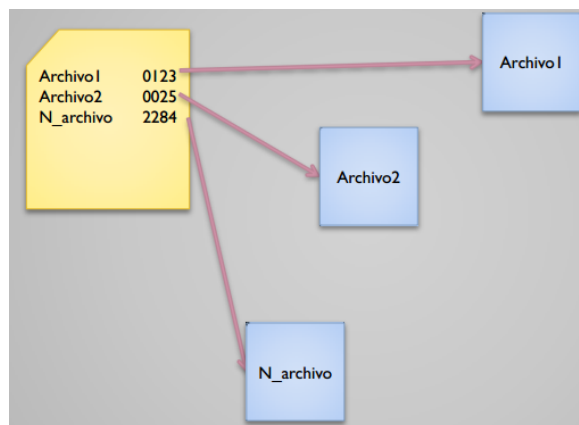
Procesamiento

- **Abrir** un archivo
- **Cerrar** un archivo
- **Leer** un archivo o directorio
- **Modificar** un archivo o directorio

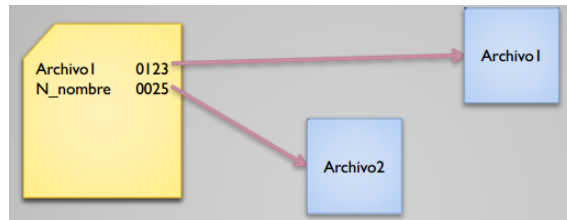
Crear un archivo o directorio



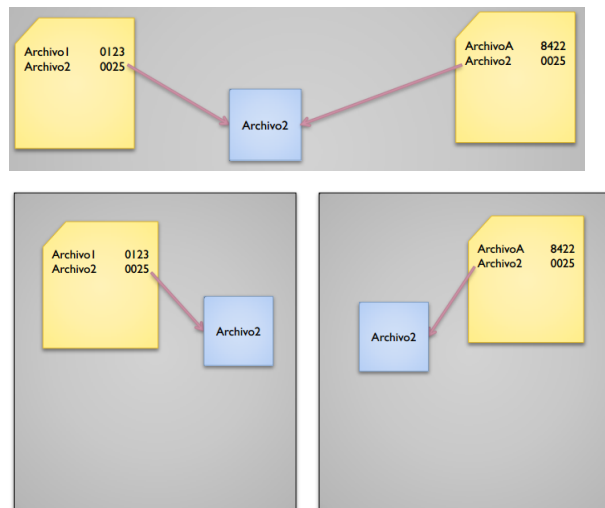
Borrar un archivo o directorio



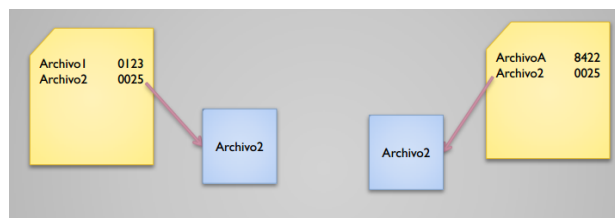
Renombrar un archivo o directorio



Mover un archivo o directorio



Copiar un archivo o directorio



Propiedades o Atributos

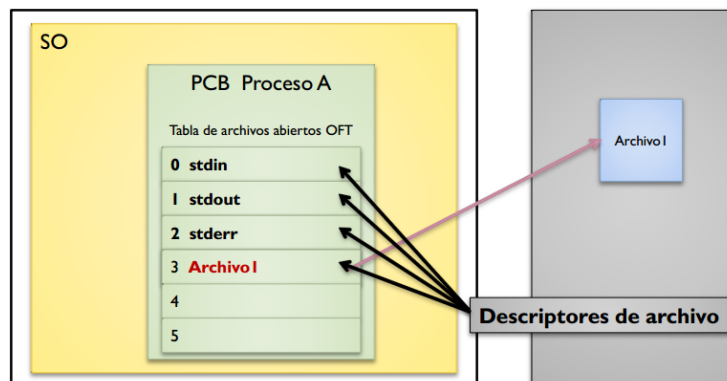
- Nombre del archivo
 - FAT16 (Ms-DOS) --> 8.3 caracteres en mayúscula (SNF)
 - FAT32 / NTFS (Windows) --> 255 caracteres (LNF)
 - HFS+ / ext4 (Linux/Mac) --> 255 caracteres

Caracteres no permitidos: " * / : < > ? \ | (en SNF tampoco espacio)
- Tipo de archivo

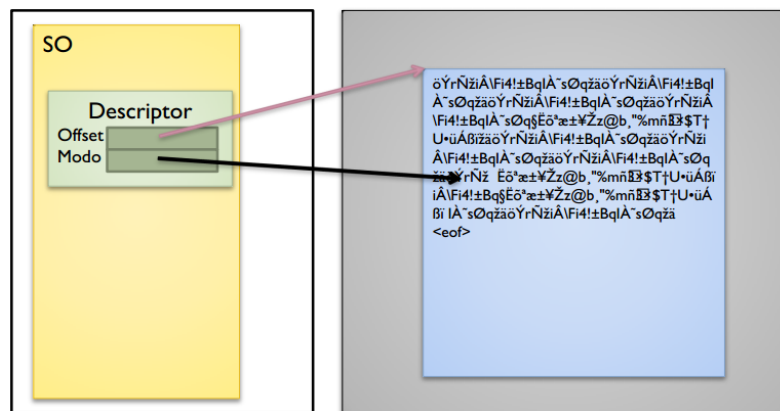
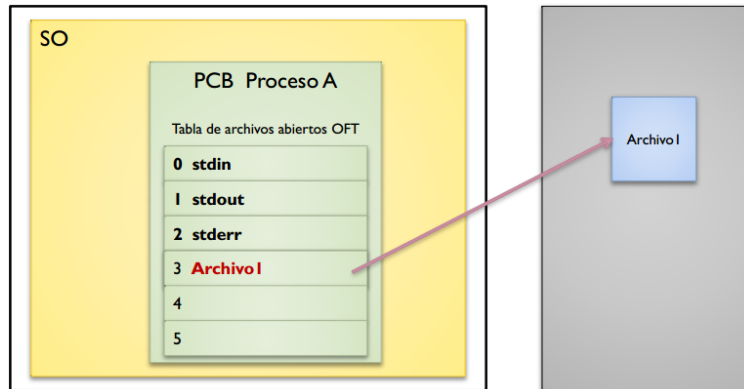
- Tamaño y tamaño en disco
- Permisos
- Propietario y pertenencia a grupo
- Momento de creación (Fecha y hora)
- Momento de la última modificación (Fecha y hora)
- **Bits (Atributos lógicos)**
 - S --> Atributo “archivos de sistema” Marca cuando un archivo es crítico o no. Los archivos de sistema normalmente lo son.
 - H --> Atributo “oculto” Es más fácil dado que se hace desde las propiedades "normales". Para hacer que un archivo sea oculto en Linux sólo tiene que ponerle el nombre empezando por punto (.hola)
 - R --> Atributo “sólo lectura” No se puede borrar con facilidad.
 - Un ejemplo de un archivo que no se puede modificar son lo que vienen grabado en los discos, vienen marcados como solo lectura.
 - A --> Atributo “modificar” Cada vez que accedas se va a marcar este bit. Ahora no es útil, pero antes era útil porque para las copias de seguridad.

Nota: Los metadatos son una parte importante de los archivos. Por ejemplo uno que tenga que ver con un jpg, como color, dimensión, etc. Pero en un word guardará el autor, el número de palabras, etc.

Abrir un archivo (llamada `open()`)



Cerrar un archivo (llamada `close()`)



El sistema operativo gestiona los dispositivos de almacenamiento suponiendo una estructura (lógica) que no tiene porqué coincidir con la estructura física real.

Los discos duros también pueden unirse formando unidades distribuidos o formando sistemas **RAID** y comportándose como una única unidad lógica.

A veces las unidades lógicas pueden referirse a unidades compartidas de una red externa **NAS** o **SAN**. (grandes servidores)

Se puede acceder a las unidades lógicas a través de un acceso especial del sistema llamado **Mi PC** o **Equipo**, según la versión.

Las unidades lógicas se simbolizan con una letra en mayúscula y dos puntos. Cada unidad tiene una letra diferente. Hay un directorio raíz para cada unidad.

- **A:** y **B:** se reservan para las unidades floppy.
- Windows por defecto se instala en la primera unidad de disco duro C:
- Conforme se instalan nuevas unidades, se hace uso de las demás letras en orden alfabético.

En la carpeta **Archivos de programa** se copian, por defecto, los archivos de las aplicaciones cuando estas se instalan.

La carpeta **Windows** contiene los archivos del sistema operativo.

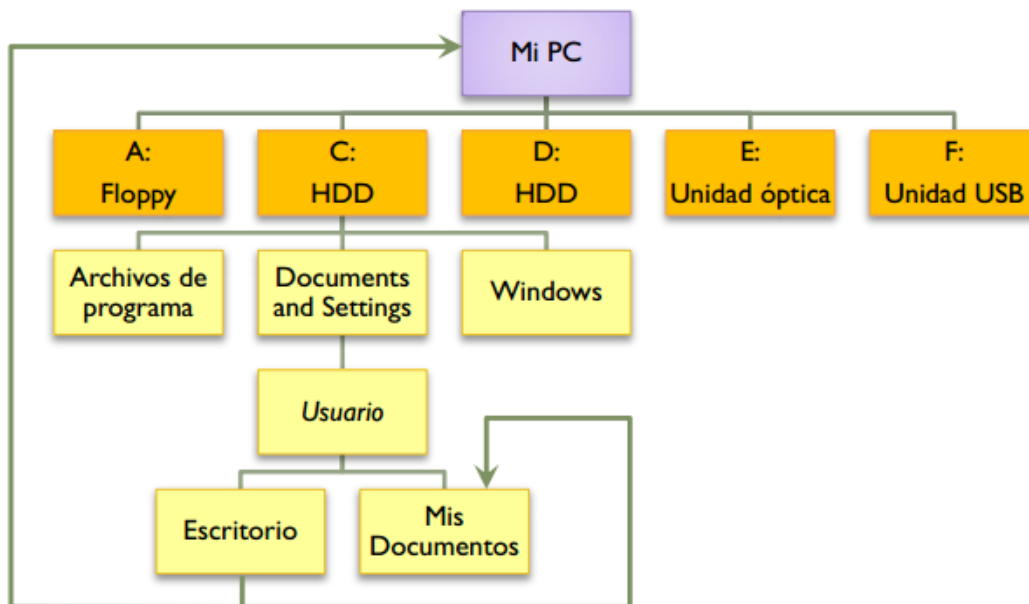
La carpeta **Documents and settings** o **Usuarios** contiene las carpetas personales de los diferentes usuarios.

Por otra parte se puede trabajar con una carpeta como si fuera una unidad lógica (comando **SUBST**).

*Nota examen: ¿Se crea un archivo cuando se formatea? Si el archivo raíz.
Subst virtualiza una carpeta vacía.*

Ejemplo de estructura en Windows XP

- Una unidad floppy.
- Un disco duro con dos particiones,
- Una unidad DVD-R,
- Un pendrive conectado vía USB



*Nota examen: Es mi pc un vínculo duro? No (es un artificio, no un archivo)
El panel del control tampoco es un vínculo duro*

Sistemas basados en Linux

Toda la estructura de archivos parte de un único directorio raíz simbolizado con el carácter **/**.

La carpeta **bin** contiene los ejecutables de los comandos básicos.

La carpeta **boot** contiene los archivos necesarios para arrancar el sistema operativo.

La carpeta **dev** contiene todos el hardware del sistema representado por archivos de **E/S**.

- **fd0 y fd1** son las unidades floppy
- **hda, hdb, hdc, hdd** son las unidades físicas IDE del sistema.
- **hda0, hda1, hda2...** son unidades lógicas, particionadas a partir de hda
- **sda, sdb, sdc...** son unidades físicas de SCSI, SATA o USB
- **sda0, sda1, sda2...** son unidades lógicas, particionadas a partir de sda

La carpeta **etc** contiene los archivos de configuración del sistema operativo y de las aplicaciones.

La carpeta **home** contiene los archivos personales del usuario activo(los otros están ocultos). La carpeta del usuario activo se simboliza como **~**

La carpeta **lib** contiene las bibliotecas necesarias para que los comandos del sistema funcionen.

La carpeta **media** o **mnt** da acceso a la unidades lógicas montadas en el sistema.

La carpeta **sbin** contiene los ejecutables demonios.

La carpeta **usr** se guardan los archivos de las aplicaciones instaladas.

Las carpetas **proc** y **system** son virtuales, las usa el sistema para su ejecución y en realidad no contienen nada.

En sistemas basados en Unix se puede “montar” unidades lógicas en una carpeta (comando **mount**).

Nota examen: Solo los archivos de bloques se pueden montar.

Rutas (Path):

Es una expresión que sirve para localizar un archivo en una estructura de directorios.

- **Ruta absoluta:** Ruta desde el directorio raíz.

Windows C:\Windows\System32\write.exe
Linux /home/carlos/Escritorio/carta

- **Ruta relativa:** Ruta desde otro directorio (directorio de trabajo)

- Carácter . --> Representa el directorio actual
- Carácter .. --> Representa el directorio padre

Windows ..\Imágenes\foto.jpg
Linux ./almacenaje/datos

Sistema de Archivos	Tamaño de archivo máximo	Número máximo de archivos	Tamaño máximo de unidad
FAT12 (Microsoft)	32 MiB	4.077	32 MiB
FAT16 (Microsoft)	2 GiB	65.517	2 GiB
FAT32 (Microsoft)	4 GiB	268.435.437	2 TiB
NTFS (Microsoft)	Tamaño unidad	4.294.967.295	16.777.216 TiB
ext2 (GNU/Linux)	2 TiB	Variable	32 TiB
ext3 (GNU/Linux)	2 TiB	Variable	32 TiB
ext4 (GNU/Linux)	16 TiB	Variable	1.073.741.824 TiB
swap (GNU/Linux)	-	-	-
YAFFS (Android)	512 MiB	262.144	8 GiB
HFS (Apple)	2 GiB	65.517	2 TiB
HFS+ (Apple)	8.388.608 TiB	Sin límite	16.777.216 TiB
CDFS ISO9660 (CD)	2 GiB	65.535	Sin límite
UDF (CD/DVD/BD)	16.384 TiB	Sin límite	Sin límite

Tabla de localización de archivos **FAT32**

Nota: En los Windows de servidor antiguos y en los Windows 2000 y en los NT, no se usaba FAT32, sino NTFS.

Es el sistema de archivos nativo de los antiguos Windows.

Al formatearse con este sistema, se reserva una sección del disco al comienzo de la unidad para la FAT.

Sencilla y económica en memoria.

La tabla utiliza entradas de 32 bits para gestionar los bloques de los archivos, pero sólo 28 bits se utilizan en realidad.

Los archivos se dispersan con facilidad (disminuyendo la eficiencia) con lo que es necesario desfragmentar la unidad con frecuencia. El cabezal tiene que hacer más búsquedas, por lo que es

menos eficiente. El NTFS también se dispersa pero no es tan problemático.

Muy poco resistente a los errores. Por lo que el sistema de archivos está duplicado como copia de seguridad. Al ser tan poco resistente, por ejemplo si había un corte de luz lo más seguro es que se dañase el sistema operativo

No almacena información sobre permisos de usuario.

Recomendada para unidades de almacenamiento pequeñas (pendrives).

Los nombres de archivo pueden tener hasta 255 caracteres (LNF). En el nombre se incluye la extensión.

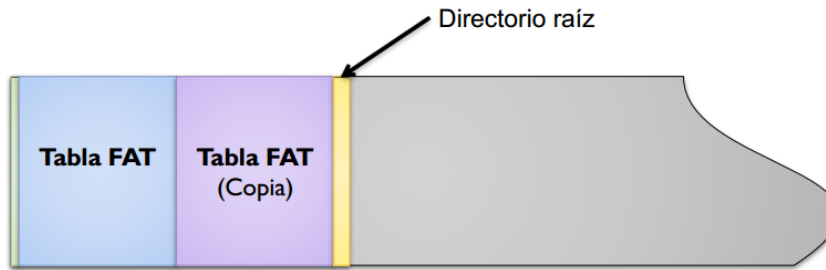
El tipo de archivo lo determina la extensión. En Windows las extensiones pueden tener más de tres caracteres.

Los demás atributos están contenidos en el directorio. Se almacenan en el directorio porque es información del sistema operativo, ya que así es más fácil de acceder a estos.

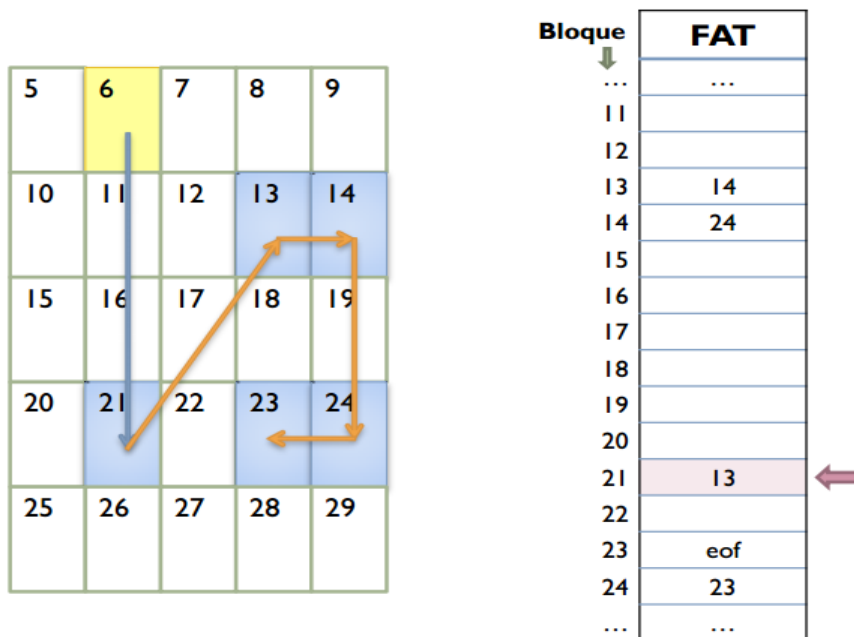
.	-	1 ^{er} bloque
..	-	1 ^{er} bloque
Nombre (255 caracteres)	Atributos	1 ^{er} bloque

Estructura

- Región reservada
- Tabla FAT
- Copia de la tabla FAT
- Bloques de datos
 - Directorio raíz de la unidad.



Nota: Se reserva una pequeña región del disco. Después tiene una Tabla FAT, que será de poco espacio. Siempre se le creará una copia de seguridad por los errores. Aunque se almacenarán los errores. Siempre se crea un Directorio raíz al formatear.



Nota: El sistema de acceso lineal tiene como ventaja ocupa poco espacio. Pero tiene problemas con los accesos y las inserciones.

Número del primer bloque del archivo (directorio). El bloque 6 tiene el nombre del primer bloque que sería 21. En el 21 aparece 13 y así hasta llegar al eof (indicador de fin). El 23 representaría la fragmentación interna.

Examen: En el 6 aparece la marca 6, porque sólo tiene un archivo y termina ahí. Se llega al 6 a partir del directorio raíz. Sistema de listas enlazadas.

0x00000000: bloque libre
0x00000001: bloque reservado
0x00000002 – 0x0FFFFFFF: bloque usado
0x0FFFFFF0 – 0x0FFFFFF6: bloques reservados
0x0FFFFFF7: marca de bloque defectuoso
0x0FFFFFF8 – 0x0FFFFFFE: bloques reservados
0x0FFFFFFF: marca **eof**.



4 bit sin uso

Nota: Bloque libre: Cuando aparece un valor 0, si dice que el bloque está libre. Aunque al final de la tabla se reserva un espacio para los valores libres, pero este tiene el valor cero.

Bloque reservado: Se reserva para cosas internas.

Bloque usado: Señala el siguiente archivo.

Sistema de archivos extendido `ext4`

¿Qué es? Es el sistema de archivos nativo de la mayoría de los sistemas basados en Linux actuales. Esta basado en UNIX, pero no es Unix. Es adecuado para grandes Discos.

Consume grandes extensiones de memoria.

Utiliza una estructura basada en árboles-H de datos capaz de almacenar todo tipo de atributos (metadatos de los nodos).

La distribución de los bloques en árbol aumenta la eficiencia en los accesos.

Los bloques contiguos se manejan en grupos (de hasta 2^{15} bloques) llamados *Extents*. Esto aumenta el rendimiento y disminuye la dispersión de los bloques por el disco duro.

Es muy resistente a los errores. Se basa en un sistema transaccional donde las distintas operaciones son atómicas.

Almacena información detallada sobre la seguridad y los permisos de usuario.

Recomendada para unidades de almacenamiento grandes.

Nota: Implementa muchas cosas, como por ejemplo para guardar una gran cantidad de metadatos.

Al formatear una unidad con este sistema, se reserva un número concreto de bloques para la tabla

de i-nodos dependiendo del tamaño de la partición.

Cada i-nodo tiene 128 byte, por lo que cada bloque de la tabla tiene 32 i-nodos.

Inicialmente los i-nodos no están en uso. Cuando se crea un archivo se marca como usado en el mapa de i-nodos. Cuando se borra un archivo, el i-nodo queda libre y se puede reutilizar para otro archivo.

Del i-nodo parte la estructura de todo el archivo.

Sólo los atributos, básicos (usuario, grupo, tamaño, momento de creación y modificación...) se guardan en el i-nodo, el nombre del archivo que permanece en el directorio.

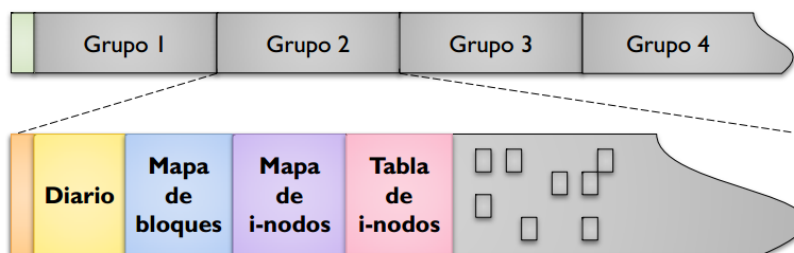
.	n° i-nodo
..	n° i-nodo
Nombre (255 caracteres)	n° i-nodo

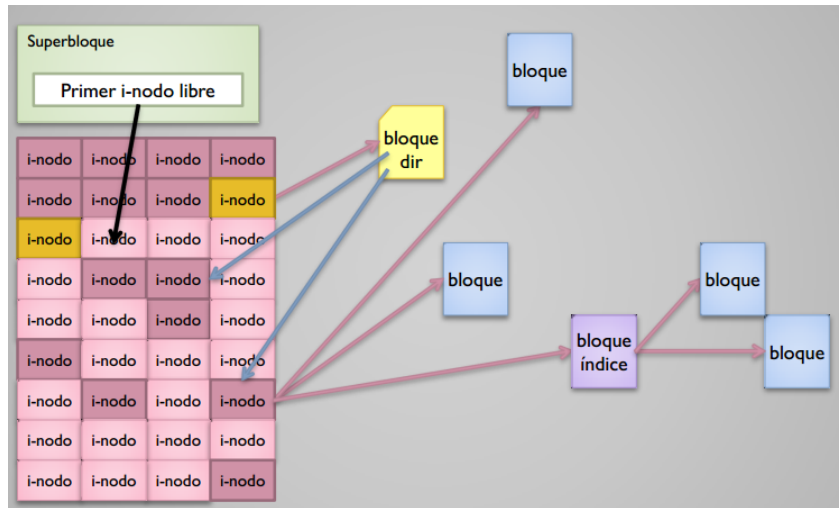
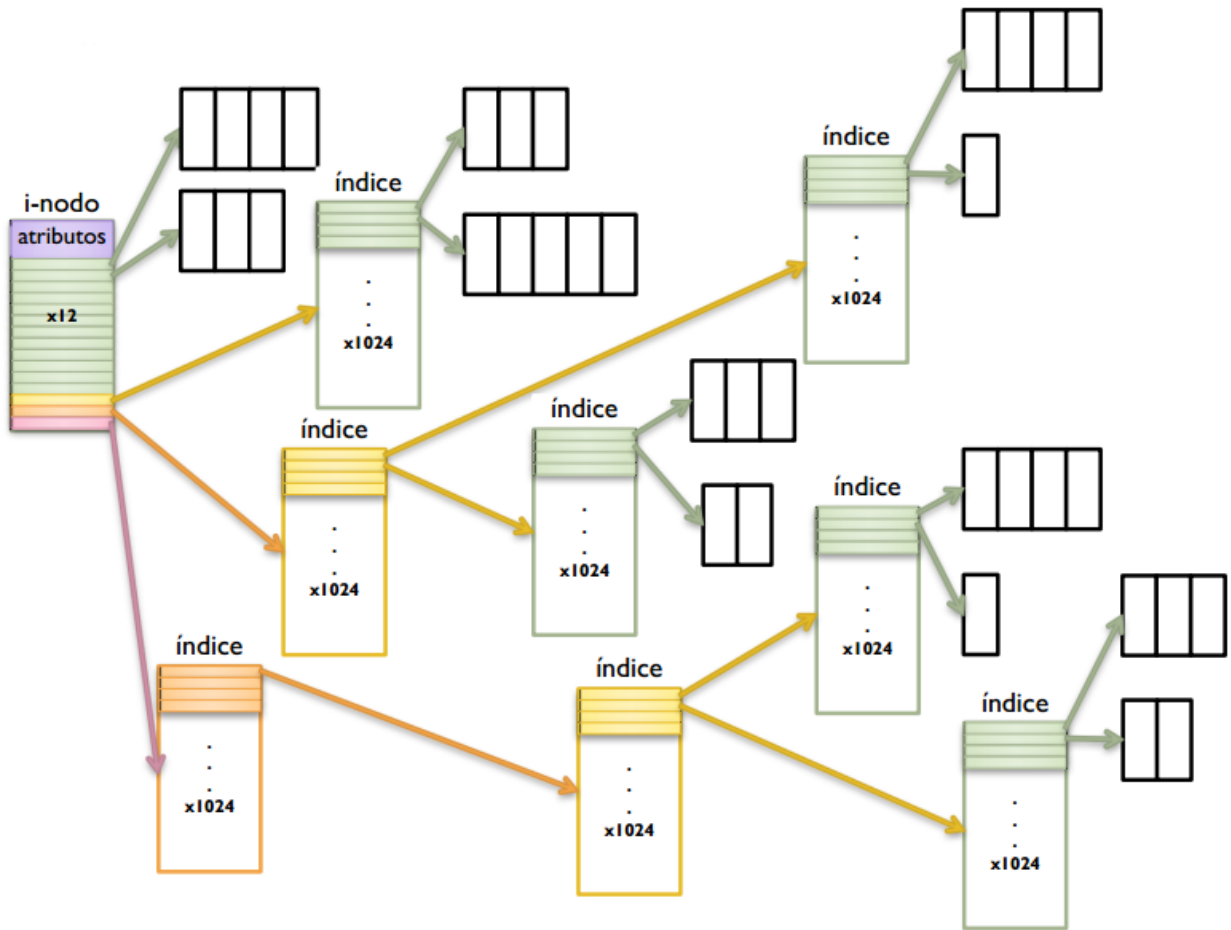
Estructura:

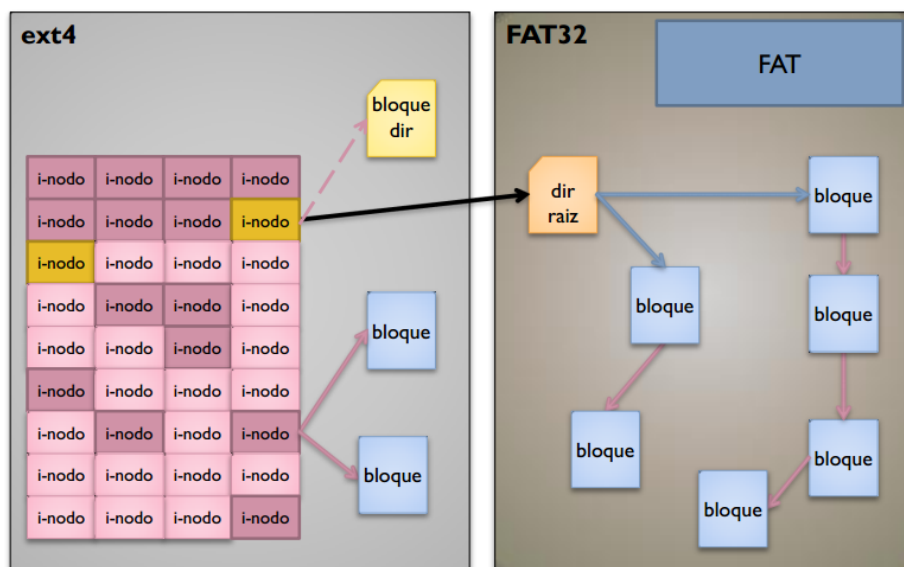
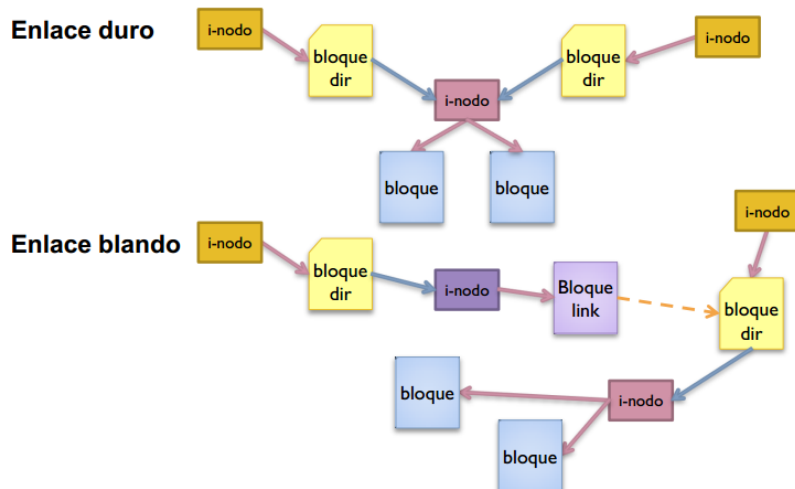
Las particiones o volúmenes se dividen en grupos de bloques.

Cada grupo contiene:

- Superbloque (descriptor de grupo)
- Diario o *Journal*
- Mapa de bloques
- Mapa de i-nodos
- Tabla de i-nodos
- Bloques de datos







Sistema de archivos de nueva tecnología NTFS

¿Qué es? Es el sistema de archivos nativo de los modernos Windows.

Es mucho más complejo y voluminoso que los sistemas de archivos FAT.

Su funcionamiento es un “completo” secreto aunque el proyecto GNU ha podido reproducir algunas operaciones mediante ingeniería inversa.

Aunque desde un sistema basado en Linux se puede leer o escribir en un sistema **NTFS**, no se puede hacer ninguna gestión de permisos.

Muy resistente a los errores.

Permite el cifrado y la compresión de archivos.

Recomendada para unidades de almacenamiento grandes.

Como tiene prestaciones muy parecidas al **ext4** se sospecha que tiene una estructura y funcionamiento similar.

Se sabe que maneja **Extents** de clústeres.

Los directorios son también muy parecidos a **ext4**

Utiliza archivos (meta-archivos) para gestionar su funcionamiento, contenidos en el directorio **System Volume Information**.

Se sabe que posee un **Archivo de registro** con una función similar al **Diario** de **ext4**.

Se sabe que posee un **Archivo de volumen** con una función similar al **Superbloque** de **ext4**.

Se sabe que posee un **Archivo de atributos** para gestionar los metadatos.

Se sabe que tiene un archivo con el mapa de bloques libres.

Se sabe que tiene un archivo de clústeres defectuosos.

Tiene una estructura llamada **Tabla Maestra de Archivos (MFT)**, similar a la **tabla de i-nodos**. Esta tabla no es estática sino que es un archivo capaz de crecer.

El equivalente de los i-nodos se llama ahora **registro de archivo**.

Los archivos pequeños y directorios no precisan clústeres de datos sino que se almacenan en su totalidad en el **MFT** como si fueran metadatos (esto es extremadamente eficiente).

Nota: Utiliza memoria dinámica. Sus archivos son dinámicos, es decir, que pueden cambiar de tamaño cuando quiera. Lo malo de usar una estructura dinámica es que se pierde tiempo. En **ext4** lo i-nodos era limitados, por lo que si necesitabas más tenías que formatear. Aquí en diferencia puede crecer cuando sea necesario

Examen: ¿Cuánto ocupa en total un archivo en **ntfs**? como mínimo 4 k. ¿y en **ext4**?: 4k y 128 bytes.

Universal Disk Format **UDF**

Es el sistema de archivos más extendido en los soportes de almacenamiento óptico.

Este formato permite la lectura, la escritura y la modificación de los archivos contenidos en discos ópticos reescribibles (RW).

La capacidad de reescritura suele implicar la instalación de un manejador externo (Nero InCD, Roxio DirectCD...)

Debido a la cantidad de espacio que se invierte en el sistema de archivos, para **CD-R** se prefiere el sistema **ISO 9660**.

Se suelen utilizar bloques de 512, 1024 o 2048 bytes.

Modos:

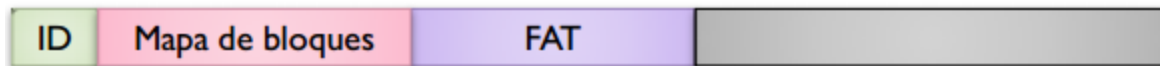
- **Plano** (Acceso aleatorio/Escritura track-at-once o disc-at-once)



- **Virtual Allocated Table (VAT)** (Acceso aleatorio/Escritura incremental)



- **Bien Administrado (Spared)** (Acceso/Escritura aleatoria)



8) Gestión de E/S

Las aplicaciones acceden a los dispositivos de E/S indirectamente, ya que se utiliza al sistema operativo de intermediario.

- **Por practicidad:** El programador de una aplicación no debe preocuparse de la arquitectura del hardware. (Capa de Abstracción)
- **Por protección:** Sólo sistema operativo tiene acceso directamente al hardware y así las aplicaciones no tienen que realizar operaciones críticas (modo kernel).
- **Por eficiencia:** El sistema operativo planifica los accesos a los dispositivos para una utilización óptima.
- **Por robustez:** El sistema operativo monitoriza los posibles errores, tratándolos y corrigiéndolos si es posible.

El sistema operativo gestiona el tráfico de datos mediante el Sistema de E/S.

El sistema de E/S diferencia entre dos tipos de dispositivos según su exclusividad:

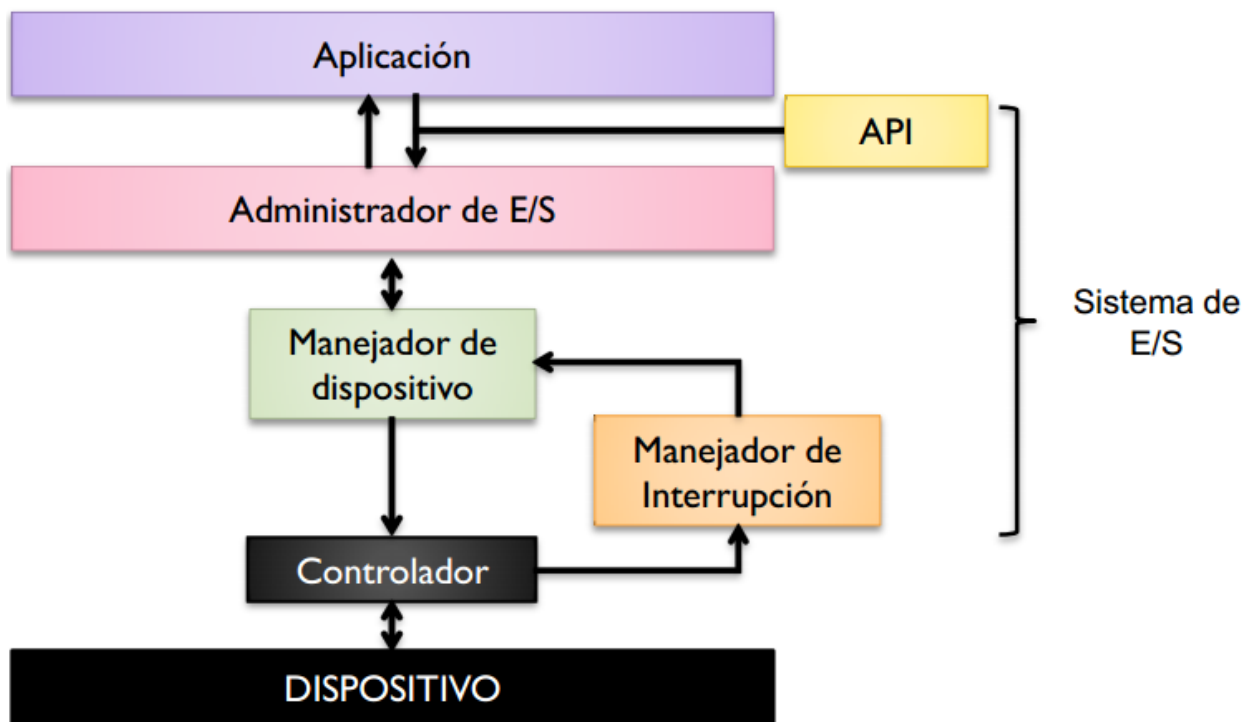
- **Dispositivos de uso exclusivo:** No pueden ser compartidos por dos procesos a la vez.
 - El sistema de E/S asigna este tipo de dispositivos a los procesos que lo soliciten y cuida de que ningún otro lo utilice.
 - Cuando el proceso finaliza, el dispositivo se libera.

- **Dispositivos de uso compartido:** Pueden compartirse concurrentemente por varios procesos.
(p.e. un router)
 - El SO debe garantizar el acceso equitativo por más de un proceso.

... o según la manera en la que se transmiten los datos:

- **Dispositivos organizados por bloques (Unidades de almacenamiento):**
 - Manejan información organizada en bloques de un tamaño fijo accesible mediante una dirección.
 - Se trata cada bloque como unidad independiente del resto.
 - Las operaciones sobre los bloques serán de lectura, escritura o búsqueda.
- **Dispositivos organizados por caracteres (Periféricos):**
 - Reciben y envían los datos como cadenas de bytes o bits sin estructura fija.
 - NO permiten operaciones de búsqueda al no disponer de sistema de direcciones.

El **Sistema de E/S** es la parte del sistema operativo que se encarga de esta gestión:



Manejadores de interrupción:

- Gestionan las interrupciones hardware generadas por los dispositivos.
- Activan al manejador de dispositivo adecuado para la gestión

Manejadores de dispositivo (drivers):

- Son una serie de rutinas que sirven para comunicar cada dispositivo hardware con el sistema operativo.
- Acceder directamente a los registros de los dispositivos.
- Cada manejador se comunica con el controlador de un dispositivo.

Subsistema independiente del dispositivo:

- Presenta una interfaz homogénea para los manejadores de dispositivo.
- Presenta a las aplicaciones una abstracción de los dispositivos.
- Protege los dispositivos del acceso no autorizado de los usuarios.
- Planifica la utilización óptima de los recursos hardware.

Interfaces de aplicación (API):

- Presentan a las aplicaciones un entorno y un juego de procedimientos para que puedan realizar llamadas al sistema (interrupciones software).

Técnicas hardware para aumentar la eficiencia:

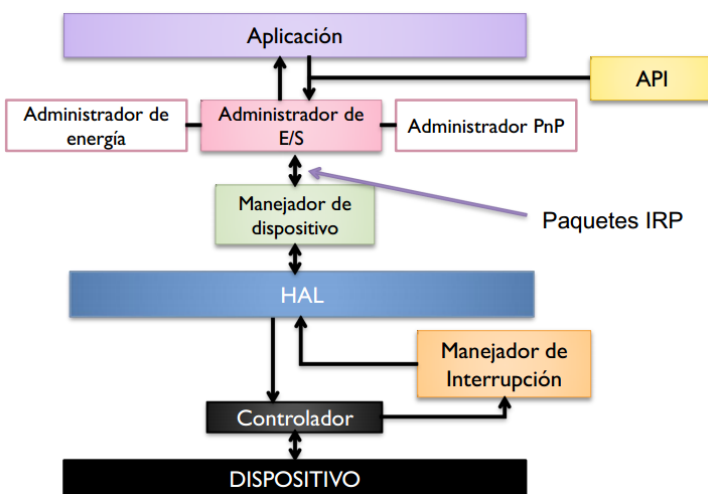
- **Acceso Directo a Memoria (DMA):**
 - La CPU normalmente tiene que controlar la transferencia de datos desde un dispositivo a memoria (o viceversa).
 - El DMA es un chip de la placa base que releva a la CPU controlando la transferencia por él.
 - **Presentan un problema:** los DMA utilizan el bus de datos para acceder a la RAM compitiendo con la CPU.
- **Procesadores de E/S (PE/S) o canales:**
 - Liberan totalmente de operaciones de E/S a la CPU.
 - Establece nuevos buses evitando la competencia con la CPU.
 - **Canal Multiplexor:** maneja la transferencia de varios dispositivos simultáneamente.
 - **Canal Selector:** maneja la transferencia de un dispositivo rápido bloqueando los demás accesos hasta que éste termine.
- **Caching (Utilización de memorias cache):**
 - Almacén auxiliar de alta velocidad para acelerar el acceso a los datos y evitar cuellos de botella.
 - Se utiliza fundamentalmente para evitar en medida de lo posible el acceso a los dispositivos (por lo general, más lentos).
- **Buffering (Utilización de Buffers) Para las cosas rápidas**
 - Almacén auxiliar para permitir que las bajas velocidades de transferencia de algunos dispositivos se compensen con la velocidad de la CPU.
 - Esta técnica no es totalmente efectiva (no se puede satisfacer una demanda masiva).
 - Lo gestiona en el **Administrador de E/S**.

Nota: P.e.: Cuando pulsamos muchas teclas a la vez (en ordenadores malos) se sobrecarga el buffer y emite una alerta sonora.

- **Spooling (Utilización de colas) Para las cosas lentas**
 - Cuando varios procesos mandan datos a un mismo dispositivo lento de uso exclusivo, se usa un espacio de memoria, con estructura de cola, donde se almacenan hasta que el dispositivo esta listo.
 - El sistema operativo crea un directorio de spooling y ejecuta un servicio/demonio especial para esta gestión.

Gestión de E/S en Windows XP (micro híbrido)

- **El administrador de E/S**
 - Es el responsable de todas las operaciones de E/S de Windows.
 - Gestiona los buffers y las cachés.
 - Controla el montaje de los sistemas de archivos.
 - Coordina al Intercambiador (Memoria Virtual)
- **Los manejadores de dispositivo (Drivers)**
 - Manejan solicitudes de E/S procedentes del administrador con un formato estándar llamado Paquete de Solicitud de E/S (IRP).
- **El administrador de Plug and Play (PnP)**
 - Se coordina con el administrador de E/S para asignar recursos a los dispositivos.
 - Detecta y gestiona a la inserción y eliminación de dispositivos, asignando el driver adecuado cuando se precisa.
- **El administrador de energía**
 - Gestiona el encendido y el apagado de los dispositivos.
- **La capa de abstracción del hardware (HAL):**
 - Capa que aísla a los drivers de los detalles del hardware
 - Es el driver por defecto de todos aquellos dispositivos que no tienen driver propio.



Drivers en Windows XP

- **Windows Driver Model (WDM)**

- Los drivers en Windows deben ajustarse a unas especificaciones preestablecidas.

- **El registro**

- Es una base de datos que almacena información del sistema y las aplicaciones instaladas.
- Parte de la información que contiene corresponde a los detalles de cada dispositivo y de su correspondiente driver.

- **Los archivos .inf**

- Son archivos útiles para la instalación y configuración de los dispositivos.

Nota examen: ¿Por qué la sección del administrador de equipos no puede acceder a las interrupciones? Porque no tiene. No tiene interrupciones porque no las maneja directamente.

Gestión de E/S en sistemas basados en Linux

Los dispositivos van a estar representados por archivos especiales de E/S en el directorio /dev.

Cada archivo especial de E/S tiene asociado un **driver**.

Nota: Cada vez que instale un driver hay que recompilar el núcleo. A veces hay driver que tienen su propio script y ya recompila el núcleo solo, pero otras veces no

Los **drivers** en Linux están compuestos de dos partes:

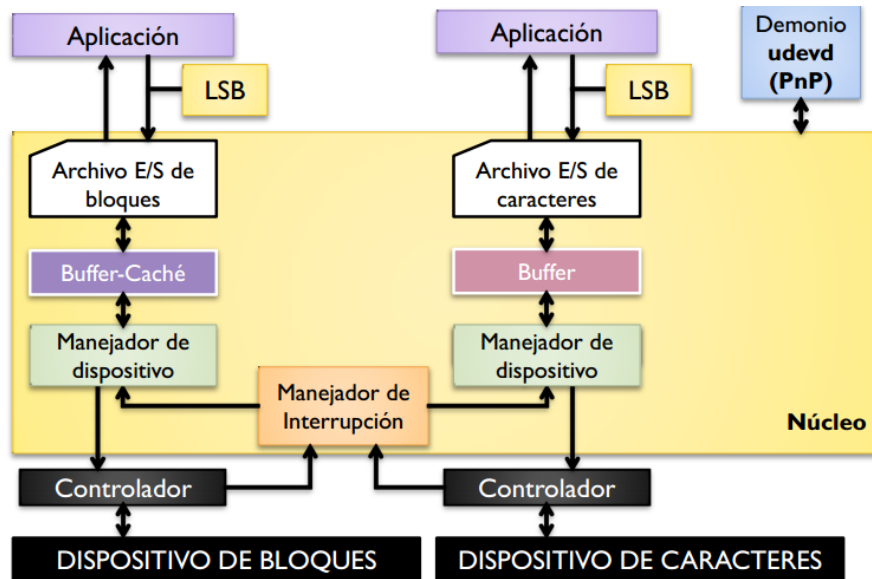
Nota: núcleo monolítico, comunicación directa, no es por paquetes.

- Una parte se comunicará con el **Administrador de E/S** en modo usuario.
- Otra parte se comunicará con el Hardware en modo *kernel*.

Se accede a los dispositivos con operaciones propias de *archivos* (read y write):

- **Dispositivos de bloques:** Utilizan técnicas de caché para reducir al mínimo el número de transferencias efectuadas.
- **Dispositivos de caracteres:** Utilizan técnicas de buffer para compensar la diferencia de velocidad (flujos).

La gestión del Plug and Play se realiza mediante un demonio (udev).



Nota: Forma de buffer. Funcionamiento de caché. Linux retarda la escritura para mover grandes cantidades de información. Si sacas en Linux el predriver rápidamente te lo puedes cargar por ese motivo.