



Tema 1

INTRODUCCIÓN A LOS SISTEMAS INFORMÁTICOS

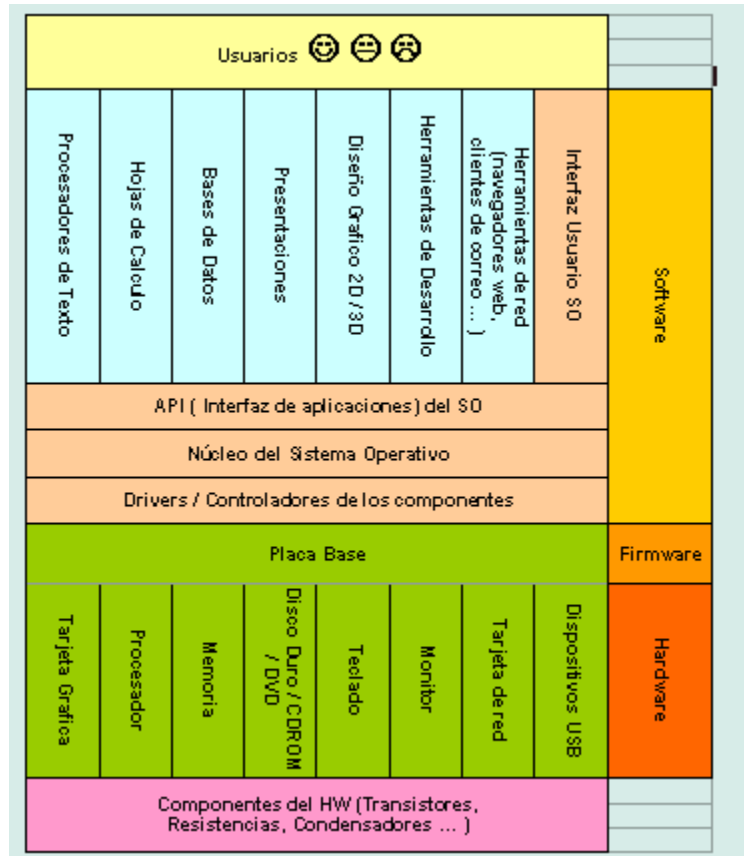
SI | Date

[TYPE HERE]

[TYPE HERE]

[TYPE HERE]

Esquema General de un Sistema Informático



Dividimos el sistema en:

- Software
- Hardware
- Firmware.

El **software** es la parte intangible o lógica del sistema, se puede clasificar en Sistema Operativo, Aplicaciones de Interacción del usuario y los datos.

El **hardware** es la parte tangible o física del sistema, los dispositivos físicos que podemos tocar como el procesador.

Dichos elementos están contruirdos por componentes electrónicos como los transistores, los conectores, las resistencias y los condensadores.



SISTEMAS INFORMÁTICOS

El **firmware** es la parte intangible de los componentes HW, hace que funcione el hw y coordina con los distintos aparatos a los que conecta el ordenador y se guarda en el chip de la **BIOS**.

En el firmware está guardado el programa que se carga cuando se enciende el ordenador y comprueba que todo está correcto. Esta Operación se llama el **Post**.

En el firmware se halla la BIOS (BASIC INPUT / OUTPUT SYSTEM). Se carga al arrancar el ordenador. Gestiona el HW. La Bios Configura el Hw y coordina con los distintos aparatos a los que conecta el ordenador.



LOS USUARIOS

Los usuarios también forman parte del sistema informático.

Hay 4 tipos de usuarios por su grado de conocimiento:

- Comunes (Usuarios finales)
- Operadores / Técnicos (Usuarios Expertos)
- Administradores. (Usuarios informáticos).
- Desarrolladores (Creadores del Sw)

Evolución histórica de los sistemas operativos

Introducción

El hardware y el software de los sistemas informáticos han evolucionado de forma paralela y conjunta en las últimas décadas. Por lo que la evolución que vamos a ver de los sistemas operativos está estrechamente relacionada con los avances en la arquitectura de los ordenadores que se produjo de cada generación.

PRIMERA GENERACIÓN (1945-1955)

Los primeros ordenadores estaban contruidos con tubos de vacío. En un principio no existían sistemas operativos, se programaba directamente sobre el hardware. Los programas estaban hechos directamente en código máquina y el control de las funciones básicas se realiza mediante paneles enchufables.

Hacia finales de 1950 aparecen las tarjetas perforadas que sustituyen los paneles enchufables. Las tarjetas perforadas supusieron un enorme paso ya que permitían codificar instrucciones de un programa y los datos en una cartulina con puntos que podía interpretar el ordenador. La mayoría de los programas usaban rutinas de E/S y un programa cargador (automatizaba la carga de programas ejecutables en la máquina) esto constituía una forma rudimentaria de sistema operativo.

2ª GENERACIÓN (1955-1965)

Esta generación se caracteriza por la aparición de los transistores que permitieron la construcción de ordenadores más pequeños y potentes. La programación se realizaba en lenguaje ensamblador y en FORTRAN sobre tarjetas perforadas. Otro aspecto importante de esta generación es el procesamiento por lotes, en el cual mientras el sistema operativo está ejecutando un proceso, éste último dispone de todos los recursos hasta su finalización.

La preparación de los trabajos se realiza a través de un lenguaje de control de trabajos conocido como JCL. El sistema operativo residía en memoria y tenía un programa de control que interpretaba las tarjetas de control, escritas JCL.



SISTEMAS INFORMÁTICOS

Dependiendo del contenido de la tarjeta de control el sistema operativo realizaba una acción determinada. Este programa de control es un antecedente de los modernos intérpretes de órdenes. Procesamiento Fuera de línea (Offline) Como mejora del procesamiento por lotes surgió el procesamiento fuera de línea (off-line), en el cual las operaciones de carga de datos y salida de resultados de un proceso podían realizarse de forma externa y sin afectar al tiempo que el procesador dedicaba a los procesos. A esto ayudó la aparición de las cintas magnéticas y las impresoras de líneas. Ejemplos de sistemas operativos de la época son FMS (Fortran Monitor System) y IBSYS.

3ª GENERACIÓN (1965-1980)

La aparición de los circuitos integrados (CI) supuso una mejora consiguiendo un menor tamaño y relación precio/rendimiento respecto de las máquinas de generaciones anteriores. En relación con los sistemas operativos, la característica principal de esta generación fue el desarrollo de la multiprogramación y los sistemas compartidos. En los sistemas multiprogramados se cargan varios programas en memoria simultáneamente y se alterna su ejecución. Esto maximiza la utilización del procesador. Como evolución de aparecen los sistemas de tiempo compartido donde el tiempo del procesador se comparte entre programas de varios usuarios pudiendo ser programas interactivos.

Algunos de los sistemas operativos de esta generación son OS/360, CTSS, MULTICS y UNIX.

4ª GENERACIÓN (1980-HASTA HOY)

En esta generación se producen grandes avances en la industria hardware como la creación de los circuitos LSI (integrados a gran escala). También aparecen los ordenadores personales, entre finales de la anterior generación y principios de la presente. Ejemplos de sistemas operativos de los primeros ordenadores personales son MS-DOS, desarrollado por Microsoft, Inc., para el IBM PC y MacOS de Apple Computer, Inc. Steve Jobs, cofundador de Apple, apostó por la primera interfaz gráfica basada en ventanas, iconos, menús y ratón a partir de una investigación realizada por Xerox.

Siguiendo esta filosofía aparecería MS Windows. Durante los 90 apareció Linux a partir del núcleo desarrollado por Linus Torvalds. Los sistemas operativos evolucionan hacia sistemas interactivos con una interfaz cada vez más amigable al usuario. Los sistemas Windows han ido evolucionando, con diferentes versiones tanto para escritorio como para servidor (Windows 3.x, 98, 2000, XP, Vista, 7, Windows Server 2003, 2008, etc), al igual que lo han hecho Linux (con multitud de distribuciones, Ubuntu, Debian, RedHat, Mandrake, etc) y los sistemas Mac (Mac OS 8, OS 9, OS X, Mac OS X 10.6 "Snow Leopard", entre otros).

Un avance importante fue el desarrollo de redes de ordenadores a mediados de los años 80 que ejecutan sistemas operativos en red y sistemas operativos distribuidos. En un sistema



SISTEMAS INFORMÁTICOS

operativo en red los usuarios tienen conocimiento de la existencia de múltiples ordenadores y pueden acceder a máquinas remotas y copiar archivos de un ordenador a otro. En un sistema distribuido los usuarios no saben dónde se están ejecutando sus programas o dónde están ubicados sus programas, ya que los recursos de procesamiento, memoria y datos están distribuidos entre los ordenadores de la red, pero todo esto es transparente al usuario.

Actualmente, existen sistemas operativos integrados, para una gran diversidad de dispositivos electrónicos, tales como, teléfonos móviles, PDAs (Personal Digital Assistant, Asistente Digital Personal u ordenador de bolsillo), otros dispositivos de comunicaciones e informática y electrodomésticos. Ejemplos de este tipo de sistemas operativos son PalmOS, WindowsCE, Android OS, etc. Haremos una referencia especial al último, Android OS, se trata de un sistema operativo basado en Linux. Fue diseñado en un principio para dispositivos móviles, tales como teléfonos inteligentes y tablets, pero actualmente se encuentra en desarrollo para su aplicación también en netbooks y PCs.

El procesamiento por lotes aparece durante:

- La 1ª generación. La 2ª generación. La 3ª generación. La 4ª generación.

Los sistemas operativos en red y distribuidos aparecen durante:

- La 1ª generación. La 2ª generación. La 3ª generación. La 4ª generación.

¿Quieres conocer cómo se crearon dos de las empresas más importantes de informática, Apple Computer y Microsoft?

Visita este interesante enlace:

<http://www.venagraphica.es/blog/wp-content/uploads/file/losPrimerosPcs.pdf>

SISTEMAS OPERATIVOS

Caso práctico

En BK Programación, los protagonistas de nuestra historia deben conocer a fondo todos los productos que el mercado ofrece en cuanto a software de sistema, y así aconsejar con garantías cuál se adecua mejor a las necesidades y situaciones que su clientela le pueda plantear.

Juan dice que deben conocer a fondo la teoría en la que están basados los desarrollos de los sistemas operativos. Conociendo cómo funcionan los sistemas “por dentro”

SISTEMAS INFORMÁTICOS

podrán llegar a ser unos buenos profesionales con criterio propio, capaces de valorar las características de cada sistema. Y lo que es más importante, conociendo los entresijos de las máquinas y el software que van a utilizar podrán sacarles el máximo partido. En BK Programación existen diferentes posturas como vimos en un apartado anterior. Esa es la razón de que no se cierren a ninguna opción, por ello, la empresa cuenta con personal especializado en sistemas Linux y para los sistemas Windows de Microsoft y Mac de Apple. En BK Programación conocen la importancia de estar al día de los productos software del mercado y los nuevos que aparecerán.

CONCEPTO Y OBJETIVOS DE LOS SISTEMAS OPERATIVOS

El sistema operativo es un conjunto de programas que se encarga de gestionar los recursos hardware y software del ordenador, por lo que actúa como una interfaz entre los programas de aplicación del usuario y el hardware puro.



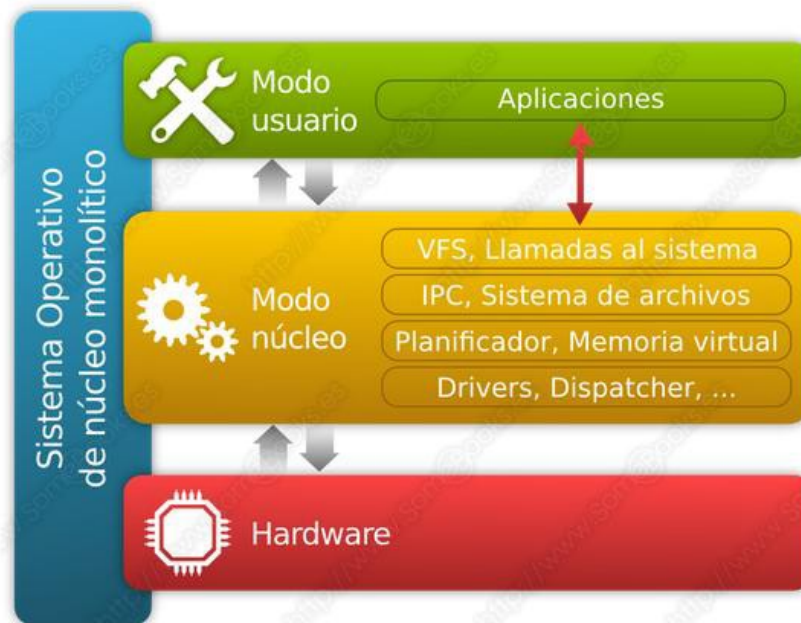
Los principales objetivos de los sistemas operativos son:

- **Abstraer al usuario de la complejidad del hardware:** El sistema operativo hace que el ordenador sea más fácil de utilizar.
- **Eficiencia:** Permite que los recursos del ordenador se utilicen de la forma más eficiente posible. Por ejemplo, se deben optimizar los accesos a disco para acelerar las operaciones de entrada y salida.
- **Permitir la ejecución de programas:** Cuando un usuario quiere ejecutar un programa, el sistema operativo realiza todas las tareas necesarias para ello, tales como cargar las instrucciones y datos del programa en memoria, iniciar dispositivos de entrada/salida y preparar otros recursos.
- **Acceder a los dispositivos entrada/salida:** El sistema operativo suministra una interfaz homogénea para los dispositivos de entrada/salida para que el usuario pueda utilizar de forma más sencilla los mismos.
- **Proporcionar una estructura y conjunto de operaciones para el sistema de archivos.**

SISTEMAS INFORMÁTICOS

- Controlar el acceso al sistema y los recursos: en el caso de sistemas compartidos, proporcionando protección a los recursos y los datos frente a usuarios no autorizados.
- Detección y respuesta ante errores: El sistema operativo debe prever todas las posibles situaciones críticas y resolverlas, si es que se producen.
- Capacidad de adaptación: Un sistema operativo debe ser construido de manera que pueda evolucionar a la vez que surgen actualizaciones hardware y software.
- Gestionar las comunicaciones en red: El sistema operativo debe permitir al usuario manejar con facilidad todo lo referente a la instalación y uso de las redes de ordenadores.
- Permitir a los usuarios compartir recursos y datos: Este aspecto está muy relacionado con el anterior y daría al sistema operativo el papel de gestor de los recursos de una red.

MODO USUARIO/NUCLEO



Para que esta protección sea posible, el sistema informático debe ser capaz de ejecutar instrucciones en dos niveles diferentes:

- En modo usuario: es el modo menos privilegiado de funcionamiento del sistema. En este modo no se permite el acceso directo al *hardware*. Las instrucciones que se ejecutan en este modo sólo pueden acceder a su propio *espacio de direcciones* de memoria y utilizan el *API* del sistema para requerir



SISTEMAS INFORMÁTICOS

los servicios del *sistema operativo*. Este es el modo de ejecución que utilizan todos los programas de aplicación que tengamos instalados.

- En modo núcleo (también llamado modo kernel) o modo supervisor: En él, las instrucciones se ejecutan en un modo privilegiado, teniendo acceso directo a toda la memoria (incluidos los espacios de direcciones de todos los procesos que estén ejecutándose). También podrán acceder a todo el hardware disponible. En este modo sólo se ejecutan algunas partes del sistema operativo.

TIPOS DE SISTEMAS OPERATIVOS

Ahora vamos a clasificar los sistemas operativos en base a su estructura, servicios que suministran y por su forma. Tipos de sistemas operativos

POR ESTRUCTURA

- Monolíticos
- Jerárquicos
- Máquina Virtual
- Microkernel o Cliente-Servidor

POR SUS SERVICIOS

- Monotarea
- Multitarea
- Monousuario
- Multiusuario
- Monoprocesador
- Multiprocesador

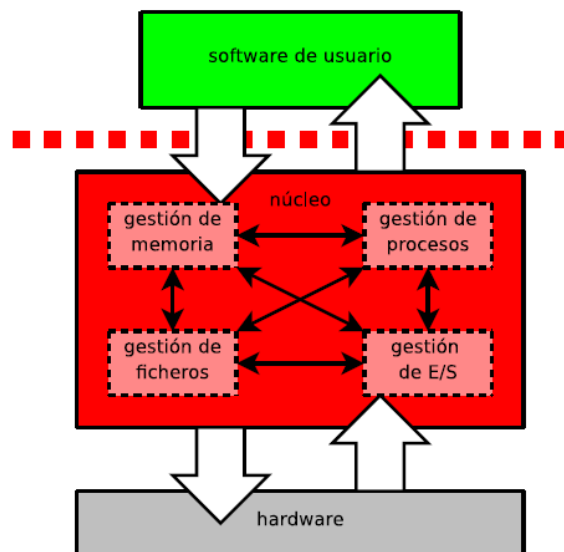
POR SU FORMA

- Sistema operativo en red
- Sistema operativo distribuido

Tipos de sistemas operativos		
Por estructura	Por sus servicios	Por su forma
Monolíticos	Monousuario	Sistema operativo en red
Jerárquicos	Multiusuario	
Máquina Virtual	Monotarea	Sistema operativo distribuido
Microkernel o Cliente-Servidor	Multitarea	
Monolíticos	Monoprocesador	
	Multiprocesador	

SISTEMAS OPERATIVOS POR SU ESTRUCTURA

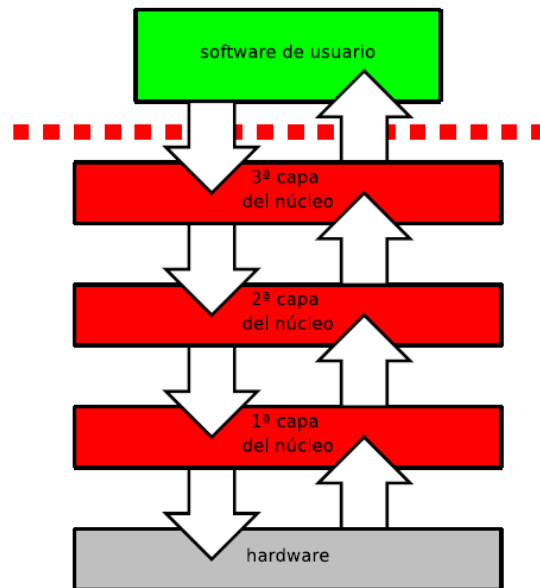
Monolíticos: Es la estructura de los primeros sistemas operativos, consistía en un solo programa desarrollado con rutinas entrelazadas que podían llamarse entre sí. Por lo general, eran sistemas operativos hechos a medida, pero difíciles de mantener.



Jerárquicos: Conforme las necesidades de los usuarios aumentaron, los sistemas operativos fueron creciendo en complejidad y funciones.

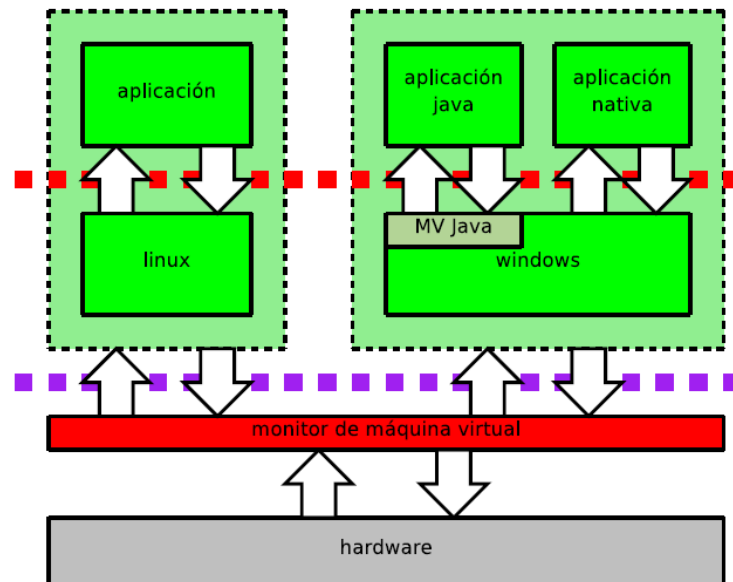
Esto llevó a que se hiciera necesaria una mayor organización del software del sistema operativo, dividiéndose en partes más pequeñas, diferenciadas por funciones y con una interfaz clara para interoperar con los demás elementos.

Un ejemplo de este tipo de sistemas operativos fue MULTICS.

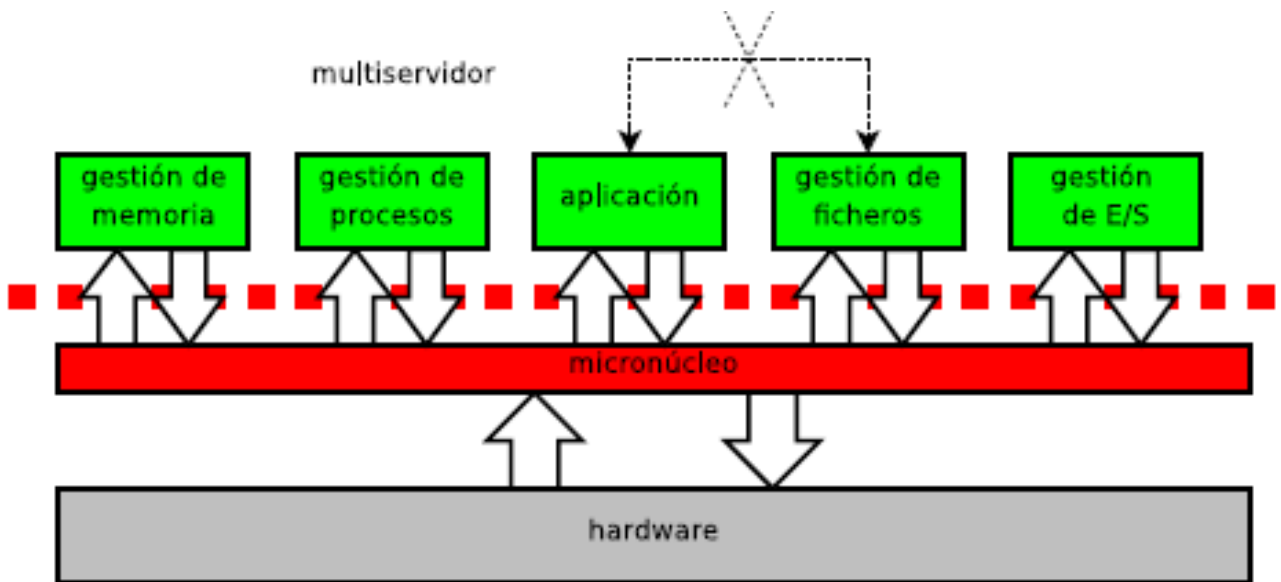


Máquina Virtual: El objetivo de los sistemas operativos es el de integrar distintos sistemas operativos dando la sensación de ser varias máquinas diferentes. Presentan una interfaz a cada proceso, mostrando una máquina que parece idéntica a la máquina real subyacente.

Estas máquinas no son máquinas extendidas, son una réplica de la máquina real, de manera que en cada una de ellas se pueda ejecutar un sistema operativo diferente, que será el que ofrezca la máquina extendida al usuario. VMware y VM/CMS son ejemplos de este tipo de sistemas operativos.



Microkernel o Cliente-Servidor: El modelo del núcleo de estos sistemas operativos distribuye las diferentes tareas en porciones de código modulares y sencillas. El objetivo es aislar del sistema, su núcleo, las operaciones de entrada/salida, gestión de memoria, del sistema de archivos, etc. Esto incrementa la tolerancia a fallos, la seguridad y la portabilidad entre plataformas de hardware. Algunos ejemplos son MAC OS X o AIX.



Sistemas operativos por sus servicios

- **Monousuario:** Son aquellos que soportan a un usuario a la vez, sin importar el número de procesos o tareas que el usuario pueda ejecutar en un mismo instante de tiempo. Ejemplos de sistemas operativos de este tipo son MS-DOS, Microsoft Windows 9x y ME, MAC OS, entre otros.
- **Multiusuario:** Son capaces de dar servicio a más de un usuario a la vez, ya sea por medio de varias terminales conectadas al ordenador o por medio de sesiones remotas en una red de comunicaciones. No importa el número de procesadores en la máquina ni el número de procesos que puede ejecutar cada usuario simultáneamente. Algunos ejemplos serán UNIX, GNU/Linux, Microsoft Windows Server o MAC OS X.
- **Monotarea:** Sólo permiten una tarea a la vez por usuario. Se puede dar el caso de un sistema multiusuario y monotarea, en el cual se admiten varios usuarios simultáneamente pero cada uno de ellos puede ejecutar sólo una tarea en un instante dado. Ejemplos de sistemas monotarea son MS-DOS, Microsoft Windows 3.x y 95 (estos últimos sólo simulan la multitarea).



SISTEMAS INFORMÁTICOS

- **Multitarea:** Permite al usuario realizar varias tareas al mismo tiempo. Algunos ejemplos son MAC OS, UNIX, Linux, Microsoft Windows 98, 2000, XP, Vista y 7.
- **Monoprocesador:** Es aquel capaz de manejar sólo un procesador, de manera que si el ordenador tuviese más de uno le sería inútil. MS-DOS y MAC OS son ejemplos de este tipo de sistemas operativos.
- **Multiprocesador:** Un sistema operativo multiprocesador se refiere al número de procesadores del sistema, éste es más de uno y el sistema operativo es capaz de utilizarlos todos para distribuir su carga de trabajo. Estos sistemas trabajan de dos formas:
 - Simétricamente (los procesos son enviados indistintamente a cualquiera de los procesadores disponibles)
 - Asimétricamente (uno de los procesadores actúa como maestro o servidor y distribuye la carga de procesos a los demás).

SISTEMAS OPERATIVOS POR SU FORMA

- **Sistemas operativos en red:** Estos sistemas tienen la capacidad de interactuar con los sistemas operativos de otras máquinas a través de la red, con el objeto de intercambiar información, transferir archivos, etc. La clave de estos sistemas es que el usuario debe conocer la ubicación de los recursos en red a los que desee acceder. Los sistemas operativos modernos más comunes pueden considerarse sistemas en red, por ejemplo: Novell, Windows Server, Linux, etc.
- **Sistemas operativos distribuidos:** Abarcan los servicios de red, las funciones se distribuyen entre diferentes ordenadores, logrando integrar recursos (impresoras, unidades de respaldo, memoria, procesos, etc.) en una sola máquina virtual que es a la que el usuario accede de forma transparente. En este caso, el usuario no necesita saber la ubicación de los recursos, sino que los referencia por su nombre y los utiliza como si fueran locales a su lugar de trabajo habitual. MOSIX es un ejemplo de estos sistemas operativos.

El sistema operativo necesita administrar los recursos para tener control sobre las funciones básicas del ordenador. Pero, ¿cuáles son los recursos que gestiona el sistema operativo? Los principales recursos que administra el sistema operativo son:

- El procesador.
- La memoria.
- Los dispositivos de entrada/salida.
- El sistema de archivos.



Núcleo

Para gestionar todos estos recursos, existe una parte muy importante del sistema operativo, el núcleo o kernel. El núcleo normalmente representa sólo una pequeña parte de todo lo que es el sistema operativo, pero es una de las partes que más se utiliza. Por esta razón, el núcleo reside por lo general en la memoria principal, mientras que otras partes del sistema operativo son cargadas en la memoria principal sólo cuando se necesitan.

Resumiendo, el núcleo supone la parte principal del código de un sistema operativo y se encarga de controlar y administrar los servicios y peticiones de recursos. Para ello se divide en distintos niveles:

- Gestión de procesos
- Gestión de memoria
- Gestión de la entrada/salida (E/S)
- Gestión del Sistema de archivos

Los servicios principales que presta un sistema operativo son: Gestión del procesador, gestión de memoria y de entrada/salida. Gestión del procesador, gestión de memoria, de Entrada/Salida y del Sistema de archivos. Gestión del procesador y gestión de memoria. Gestión de memoria, de entrada/ salida y del sistema de archivos.

Gestión de procesos

Caso práctico

Ana y Carlos tienen dudas sobre lo que afecta a la eficiencia de un ordenador. María y Juan les explican que el rendimiento efectivo de un ordenador no sólo se basa en sus prestaciones hardware, sino también en el software que se ejecute y en el tipo de carga de trabajo que procese.

La ejecución de los distintos programas que se lanzan en un equipo se traduce en multitud de procesos que requieren recursos. Uno de los principales recursos del ordenador es el procesador.

María y Juan le comentan a Ana y Carlos con más de detalle en qué consiste la gestión de procesos y que opciones existen. Entre las principales tareas del sistema operativo está la de administrar los procesos del sistema. ¿A qué nos referimos cuando hablamos de procesos? Un proceso es un programa en ejecución. Un proceso simple tiene un hilo

de ejecución (o subproceso), en ocasiones, un proceso puede dividirse en varios subprocesos.

Un hilo es básicamente 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.



En los sistemas operativos modernos los procesos pueden tener diferentes estados, según el momento de creación, si están en ejecución, si se encuentran a la espera de algún recurso, etc. Pero podemos hacer una simplificación, y un proceso, en un instante dado, puede estar en uno de los tres estados siguientes:

- Listo.
- En ejecución.
- Bloqueado.

Los procesos en estado listo son los que pueden pasar a estado de ejecución si el planificador del sistema operativo los selecciona, esto es, cuando llegue su turno (según el orden de llegada o prioridad).

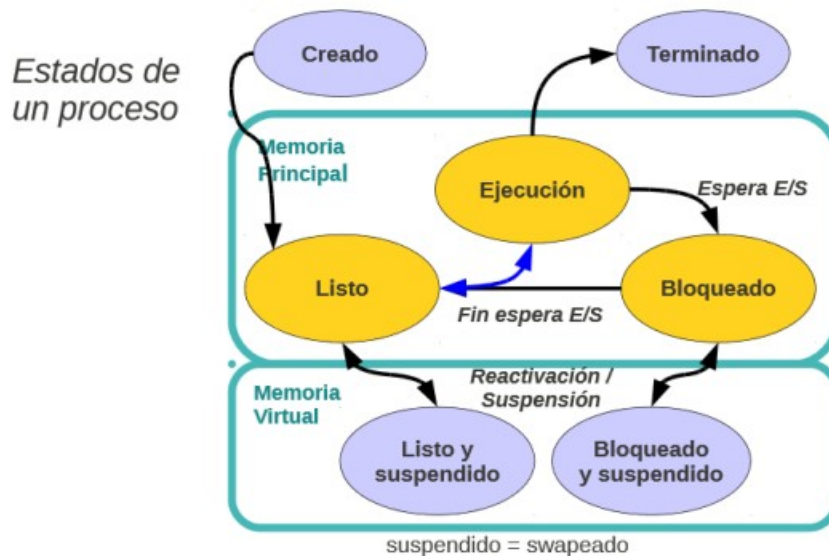
Los procesos en estado de ejecución son los que se están ejecutando en el procesador en un momento dado.

Los procesos que se encuentran en estado bloqueado están esperando la respuesta de algún otro proceso para poder continuar con su ejecución, por ejemplo, una operación de entrada/salida. El sistema operativo sigue la pista de en qué estado se encuentran los procesos, decide qué procesos pasan a ejecución, cuáles quedan bloqueados, en definitiva, gestiona los cambios de estado de los procesos.

Los procesos pueden comunicarse entre sí o ser independientes. En el primer caso, los procesos necesitarán sincronizarse y establecer una serie de mecanismos para la

comunicación; por ejemplo, los procesos que pertenecen a una misma aplicación y necesitan intercambiar información. En el caso de procesos independientes estos, por lo general, no interactúan y un proceso no requiere información de otros.

En la planificación del procesador se decide cuánto tiempo de ejecución se le asigna a cada proceso del sistema y en qué momento y cada proceso va a estar en un estado según sus propiedades en el SO.



ESTADOS DE UN PROCESO

- **Creado:** Se le están creando su tabla y cargando su código
- **Ejecución:** Estado en el que se encuentra un proceso cuando tiene el control del procesador. En un sistema monoprocesador este estado sólo lo puede tener un proceso.
- **Listo:** Aquellos procesos que están dispuestos para ser ejecutados, pero no están en ejecución por alguna causa (Interrupción, haber entrado en cola estando otro proceso en ejecución, etc.).
- **Bloqueado:** Son los procesos que no pueden ejecutarse de momento por necesitar algún recurso no disponible (generalmente recursos de entrada/salida).
- **Suspendido bloqueado:** Es el proceso que fue suspendido en espera de un evento, sin que hayan desaparecido las causas de su bloqueo.
- **Suspendido programado:** Es el proceso que han sido suspendido, pero no tiene causa para estar bloqueado.

Si el sistema es monousuario y monotarea no habrá que decidir, pero en el resto de los sistemas multitarea esta decisión es fundamental para el buen funcionamiento del sistema, ya que determinará la correcta ejecución de los distintos programas de aplicación que se estén ejecutando.

PLANIFICACIÓN DEL PROCESADOR

En la planificación del procesador se decide cuánto tiempo de ejecución se le asigna a cada proceso del sistema y en qué momento y cada proceso va a estar en un estado según sus propiedades en el SO.

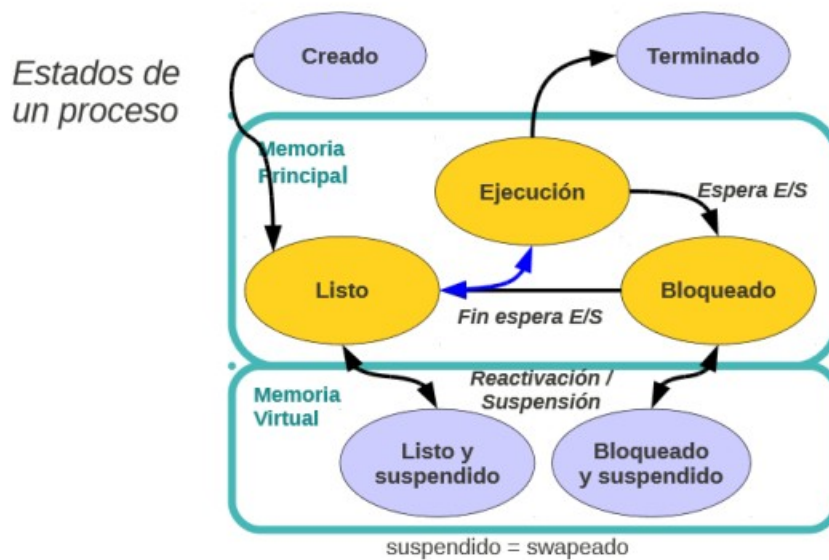


TABLA DE CONTROL DEL PROCESO

El sistema operativo almacena en una tabla denominada tabla de control de procesos con la información relativa a cada proceso que se está ejecutando en el procesador. Este es:

Nombre	Descripción
Imagen	Nombre del fichero ejecutable que dio lugar al proceso
PID	Numero de Identificación del proceso.
PPID	Numero de Identificación del proceso padre.
Nombre de usuario	Información sobre el usuario y grupo que lo han lanzado.
Estado	Estado del proceso en el procesador.
Datos de estado	El contenido de los registros internos, contador de programa, etc. Es decir, el entorno volátil del proceso.
Tiempo de CPU	Información de control de proceso.
Uso de memoria	Segmentos de memoria asignados.
Recursos asignados	Descriptores de ficheros, dispositivos o sockets de red asignados al proceso.
Nombre de sesión y Numero de sesión	Datos sobre la sesión en la que se está ejecutando
Título de ventana	Nombre del interfaz Grafico del proceso en el escritorio

ESTRATEGIA DE PLANIFICACIÓN

Una estrategia de planificación debe buscar que los procesos obtengan sus turnos de ejecución de forma apropiada (momento en que se le asigna el uso de la CPU), junto con un buen rendimiento y minimización de la sobrecarga (overhead) del planificador mismo. En general, se buscan cinco objetivos principales:

- 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. Esto es, el usuario no debe percibir que su programa se ha parado o “colgado”.

Tipos de Procesos

La carga de trabajo de un sistema informático a otro puede variar considerablemente, esto depende de las características de los procesos. Nos podemos encontrar:

Características	Descripción
Uso de CPU	Procesos que hacen un uso intensivo de la CPU. Por ejemplo calculos matemáticos
E/S	Procesos que realizan una gran cantidad de operaciones de Entrada/Salida .Esto es acceso a disco duro o memoria
Tipos de Procesos	<ul style="list-style-type: none"> • Procesos por lotes (el usuario no hace nada) • Procesos interactivos (el usuario interactua mucho) • Procesos en tiempo real(Tienen que ejecutarse inmediatamente).
Duración	Procesos de menor o mayor duración.

En función de cómo sean la mayoría de los procesos habrá algoritmos de planificación que den un mejor o peor rendimiento al sistema.

PLANIFICACIÓN APROPIATIVA Y NO APROPIATIVA

La planificación no apropiativa (en inglés, no preemptive) es aquella en la que, cuando a un proceso le toca su turno de ejecución, ya no puede ser suspendido; es decir, no se le puede arrebatar el uso de la CPU, hasta que el proceso no lo determina no se podrá ejecutar otro proceso. Este esquema tiene sus problemas, puesto que, si el proceso contiene ciclos infinitos, el resto de los procesos pueden quedar aplazados indefinidamente. Otro caso puede ser el de los procesos largos que penalizarían a los cortos si entran en primer lugar.

La planificación apropiativa (en inglés, preemptive) supone que el sistema operativo puede arrebatar el uso de la CPU a un proceso que esté ejecutándose. En la planificación apropiativa existe un reloj que lanza interrupciones periódicas en las cuales el planificador toma el control y se decide si el mismo proceso seguirá ejecutándose o se le da su turno a otro proceso. En ambos enfoques de planificación se pueden establecer distintos algoritmos de planificación de ejecución de procesos.

ALGORITMOS

Algunos de los algoritmos para decidir el orden de ejecución de los procesos en el sistema son:

FIFO

En el algoritmo **"Primero en llegar, primero en ejecutarse"**: Su tiempo de respuesta puede ser alto, especialmente si varían mucho los tiempos de ejecución. La sobrecarga del sistema es mínima. Penaliza los procesos cortos y los procesos con operaciones de Entrada/Salida.

Algoritmo basado en prioridades

- A cada proceso en el sistema se le asigna un entero basándose en algún criterio



SISTEMAS INFORMÁTICOS

- Generalmente, un valor numérico bajo indica 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. **Produce Muerte por inanición.**

Ejemplos de algoritmos basados en prioridades son:

Shortest Job First (SJF)

Es non-preemptive, **se ejecutan los procesos en orden ascendente según el tiempo de ejecución de cada uno**. Se asigna la CPU al proceso más corto, en caso de “empate” se usa FCFS.

Es óptimo con respecto al tiempo de espera promedio mínimo para un conjunto de procesos. Apropiado para procesos batch donde los tiempos de ejecución se conocen de antemano. Ofrece un límite teórico (para el tiempo de espera promedio) frente al cual pueden compararse otros algoritmos.

¿Quieres saber cómo se comunican y sincronizan entre sí los procesos? Visita el siguiente enlace:

<http://www.redes-linux.com/apuntes/so1/teoria/tema4.pdf>

Round Robin

Algoritmo Preemptive, apto para procesos interactivos, simples, justos y ampliamente usados:

- A cada proceso se le asigna un intervalo de tiempo llamado quantum. Si un proceso continúa su ejecución superando su quantum, es preempted y la CPU es dada a otro proceso.
- Si un proceso se bloquea o finaliza antes de que expire el quantum, la CPU se reasigna en ese momento.
- Cuando un proceso usa todo su quantum se lo coloca al final de la lista de procesos listos – suposición: todos los procesos tienen la misma importancia.

Gestión de memoria

Caso práctico

Ana y Carlos tras entender la importancia de una adecuada gestión de procesos, preguntan: — ¿Qué ocurre con la memoria principal? ¿Cómo la utiliza el sistema operativo para poder ejecutar los procesos de manera equilibrada? —Juan les explica que existen diferentes posibilidades en la gestión de la memoria que hace el sistema operativo. Pero lo primero es tener claro ciertos conceptos sobre el uso de la memoria. Juan comienza su explicación...

FUNCIONES DE LA GESTIÓN DE MEMORIA

Hemos visto en la gestión de procesos que el recurso compartido es el procesador. Sin embargo, para que un proceso se pueda ejecutar no sólo requiere tiempo de procesamiento sino también estar cargado en memoria principal. Esto es así, porque ningún proceso se puede activar antes de que se le asigne el espacio de memoria que requiere. Así, la memoria se convierte en otro recurso clave que tendrá que gestionar el sistema operativo y la parte encargada de ello se denomina gestor de memoria. La función principal del gestor de memoria es la de asignar memoria principal a los procesos que la soliciten.

Otras funciones serán:

Función	Descripción
Memoria Libre	Controlar las zonas de memoria que están asignadas y cuáles no.
Asignar y retirar Memoria	Asignar memoria a los procesos cuando la necesiten y retirársela cuando terminen.
Proteccion del espacio	Evitar que un proceso acceda a la zona de memoria asignada a otro proceso.
Gestion del Swap	Gestionar el intercambio entre memoria principal y memoria secundaria en los casos en que la memoria principal está completamente ocupada, etc.

REQUISITOS DE GESTIÓN DE MEMORIA

De este modo, la gestión de memoria va a tener que cubrir los siguientes requisitos:

- **Reubicación:** En un sistema multitarea la memoria va a estar compartida entre varios procesos, el gestor de memoria debe decidir 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, Para ello, se comprueba que las referencias a la memoria generadas por un proceso durante su ejecución sólo hacen referencia a la zona de memoria asignada a ese proceso y no acceden a zonas prohibidas, áreas de memoria donde estén otros procesos.

- **Control de memoria:** El sistema operativo, a través del gestor de memoria, tiene que controlar las zonas de memoria libres y las asignadas, además de saber las zonas de memoria que corresponden a cada proceso.
- **Controlar y evitar en lo posible casos de fragmentación de la memoria:** Existen dos tipos de fragmentación de la memoria principal, la fragmentación interna y la externa. La fragmentación interna sucede al malgastarse el espacio interno de una partición cuando el proceso o bloque de datos cargado es más pequeño que la partición. Por el contrario, la fragmentación externa sucede cuando la memoria externa a todas las particiones se divide cada vez más y van quedando huecos pequeños y dispersos en memoria difícilmente reutilizables.
- **Organización lógica y física:** En ocasiones la memoria principal no es suficiente para proporcionar toda la memoria que necesita un proceso o para almacenar todos los procesos que se pueden ejecutar. Entonces los procesos pueden ser intercambiados a disco y más tarde, si es necesario, vueltos a cargar en memoria. Por lo que el gestor de memoria se encarga de gestionar la transferencia de información entre la memoria principal y la secundaria (disco).

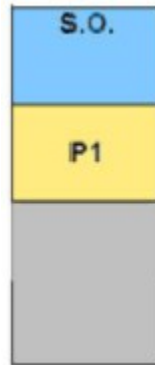
El sistema de gestión de la memoria que se use dependerá del ordenador y sistema operativo en particular que se tenga. Las opciones en la gestión de memoria se dividen en función del número de procesos albergados en memoria (monotarea/multitarea) y de si se utiliza memoria real o virtual.

Gestión de la memoria con memoria real y virtual				
Memoria Real	Memoria Real		Memoria Virtual	
	Multitarea		Multitarea	
	Particiones		Memoria virtual paginada	Memoria virtual segmentada
Monotarea	Fijas	Variables	Combinación	
	Paginación pura	Segmentación pura		
	Relocalización		Protección	

GESTIÓN DE MEMORIA EN SISTEMAS OPERATIVOS MONOTAREA

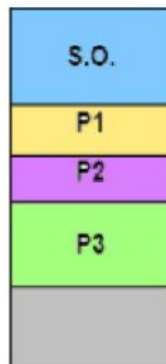
En sus orígenes los sistemas operativos no incluían ningún gestor de memoria, y el programador tenía un control completo sobre el espacio total de memoria. La memoria real se utiliza para almacenar el programa que se esté ejecutando en un momento dado. Conforme los procesos se ejecutan secuencialmente 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. Por ello, se necesita un mecanismo de protección para evitar accesos a la parte del sistema operativo de los procesos de usuario.



GESTIÓN DE MEMORIA EN SISTEMAS OPERATIVOS MULTITAREA

Actualmente la mayoría de los sistemas operativos son sistemas multitarea, en los que va a haber varios procesos simultáneamente en ejecución. Para que esto sea posible, todos estos procesos deberán estar también simultáneamente en memoria, pues ésta es una condición necesaria para que un proceso pueda ejecutarse. Por tanto, deberá haber mecanismos de gestión para distribuir la memoria principal entre todos estos procesos que quieren ejecutarse.



INTERCAMBIO O SWAPPING

Como sabemos la memoria principal es un recurso limitado, por ello puede ocurrir que haya más procesos esperando a ser cargados en memoria que zonas libres en la misma. En estos casos, el gestor de memoria sacará de la memoria algunos procesos (bloqueados, suspendidos, que estén esperando a que finalice una operación de entrada/salida, etc.) y los llevará a un área de disco (memoria secundaria), conocida como área de intercambio o de swap. A esta operación se la denomina intercambio o swapping. Los procesos permanecerán allí hasta que existan huecos libres en memoria y puedan ser recuperados de disco y reubicados en memoria principal.

MEMORIA VIRTUAL

En las técnicas de Memoria Real los procesos se cargaban enteros en la memoria y accedían a memoria físicamente, pero podría suceder que existan procesos grandes que no quepan en las particiones de la memoria y por tanto, no puedan ser cargados por completo en la memoria.

La memoria virtual da una solución a estos casos, ya que permite dividir los procesos en varias partes y cargar sólo algunas de ellas en memoria.

La memoria virtual se basa en el uso de las técnicas de paginación o segmentación o la combinación de ambas y son tan usuales que están hasta integradas en la CPU.

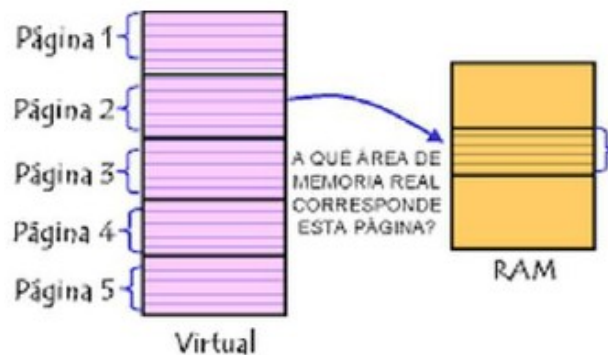
TÉCNICAS DE MEMORIA VIRTUAL

Paginación pura

La idea es la de dividir la memoria principal en un conjunto de particiones conocidas como “marcos de página” de igual tamaño.

Cada proceso se divide a su vez en una serie de partes llamadas “páginas” del mismo tamaño que los marcos.

El proceso se carga en memoria situando todas sus páginas en los marcos de página de la memoria, sin embargo, las páginas no tienen por qué estar contiguas en memoria. Como ventaja reduce la fragmentación externa de la memoria principal. Sin embargo, puede aparecer cierta fragmentación interna.



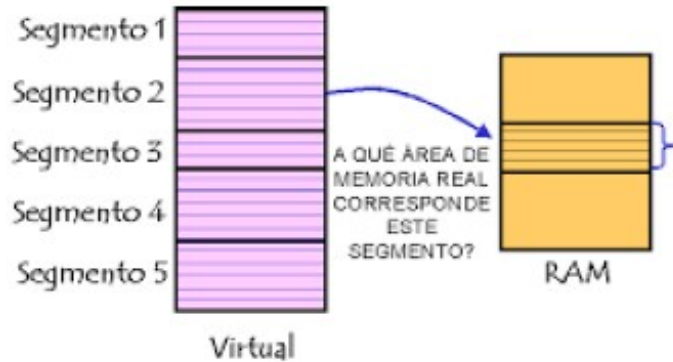
Segmentación pura

Cada proceso se divide en una serie de segmentos. La peculiaridad de estos segmentos es que su tamaño no tiene que ser el mismo y puede variar hasta un límite máximo.

Un proceso se carga situando todos sus segmentos en particiones dinámicas que no tienen que estar contiguas en memoria.

Este sistema reduce la fragmentación interna de la memoria principal. Como hemos comentado, no todas las partes de un proceso pueden estar cargadas en memoria en

un instante determinado. Por ello, cuando un proceso haga referencia a un parte que no se encuentre asignada en memoria provocará un fallo de página o segmento, y el gestor de memoria traerá dicha parte del proceso de disco a memoria.

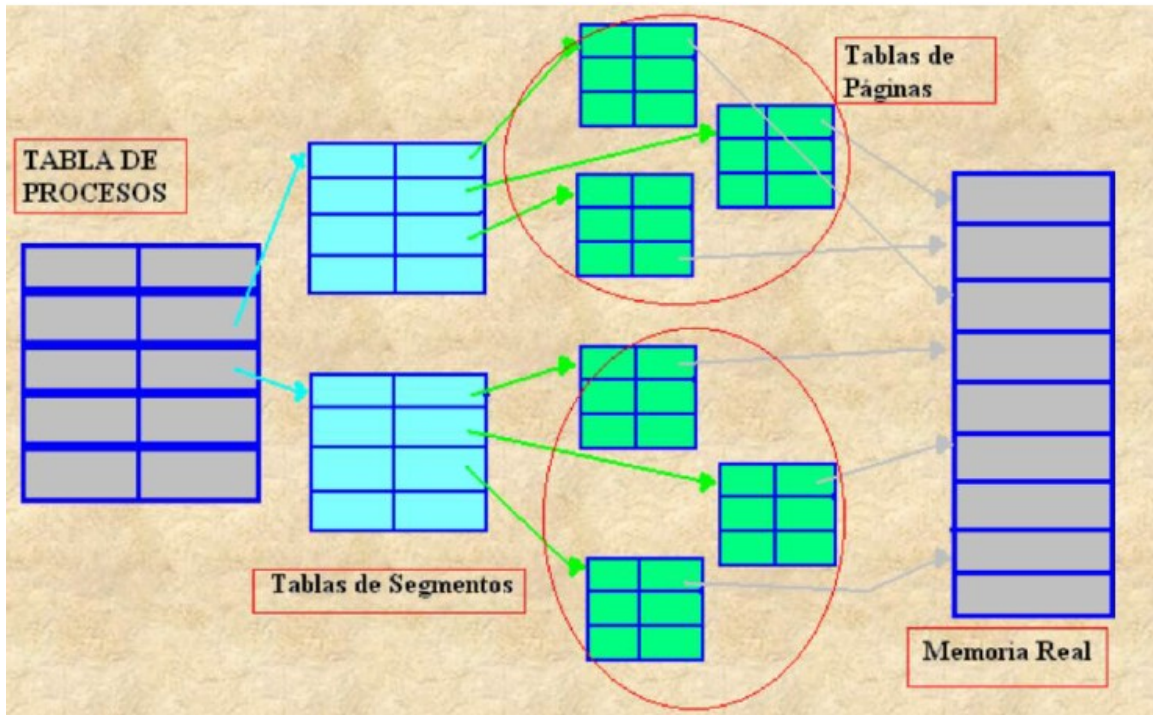


La utilización de las técnicas de paginación o segmentación por parte de la memoria virtual se conocen como:

- **Memoria Virtual Paginada:** Sigue el funcionamiento de la paginación simple, pero no es necesario cargar todas las páginas de un proceso para que éste pueda ejecutarse. Las páginas que no se encuentren y se necesiten se traerán posteriormente a memoria de manera automática. Reduce la fragmentación
- **Memoria Virtual Segmentada:** En este caso la operación sería la misma que en la segmentación simple, pero tampoco será necesario cargar todos los segmentos de un proceso. Si se necesitan más segmentos no asignados en memoria se traerán en el momento en que sean referenciados.
- **Combinación de las técnicas de segmentación y paginación:** En la figura siguiente vemos el funcionamiento de la combinación de ambas técnicas.

IMPORTANTE

1. Cada proceso tiene asignada una tabla de segmentos
2. Cada segmento tiene asignada una serie de páginas con el espacio que le haga falta.
3. La memoria Real está dividida en Marcos de páginas que albergan esta paginas
4. Si en algún momento de la ejecución del programa hace falta una página se carga en un marco.



GESTIÓN DE E/S

Una de las funciones del ordenador era procesar la información, dicha información la obtiene y muestra a través de los periféricos. La parte del sistema operativo que se encarga de este proceso es la gestión de la E/S (entrada/salida).

Los periféricos se clasificaban en periféricos:

- **De entrada:** son periféricos que reciben información y la transmiten al ordenador para su procesamiento, por ejemplo: el ratón, el teclado, el escáner, etc.
- **De salida:** periféricos que presentan la información procesada por el ordenador, por ejemplo: la impresora, el plotter (para impresión de planos y cartografía), etc.
- **De entrada y salida:** Aúnan ambas funciones, por ejemplo: el monitor, el disco duro, unidad de lectura y grabación de DVD, etc.

El sistema operativo hace que los dispositivos se conecten al sistema y realicen sus funciones de forma adecuada y eficiente. El sistema operativo abstrae de la complejidad y peculiaridad hardware de cada periférico para que las aplicaciones de usuario puedan hacer uso de los periféricos de una manera estandarizada y más sencilla.

El sistema operativo actúa pues como intermediario entre ellos, gracias a dos elementos:

- Controladores de dispositivo en particular
- Estructura de datos de la E/S del SO.

CONTROLADORES DE DISPOSITIVOS

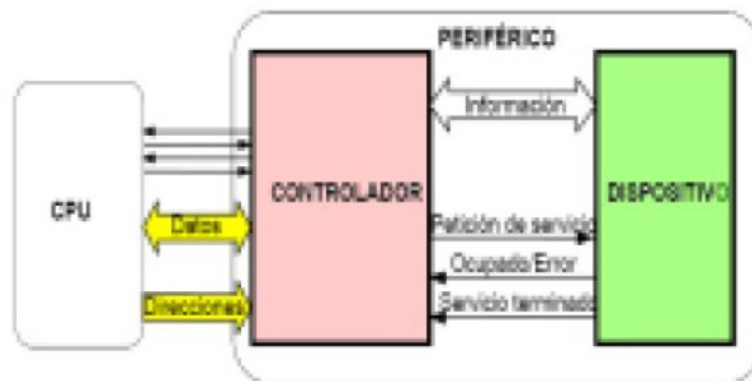
¿Cómo pueden entenderse los programas de aplicación con los dispositivos periféricos?

Como hay multitud de tipos y fabricantes de periféricos, esto conlleva que tanto el sistema operativo como los fabricantes de periféricos deben estandarizar el acceso a los dispositivos utilizando lo que se denominan controladores de dispositivos (device drivers).

Un controlador, se encarga de la comunicación entre la CPU y el dispositivo mecánico, electromecánico o electromagnético.

El controlador es un software es suministrado por el fabricante del dispositivo o bien por el desarrollador del sistema operativo.

De esta manera, estos controladores actúan como interfaz entre los programas y el hardware.



ESTRUCTURA DE DATOS DE LA E/S

El SO cuenta con una estructura de datos que utilizan los dispositivos periféricos para manejar la información y comunicación entre dispositivos o entre estos y la CPU. Las más utilizadas son los spools y los buffers.

Spools

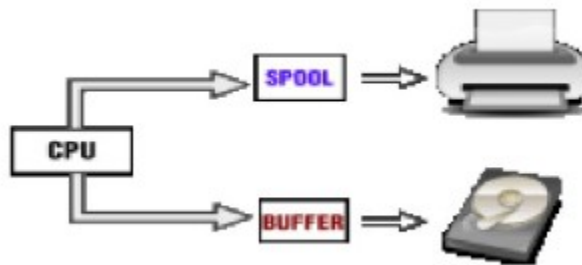
Los datos de salida se almacenan de forma temporal en una cola situada en un dispositivo de almacenamiento masivo (spool), hasta que el dispositivo periférico requerido se encuentre libre.

De este modo se evita que un programa quede retenido porque el periférico no esté disponible. El sistema operativo dispone de llamadas para añadir y eliminar archivos del spool. Se **utiliza en dispositivos que no admiten intercalación**, como ocurre en la impresora, ya que no puede empezar con otro hasta que no ha terminado.

Buffers

Es para dispositivos que pueden atender peticiones de distintos orígenes. En este caso. Los datos no tienen que enviarse completos, pueden enviarse porciones que el buffer retiene de forma temporal.

También se utilizan para acoplar velocidades de distintos dispositivos. Así, si un dispositivo lento va a recibir información más rápido de lo que puede atenderla se emplea un buffer para retener temporalmente la información hasta que el dispositivo pueda asimilarla. Esto ocurre entre una grabadora de DVD y el disco duro, ya que la primera funciona a una menor velocidad que el segundo.



TÉCNICAS DE LA E/S

Las distintas formas de funcionamiento de la E/S en los sistemas operativos según la intervención de la CPU en estos momentos tenemos:

- **E/S programada:** la CPU tiene todo el protagonismo ya que inicia y lleva a cabo la transferencia. Esta técnica repercute en la velocidad de proceso del ordenador porque la CPU debe dejar todo lo que está haciendo para ocuparse del proceso de entrada/salida.
- **E/S por interrupciones:** la CPU ejecuta la transferencia pero el inicio es pedido por el periférico que indica así su disponibilidad. La CPU no pregunta a los dispositivos sino que son estos los que la avisan cuando es necesario.
- **E/S por acceso directo a memoria (DMA):** la transferencia es realizada por un controlador especializado. Esta técnica acelera enormemente el proceso de la E/S y libera a la CPU de trabajo. Lo habitual es que los datos que se quieren escribir en el dispositivo o que son leídos del dispositivo provengan o vayan a la memoria del ordenador, pues bien en este caso. La CPU inicia el proceso, pero

SISTEMAS INFORMÁTICOS

luego este continúa sin necesitar a la CPU, con lo que se acelera mucho el proceso de entrada/salida y se libera a la CPU del proceso.

